The automultiplechoice package*

Alexis Bienvenüe
paamc@passoire.fr

April 17, 2025

Abstract

This package helps designing multiple choice exams ready for automated marking from papers scans.

Answers and questions are optionaly shuffled, creating different sheets for every student.

1 Introduction

The package automultiplechoice helps formatting multiple choice questionnaries with automated marking from papers scans in mind:

- The package can produce different copies of the question sheet for each student, optionaly shuffling answers and questions for each student.
- Markers can be printed on each sheet, so as to be able to analyse scans after examination. All the needed information about the position of the markers and the boxes to be checked by the students is given in an auxiliary file during LaTeX run.

See Auto Multiple Choice (AMC) software (https://www.auto-multiple-choice.net/) for an integration of this package, with user interface for automated marking.

2 Samples

We begin with several samples to see what can be done with the automultiplechoice package. All automultiplechoice commands and options will be detailed further.

For all these samples, two sets of questions are used: a group of geography questions, and a group of history questions. These are defined in a common LaTeX file named `questions.tex`:

```
\element{geography}{
  \begin{question}{Ghana}
    What is the capital of Ghana?
    \begin{choiceshoriz}
      \correctchoice{Accra}
      \wrongchoice{Addis Abeba}
      \wrongchoice{Ankara}
      \wrongchoice{Apia}
```

*This document corresponds to version revision: `r:ec7413fa` from AMC `1.7.0`

```
    \end{choiceshoriz}
  \end{question}
}

\element{geography}{
  \begin{question}{Thailand}
    What is the capital of Thailand?
    \begin{choiceshoriz}
      \correctchoice{Bangkok}
      \wrongchoice{Banjul}
      \wrongchoice{Beijing}
      \wrongchoice{Beirut}
      \wrongchoice{Berlin}
    \end{choiceshoriz}
  \end{question}
}

\element{geography}{
  \begin{question}{Egypt}
    What is the capital of Egypt?
    \begin{choices}
      \correctchoice{Cairo}
      \wrongchoice{Caracas}
      \wrongchoice{Cayenne}
      \wrongchoice{Chisinau}
      \wrongchoice{Conakry}
    \end{choices}
  \end{question}
}

\element{geography}{
  \begin{question}{Ireland}
    What is the capital of Ireland?
    \begin{multicols}{3}
      \begin{choices}
        \correctchoice{Dublin}
        \wrongchoice{Dili}
        \wrongchoice{Djibouti}
        \wrongchoice{Doha}
        \wrongchoice{Dakar}
        \wrongchoice{Dhaka}
      \end{choices}
    \end{multicols}
  \end{question}
}

\element{history}{
  \begin{questionmult}{1901}
    Which of the following events are taking place during the year
```

```
    1901?
    \begin{choices}
      \correctchoice{Funeral of Queen Victoria in London}
      \correctchoice{Official end of the Caste War of Yucat\'an}
      \wrongchoice{King George of Greece becomes absolute monarch of Crete}
      \wrongchoice{The first line of the Paris M\'etro is opened}
    \end{choices}
  \end{questionmult}
}

\element{history}{
  \begin{questionmult}{1850}
    Which of the following events are taking place during the year
    1850?
    \begin{choices}
      \correctchoice{American Express is founded by Henry Wells \& William Fargo}
      \wrongchoice{Napoleon Bonaparte crosses the Alps and invades Italy}
      \wrongchoice{Kwang-su becomes emperor of China}
      \wrongchoice{First horse-drawn omnibuses established in London}
    \end{choices}
  \end{questionmult}
}

\element{history}{
  \begin{questionmult}{1971}
    Which of the following events are taking place during the year
    1971?
    \begin{choices}
      \correctchoice{Apollo 14 lands on the Moon}
      \correctchoice{The Soviet Union launches Salyut 1}
      \correctchoice{Death of Louis Armstrong}
      \wrongchoice{The first commercial Concorde flight takes off}
    \end{choices}
  \end{questionmult}
}
```

We will ask automultiplechoice package to include two geography questions and two history questions at random for each student, shuffling questions and answers, with the following code:

```
\cleargroup{all}
\shufflegroup{geography}
\copygroup[2]{geography}{all}
\shufflegroup{history}
\copygroup[2]{history}{all}
\shufflegroup{all}
\insertgroup{all}
```

You can read these commands as "clear group `all`, shuffle questions inside group `geography` and copy the first two to group `all`, do the same for group `history`, shuffle the four questions copied into `all` and print them".

2.1 Standard layout

A set of 30 students sheets can be produced from the following LaTeX source named `sample-amc.tex`:

```
\documentclass{article}
\usepackage{automultiplechoice}
\usepackage{multicol}
\begin{document}

\input{questions.tex}

\onecopy{30}{

\noindent{\bf AMC  \hfill SAMPLE TEST}

\vspace{3ex}

For this test, package {\sf automultiplechoice} is used without any
option. Page markers are printed in view of an automated marking from
papers scans. DRAFT indications can be cancelled using {\tt nowatermark} option.

Commands from {\sf automultiplechoice} are used to print, for each
student, two geography questions and two history questions, at
random. Questions and answers are shuffled.

\vspace{3ex}

\cleargroup{all}

\shufflegroup{geography}
\copygroup[2]{geography}{all}
\shufflegroup{history}
\copygroup[2]{history}{all}
\shufflegroup{all}
\insertgroup{all}

}
\end{document}
```

producing a 30-pages document (every page has number 1), from which we show the first pages on page 7.

Note that "DRAFT" indications can be cancelled using option `nowatermark` , or using AMC software.

You can see on each page markers that can be used for automated completed answer sheets scans analysis:

- Four circles ● are printed in the corners, to be able to analyse any rotation or scaling of the scans.
- Binary boxes are printed in the header area, so as to be able to read student sheet number and page number. On page 2 for example, you can see that these binary boxes are coding `2/1/59`:

+2/1/59+

Here, 2 is the student sheet number, 1 is the page number for this student, and 59 is a checking value that can be used for checking correct identification from a scan.

If you also use `calibration` option , automultiplechoice will produce a `.xy` file with informations about the exact position in the page of all the markers, and all the boxes. This option is automatically set by AMC software, which then use the information in the `.xy` file for automated marking.

2.2 Separate answer sheet

In some situations, you may need a separate answer sheet:

- this makes cheating even more dificult;
- this can reduce the number of pages to scan.

This is done using `separateanswersheet` option of automultiplechoice package. You also have to use commands `\AMCformBegin` to indicate the beginning of this separate answer sheet (usually after a `\clearpage` or `\AMCcleardoublepage` command), and `\AMCform` to insert the form to be completed by the students, as in the following example (`sample-separate.tex`):

```
\documentclass{article}
\usepackage[separateanswersheet]{automultiplechoice}
\usepackage{multicol}
\begin{document}

\input{questions.tex}

\onecopy{30}{

\noindent{\bf AMC  \hfill SAMPLE TEST}

\vspace{3ex}

For this test, package {\sf automultiplechoice} is used with {\tt
  separateanswersheet} option, so that all answers are to be filled on
a separate sheet by students. Page markers are printed in view of an
automated marking from papers scans. DRAFT indications can be
cancelled using {\tt nowatermark} option.

Commands from {\sf automultiplechoice} are used to print, for each
student, two geography questions and two history questions, at
random. Questions and answers are shuffled.

\vspace{3ex}

\cleargroup{all}
```

```
\shufflegroup{geography}
\copygroup[2]{geography}{all}
\shufflegroup{history}
\copygroup[2]{history}{all}
\shufflegroup{all}
\insertgroup{all}

\clearpage

\AMCformBegin

This is the answer sheet: all answers are to be ticked on this page to
be taken into account.

\vspace{2ex}

\AMCform

}
\end{document}
```

First pages of the result are shown on page 8. There are now 2 pages per student: the first with questions, and the second for answers. Only the second will be completed by the students, and scanned for analysis.

2.3 Without markers

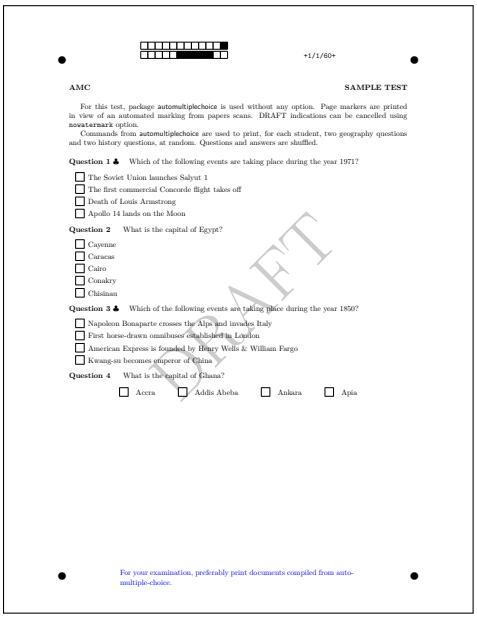
With the `nopage` option , package automultiplechoice does not include any page markers for scan processing. I'm afraid you can't use any automated marking software with this layout, but you can still use answer sheet and corrected answer sheet (option `indivanswers` , added here) for a manual marking...

The LaTeX source `sample-plain.tex` that only differs from `sample-amc.tex` by its options passed to automultiplechoice:

```
\usepackage[nopage,indivanswers]{automultiplechoice}
```

produces a 30-pages document, from which we show the first pages on page 9.

First pages from LaTeX source detailed in section 2.1 – see `sample-amc.pdf`



+1/1/60+

AMC **SAMPLE TEST**

For this test, package `automultiplechoice` is used without any option. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using `nowatermark` option.

Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

Question 1 ♣ Which of the following events are taking place during the year 1971?

- ☐ The Soviet Union launches Salyut 1
- ☐ The first commercial Concorde flight takes off
- ☐ Death of Louis Armstrong
- ☐ Apollo 14 lands on the Moon

Question 2 What is the capital of Egypt?

- ☐ Cayenne
- ☐ Caracas
- ☐ Cairo
- ☐ Conakry
- ☐ Chisinau

Question 3 ♣ Which of the following events are taking place during the year 1850?

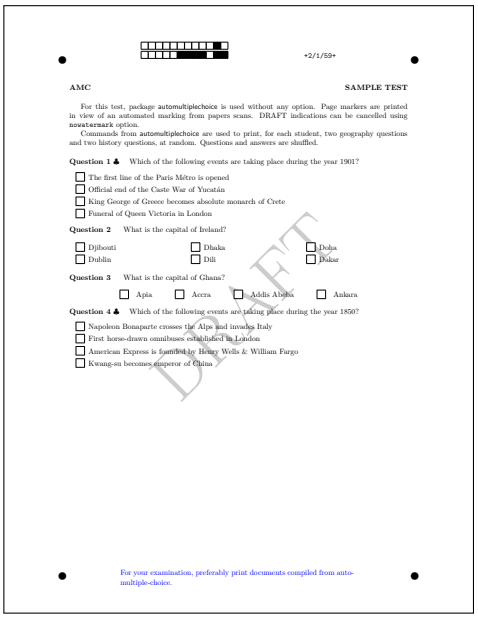
- ☐ Napoleon Bonaparte crosses the Alps and invades Italy
- ☐ First horse-drawn omnibuses established in London
- ☐ American Express is founded by Henry Wells & William Fargo
- ☐ Kwang-su becomes emperor of China

Question 4 What is the capital of Ghana?

☐ Accra ☐ Addis Abeba ☐ Ankara ☐ Apia

DRAFT

For your examination, preferably print documents compiled from automultiple-choice.



+2/1/59+

AMC **SAMPLE TEST**

For this test, package `automultiplechoice` is used without any option. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using `nowatermark` option.

Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

Question 1 ♣ Which of the following events are taking place during the year 1901?

- ☐ The first line of the Paris Métro is opened
- ☐ Official end of the Caste War of Yucatán
- ☐ King George of Greece becomes absolute monarch of Crete
- ☐ Funeral of Queen Victoria in London

Question 2 What is the capital of Ireland?

☐ Djibouti ☐ Dhaka ☐ Doha
☐ Dublin ☐ Dili ☐ Dakar

Question 3 What is the capital of Ghana?

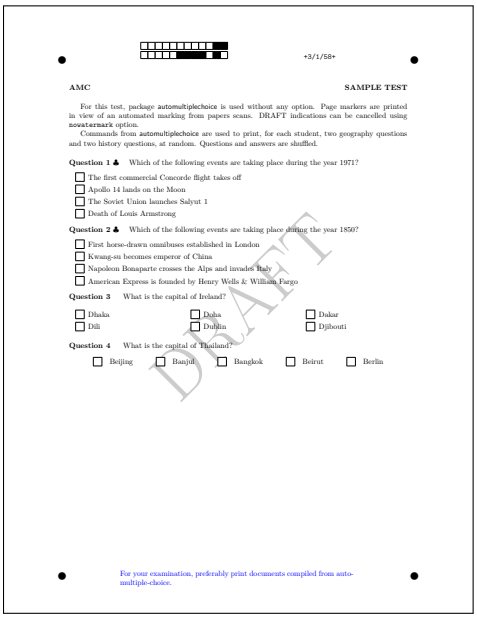
☐ Apia ☐ Accra ☐ Addis Abeba ☐ Ankara

Question 4 ♣ Which of the following events are taking place during the year 1850?

- ☐ Napoleon Bonaparte crosses the Alps and invades Italy
- ☐ First horse-drawn omnibuses established in London
- ☐ American Express is founded by Henry Wells & William Fargo
- ☐ Kwang-su becomes emperor of China

DRAFT

For your examination, preferably print documents compiled from automultiple-choice.



+3/1/58+

AMC **SAMPLE TEST**

For this test, package `automultiplechoice` is used without any option. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using `nowatermark` option.

Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

Question 1 ♣ Which of the following events are taking place during the year 1971?

- ☐ The first commercial Concorde flight takes off
- ☐ Apollo 14 lands on the Moon
- ☐ The Soviet Union launches Salyut 1
- ☐ Death of Louis Armstrong

Question 2 ♣ Which of the following events are taking place during the year 1850?

- ☐ First horse-drawn omnibuses established in London
- ☐ Kwang-su becomes emperor of China
- ☐ Napoleon Bonaparte crosses the Alps and invades Italy
- ☐ American Express is founded by Henry Wells & William Fargo

Question 3 What is the capital of Ireland?

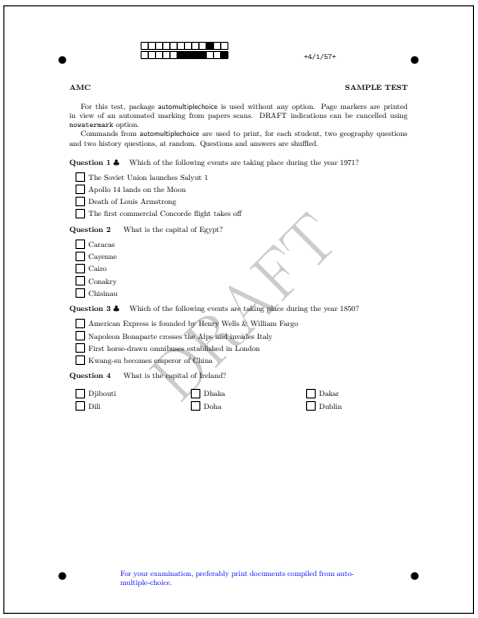
☐ Dhaka ☐ Doha ☐ Dakar
☐ Dili ☐ Dublin ☐ Djibouti

Question 4 What is the capital of Thailand?

☐ Beijing ☐ Banjul ☐ Bangkok ☐ Beirut ☐ Berlin

DRAFT

For your examination, preferably print documents compiled from automultiple-choice.



+4/1/57+

AMC **SAMPLE TEST**

For this test, package `automultiplechoice` is used without any option. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using `nowatermark` option.

Commands from `automultiplechoice` are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

Question 1 ♣ Which of the following events are taking place during the year 1971?

- ☐ The Soviet Union launches Salyut 1
- ☐ Apollo 14 lands on the Moon
- ☐ Death of Louis Armstrong
- ☐ The first commercial Concorde flight takes off

Question 2 What is the capital of Egypt?

- ☐ Caracas
- ☐ Cayenne
- ☐ Cairo
- ☐ Conakry
- ☐ Chisinau

Question 3 ♣ Which of the following events are taking place during the year 1850?

- ☐ American Express is founded by Henry Wells & William Fargo
- ☐ Napoleon Bonaparte crosses the Alps and invades Italy
- ☐ First horse-drawn omnibuses established in London
- ☐ Kwang-su becomes emperor of China

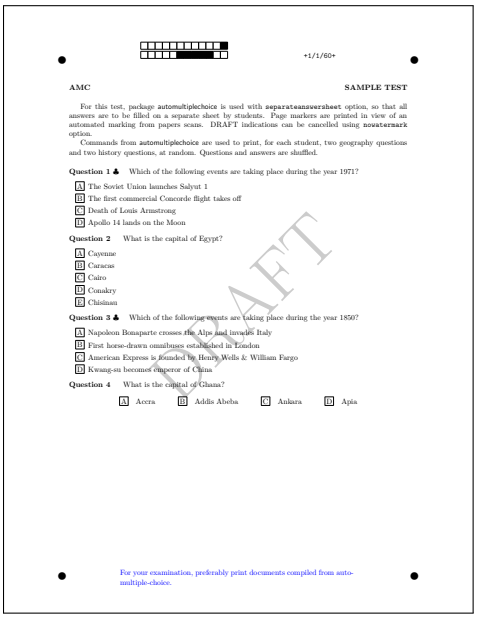
Question 4 What is the capital of Ireland?

☐ Djibouti ☐ Dhaka ☐ Dakar
☐ Dili ☐ Doha ☐ Dublin

DRAFT

For your examination, preferably print documents compiled from automultiple-choice.

First pages from LaTeX source detailed in section 2.2 – see `sample-separate.pdf`



+1/1/60+

AMC **SAMPLE TEST**

For this test, package **automultiplechoice** is used with **separateanswersheet** option, so that all answers are to be filled on a separate sheet by students. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using **nowatermark** option.

Commands from **automultiplechoice** are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

Question 1 ♣ Which of the following events are taking place during the year 1971?

- A The Soviet Union launches Salyut 1
- B The first commercial Concorde flight takes off
- C Death of Louis Armstrong
- D Apollo 14 lands on the Moon

Question 2 What is the capital of Egypt?

- A Cayenne
- B Caracas
- C Cairo
- D Conakry
- E Chisinau

Question 3 ♣ Which of the following events are taking place during the year 1850?

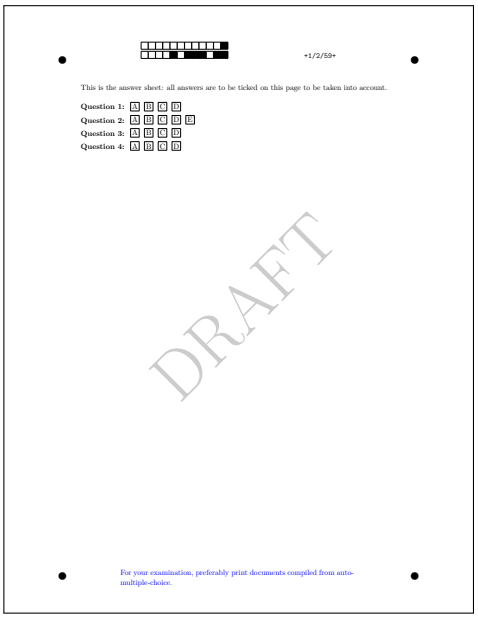
- A Napoleon Bonaparte crosses the Alps and invades Italy
- B First horse-drawn omnibuses established in London
- C American Express is founded by Henry Wells & William Fargo
- D Kwang-su becomes emperor of China

Question 4 What is the capital of Ghana?

A Accra B Addis Abeba C Ankara D Apia

DRAFT

For your examination, preferably print documents compiled from automultiple-choice.



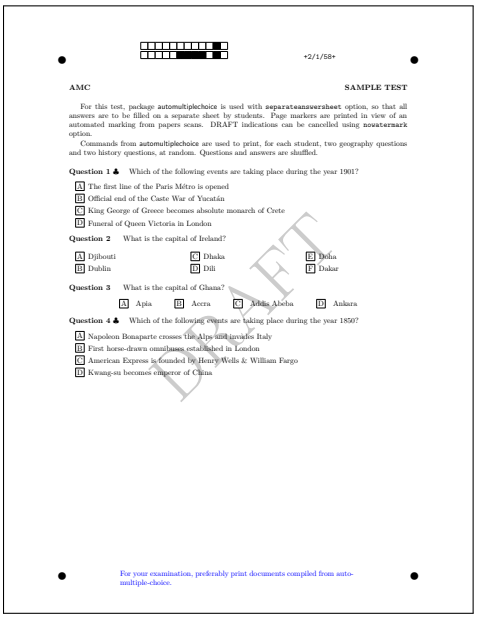
+1/2/59+

This is the answer sheet: all answers are to be ticked on this page to be taken into account.

Question 1: A B C D
Question 2: A B C D E
Question 3: A B C D
Question 4: A B C D

DRAFT

For your examination, preferably print documents compiled from automultiple-choice.



+2/1/58+

AMC **SAMPLE TEST**

For this test, package **automultiplechoice** is used with **separateanswersheet** option, so that all answers are to be filled on a separate sheet by students. Page markers are printed in view of an automated marking from papers scans. DRAFT indications can be cancelled using **nowatermark** option.

Commands from **automultiplechoice** are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

Question 1 ♣ Which of the following events are taking place during the year 1901?

- A The first line of the Paris Métro is opened
- B Official end of the Caste War of Yucatán
- C King George of Greece becomes absolute monarch of Crete
- D Funeral of Queen Victoria in London

Question 2 What is the capital of Ireland?

A Djibouti C Dhaka E Doha
B Dublin D Dili F Dakar

Question 3 What is the capital of Ghana?

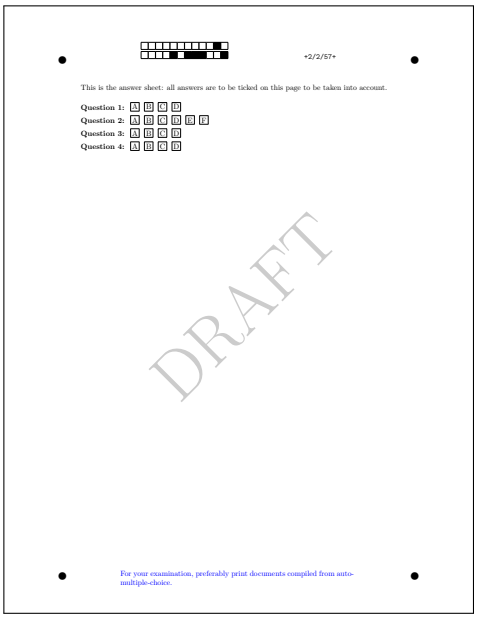
A Apia B Accra C Addis Abeba D Ankara

Question 4 ♣ Which of the following events are taking place during the year 1850?

- A Napoleon Bonaparte crosses the Alps and invades Italy
- B First horse-drawn omnibuses established in London
- C American Express is founded by Henry Wells & William Fargo
- D Kwang-su becomes emperor of China

DRAFT

For your examination, preferably print documents compiled from automultiple-choice.



+2/2/57+

This is the answer sheet: all answers are to be ticked on this page to be taken into account.

Question 1: A B C D
Question 2: A B C D E F
Question 3: A B C D
Question 4: A B C D

DRAFT

For your examination, preferably print documents compiled from automultiple-choice.

First pages from LaTeX source detailed in section 2.3 – see `sample-plain.pdf`

AMC **SAMPLE TEST**

For this test, package automultiplechoice is used with the following options:

- `nopage`, so that no page markers are printed: nothing is planed for future automated marking from papers scans.
- `indivanswers`, so that correct answers are indicated (this is the corrected answer sheet. Without this option, you get the question sheet).

Commands from automultiplechoice are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

Question 1 ♣ Which of the following events are taking place during the year 1971?

- ■ The Soviet Union launches Salyut 1
- ☐ The first commercial Concorde flight takes off
- ■ Death of Louis Armstrong
- ■ Apollo 14 lands on the Moon

Question 2 What is the capital of Egypt?

- ☐ Cayenne
- ☐ Caracas
- ■ Cairo
- ☐ Conakry
- ☐ Chisinau

Question 3 ♣ Which of the following events are taking place during the year 1850?

- ☐ Napoleon Bonaparte crosses the Alps and invades Italy
- ☐ First horse-drawn omnibuses established in London
- ■ American Express is founded by Henry Wells & William Fargo
- ☐ Kwang-su becomes emperor of China

Question 4 What is the capital of Ghana?

■ Accra ☐ Addis Abeba ☐ Ankara ☐ Apia

1

AMC **SAMPLE TEST**

For this test, package automultiplechoice is used with the following options:

- `nopage`, so that no page markers are printed: nothing is planed for future automated marking from papers scans.
- `indivanswers`, so that correct answers are indicated (this is the corrected answer sheet. Without this option, you get the question sheet).

Commands from automultiplechoice are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

Question 1 ♣ Which of the following events are taking place during the year 1901?

- ☐ The first line of the Paris Métro is opened
- ■ Official end of the Caste War of Yucatán
- ☐ King George of Greece becomes absolute monarch of Crete
- ■ Funeral of Queen Victoria in London

Question 2 What is the capital of Ireland?

☐ Djibouti ☐ Dhaka ☐ Doha
■ Dublin ☐ Dili ☐ Dakar

Question 3 What is the capital of Ghana?

☐ Apia ■ Accra ☐ Addis Abeba ☐ Ankara

Question 4 ♣ Which of the following events are taking place during the year 1850?

- ☐ Napoleon Bonaparte crosses the Alps and invades Italy
- ☐ First horse-drawn omnibuses established in London
- ■ American Express is founded by Henry Wells & William Fargo
- ☐ Kwang-su becomes emperor of China

1

AMC **SAMPLE TEST**

For this test, package automultiplechoice is used with the following options:

- `nopage`, so that no page markers are printed: nothing is planed for future automated marking from papers scans.
- `indivanswers`, so that correct answers are indicated (this is the corrected answer sheet. Without this option, you get the question sheet).

Commands from automultiplechoice are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

Question 1 ♣ Which of the following events are taking place during the year 1971?

- ☐ The first commercial Concorde flight takes off
- ■ Apollo 14 lands on the Moon
- ■ The Soviet Union launches Salyut 1
- ■ Death of Louis Armstrong

Question 2 ♣ Which of the following events are taking place during the year 1850?

- ☐ First horse-drawn omnibuses established in London
- ☐ Kwang-su becomes emperor of China
- ☐ Napoleon Bonaparte crosses the Alps and invades Italy
- ■ American Express is founded by Henry Wells & William Fargo

Question 3 What is the capital of Ireland?

☐ Dhaka ☐ Doha ☐ Dakar
☐ Dili ■ Dublin ☐ Djibouti

Question 4 What is the capital of Thailand?

☐ Beijing ☐ Banjul ■ Bangkok ☐ Beirut ☐ Berlin

1

AMC **SAMPLE TEST**

For this test, package automultiplechoice is used with the following options:

- `nopage`, so that no page markers are printed: nothing is planed for future automated marking from papers scans.
- `indivanswers`, so that correct answers are indicated (this is the corrected answer sheet. Without this option, you get the question sheet).

Commands from automultiplechoice are used to print, for each student, two geography questions and two history questions, at random. Questions and answers are shuffled.

Question 1 ♣ Which of the following events are taking place during the year 1971?

- ■ The Soviet Union launches Salyut 1
- ■ Apollo 14 lands on the Moon
- ■ Death of Louis Armstrong
- ☐ The first commercial Concorde flight takes off

Question 2 What is the capital of Egypt?

- ☐ Caracas
- ☐ Cayenne
- ■ Cairo
- ☐ Conakry
- ☐ Chisinau

Question 3 ♣ Which of the following events are taking place during the year 1850?

- ■ American Express is founded by Henry Wells & William Fargo
- ☐ Napoleon Bonaparte crosses the Alps and invades Italy
- ☐ First horse-drawn omnibuses established in London
- ☐ Kwang-su becomes emperor of China

Question 4 What is the capital of Ireland?

☐ Djibouti ☐ Dhaka ☐ Dakar
☐ Dili ☐ Doha ■ Dublin

1

3 Usage

3.1 Package options

The following options are available for package automultiplechoice:

`noshuffle` cancels answers shuffling for all questions.

`noshufflegroups` cancels groups shuffling.

`answers` produces a common corrected answers sheet.

`indivanswers` shows the boxes that corresponds to correct choices on the question sheet.

`box` includes every question in a LaTeX box, so that they can't be cutted on two different pages.

`asbox` does the same for questions in the separate answer sheet.

`separateanswersheet` asks for a separate answer sheet (see section 2.2 for an example). Commands `\AMCformBegin` and `\AMCform` must be used to describe the separate answer sheet (see section 3.6).

`digits` puts digits instead of letters in the boxes, when `separateanswersheet` (or `insidebox`) is used.

`outsidebox` prints boxes labels outside the boxes on the answersheet when `separateanswersheet` is set.

`init` initializes the random generator from time. *This option is only for testing: don't use it for a real exam!*

`completemulti` adds an answer "None of these answers are correct." at the end of each multiple question (question with no, one or several correct answers), so as to make the difference between "I don't know" and "I think none of the answers are correct".

`insidebox` puts a letter (or a digit if `digits` option is used) inside the boxes, even if `separateanswersheet` is not used. The `insidebox` option is implicitely called when using `separateanswersheet`: no need to call it then.

`calibration` asks for logging positions of boxes and markers in the `.xy` file. Without this option, a LaTeX run updates the document but not the `.xy` file.

`nowatermark` cancels the "DRAFT" indications above pages.

`catalog` is used for formatting a catalog of questions, not an exam. Then the question identifiers will be printed.

`keys` defines the way the question identifiers will be printed on the catalog file. With `keys=next` (the default), the question identifiers will be printed next to the questions numbers. With `keys=line`, the question identifiers will be printed on one line before the question text, so that the question will look close to the final result on the exam copies.

`francais` asks for french localisation.

`lang=XX` asks for localisation in `XX` language. At present, only `CA` (Catalan), `DE` (German), `ES` (Spanish), `FR` (French), `IT` (Italian), `JA` (Japanese), `NO` (Norwegian) and `NL` (Dutch) are available.

`plain` cancels environ and etex automatic loading. The default behaviour is to load environ and etex packages if available, as they improve automultiplechoice. This is not done when `plain` option is set.

`nopage` cancels markers print and page layout definition (see sample in section 2.3).

`automarks` , when used with `separateanswersheet`, cancels markers print on the subject page (they are only shown on the answer sheet pages).

`postcorrect` tells that correct answers won't be given in the LaTeX source. The teacher will fill one answer sheet for AMC to analyse the scan and set correct answers from it.

`fullgroups` cancels the use of the optional parameter of `\insertgroup` and `\copygroup`, so that each group is always fully inserted and fully copied.

`storebox` asks to use `\storebox` instead of `\savebox` to handle ovals (when using oval shape). The package `storebox` will be loaded.

`pdfform` use this option to produce PDF forms. The PDF sheet won't be printed, but filled by each student with a PDF reader. The completed PDF will then be sent to the teacher, and given to AMC for data capture.

See also section 3.8 for a french version of some of these options.

3.2 Questions and answers

We make a difference between two kind of multiple choice questions:

- **Simple questions**: there is one and only one correct choices among the proposed choices, *and this is announced to the student*. Thus, the student is asked to check one answer if he thinks this is the good one, and to check none if he has no idea.
- **Multiple questions**: there can be zero, one or several correct choices among the proposed choices. This is also announced to the student (using the `\multiSymbole` sign, with default ♣), so that the student is asked to check all the boxes corresponding to correct choices, and to let unchecked all boxes corresponding to wrong choices.

`question` (*env.*)
`questionmult` (*env.*)

Simple questions are enclosed in a `{question}{`⟨*id*⟩`}` environment, and multiple questions are enclosed in a `{questionmult}{`⟨*id*⟩`}` environment. These environments contain the question text, and the proposed choices inside a `choices`-like environment (see next). The ⟨*id*⟩ argument is a question identifier. Each question must have a unique identifier, different from the other questions identifiers.

Questions environments

Question 1 What is the elevation of Mount Everest?

- ☐ 8,253 m
- ☐ 8,810 m
- ☐ 8,848 m

Question 2 ♣ Which contries are in the Americas?

- ☐ Cambodia
- ☐ Guatemala
- ☐ Canada
- ☐ Switzerland

```
\begin{question}{everest}
  What is the elevation of
  Mount Everest?
  \begin{choices}
    \correctchoice{8,848\,m}
    \wrongchoice{8,253\,m}
    \wrongchoice{8,810\,m}
  \end{choices}
\end{question}

\begin{questionmult}{americas}
  Which contries are in the Americas?
  \begin{choices}
    \correctchoice{Guatemala}
    \correctchoice{Canada}
    \wrongchoice{Switzerland}
    \wrongchoice{Cambodia}
  \end{choices}
\end{questionmult}
```

`\AMCcompleteMulti`
`\AMCnoCompleteMulti`

For multiple questions, it is sometimes useful to make the difference between a student who thinks that none of he choices are correct, and a student who did not answer the question. The use of package option `completemulti` can be used in this case: it adds a choice to all multiple questions. Commands `\AMCcompleteMulti` and `\AMCnoCompleteMulti` can also be used to change this behaviour for a single question.

Additional answer "*none*" for a single question

Question 3 ♣ Which contries are in the Americas?

- ☐ Guatemala
- ☐ Cambodia
- ☐ Canada
- ☐ Switzerland
- ☐ *None of these answers are correct.*

```
\begin{questionmult}{americas}
  \AMCcompleteMulti
  Which contries are in the Americas?
  \begin{choices}
    \correctchoice{Guatemala}
    \correctchoice{Canada}
    \wrongchoice{Switzerland}
    \wrongchoice{Cambodia}
  \end{choices}
\end{questionmult}
```

`choices` (*env.*)
`choiceshoriz` (*env.*)
`choicescustom` (*env.*)

Depending on the formatting style for answers, one can choose one of the following ones:

- Environment `choices` is usualy chosen for long answers:

The `choices` environment

Question 4 ♣ What are the possible uses of latex?

- ☐ Latex is used as a fuel for some space launch vehicles.
- ☐ Latex from the chicle and jelutong trees is used in chewing gum.
- ☐ Natural rubber is the most important product obtained from latex.

```
\begin{questionmult}{latex}
  What are the possible uses of latex?
  \begin{choices}
    \correctchoice{Natural rubber is
      the most important product
      obtained from latex.}
    \correctchoice{Latex from the chicle
      and jelutong trees is used in
      chewing gum.}
    \wrongchoice{Latex is used as a fuel
      for some space launch vehicles.}
  \end{choices}
\end{questionmult}
```

- environment `choiceshoriz` is chosen for short answers:

The `choiceshoriz` environment

Question 5 How many legs for an insect?

☐ 2 ☐ 6 ☐ 8

```
\begin{question}{insect}
  How many legs for an insect?
  \begin{choiceshoriz}
    \correctchoice{6}
    \wrongchoice{2}
    \wrongchoice{8}
  \end{choiceshoriz}
\end{question}
```

- environment `choicescustom` is provided to customize answers formatting. See 3.9.3 for details.

`\correctchoice` `\wrongchoice`

As you have seen in these examples, the `choices`-like environments contain `\correctchoice{`⟨*text*⟩`}` and `\wrongchoice{`⟨*text*⟩`}` commands, with the text of the proposed choice as argument.

3.3 Scoring

`\scoring` `\scoringDefaultM` `\scoringDefaultS` `\QuestionIndicative`

Scoring strategies can be given in the LaTeX source. They don't have any impact on the question sheet: they are only transmitted to the analysis software through the `.amc` file. See AMC documentation to write proper commands for your needs. `\scoring{`⟨*score*⟩`}` can be used inside a `question` or `questionmult` environment to describe the scoring strategy for the question, or after a `\correctchoice` or `\wrongchoice` command to describe score associated to a particular choice. `\scoringDefaultM{`⟨*score*⟩`}` and `\scoringDefaultS{`⟨*score*⟩`}` define default scoring strategies for multiple and simple questions. `\QuestionIndicative` tags a question that is not taken into account to compute the mark – for example, it can be used for a question about the way students have enjoyed the course.

3.4 Groups of questions

Several commands are available that allows shuffling questions for each question sheet. They handle groups of questions. These groups will usualy contain questions, but can be made of any LaTeX content.

`\element` `\shufflegroup` `\insertgroup` `\insertgroupfrom`

The command `\element{`⟨*groupname*⟩`}{`⟨*content*⟩`}` adds element with content ⟨*content*⟩ to the group named ⟨*groupname*⟩. The command `\shufflegroup{`⟨*groupname*⟩`}` shuffles elements of group named ⟨*groupname*⟩. The command `\insertgroup[`⟨*n*⟩`]{`⟨*groupname*⟩`}` inserts elements of group ⟨*groupname*⟩ one after one. If optional parameter ⟨*n*⟩ is given, only the first ⟨*n*⟩ elements of the group are inserted in the document. If not, or if ⟨*n*⟩ is negative, all the elements are inserted. The command `\insertgroupfrom[`⟨*n*⟩`]{`⟨*groupname*⟩`}{`⟨*i*⟩`}` does the same, starting from element at index ⟨*i*⟩ (the first element has index 0).

As an example without questions in groups elements, let us create a small group named `serie`, containing five elements, and play with it:

Managing groups

Numbers: one two three four five.
Three numbers from the second (index=1) one: two three four.
Two of them after shuffling: two four.

```
\element{serie}{ one}
\element{serie}{ two}
\element{serie}{ three}
\element{serie}{ four}
\element{serie}{ five}
Numbers:\insertgroup{serie}.

Three numbers from the second (index=1)
one:\insertgroupfrom[3]{serie}{1}.

\shufflegroup{serie}
Two of them after
shuffling:\insertgroup[2]{serie}.
```

`\cleargroup` `\copygroup` `\copygroupfrom`

The command `\cleargroup{`⟨*groupname*⟩`}` clears all the elements of group ⟨*groupname*⟩, making an empty group. The command `\copygroup[`⟨*n*⟩`]{`⟨*from*⟩`}{`⟨*to*⟩`}` copies the elements of group ⟨*from*⟩ to group ⟨*to*⟩ – if optional parameter ⟨*n*⟩ is given, only the ⟨*n*⟩ first elements are copied. If not, or if ⟨*n*⟩ is negative, all the elements are copied. The command `\copygroupfrom[`⟨*n*⟩`]{`⟨*from*⟩`}{`⟨*to*⟩`}{`⟨*i*⟩`}` does the same, starting from element at index ⟨*i*⟩ (the first element has index 0).

As an example again without questions:

Copying elements from a group to another

Three digits from 2 to 4 and two letters: A 2 3 F 4.
Three digits and two letters: 2 8 4 E D.
Three digits and two letters: 4 E 2 5 A.

```
\element{digits}{ 1}\element{digits}{ 2}\element{digits}{ 3}
\element{digits}{ 4}\element{digits}{ 5}\element{digits}{ 6}
\element{digits}{ 7}\element{digits}{ 8}\element{digits}{ 9}
\element{letters}{ A}\element{letters}{ B}\element{letters}{ C}
\element{letters}{ D}\element{letters}{ E}\element{letters}{ F}

\shufflegroup{letters}
\cleargroup{mixed}
\copygroupfrom[3]{digits}{mixed}{1}\copygroup[2]{letters}{mixed}
\shufflegroup{mixed}
Three digits from 2 to 4 and two letters:\insertgroup{mixed}.

\shufflegroup{digits}\shufflegroup{letters}
\cleargroup{mixed}
\copygroup[3]{digits}{mixed}\copygroup[2]{letters}{mixed}
\shufflegroup{mixed}
Three digits and two letters:\insertgroup{mixed}.

\shufflegroup{digits}\shufflegroup{letters}
\cleargroup{mixed}
\copygroup[3]{digits}{mixed}\copygroup[2]{letters}{mixed}
\shufflegroup{mixed}
Three digits and two letters:\insertgroup{mixed}.
```

You can find an example involving questions in section 2.

3.5 Students identification

`\namefield` `\AMCcodeGrid` `\AMCcodeGridInt`

There are two ways to associate students to their sheets.

- Always add to one page of each copy some place for the student to write down his name. If you want AMC software to be able to cut the scan arount this area to present it to you and ask you to read the written name (this is called manual association), you must use the `\namefield{`⟨*descr*⟩`}` command. The ⟨*descr*⟩ argument contains the LaTeX code used to format the name field on the page. For example:

The name field

Name and surname:

.....

```
\namefield{\fbox{
  \begin{minipage}{15em}
    Name and surname:\vspace*{3ex}\par
    \noindent\dotfill\vspace{2mm}
  \end{minipage}
}}
```

You can see that the `\namefield` command has no effect on the produced document. In fact, its only purpose is to log in the `.xy` file information about the position of the name field on the page, to be used by the software analysing the scans.

- For automated student identification, if for example students have a 6-digits student number, you can ask them to code it somewhere on the question sheet. This can be done using the `\AMCcodeGridInt[`⟨*opts*⟩`]{`⟨*key*⟩`}{`⟨*ndigits*⟩`}` command, where ⟨*key*⟩ is the key identifier, that can be used to retrieve coded student numbers from the scans, and ⟨*ndigits*⟩ is the number of digits for numbers to be coded.

Student ID

☐0	☐0	☐0	☐0	☐0	☐0
☐1	☐1	☐1	☐1	☐1	☐1
☐2	☐2	☐2	☐2	☐2	☐2
☐3	☐3	☐3	☐3	☐3	☐3
☐4	☐4	☐4	☐4	☐4	☐4
☐5	☐5	☐5	☐5	☐5	☐5
☐6	☐6	☐6	☐6	☐6	☐6
☐7	☐7	☐7	☐7	☐7	☐7
☐8	☐8	☐8	☐8	☐8	☐8
☐9	☐9	☐9	☐9	☐9	☐9

```
\AMCcodeGridInt{student}{6}
```

For smaller number of digits, the "horizontal" form can be preferred:

Student ID, horizontal form

☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9
☐0	☐1	☐2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

```
\AMCcodeGridInt[h]{student}{3}
```

3.6 Separate answer sheet

`\AMCformBegin` `\AMCform` `\AMCcleardoublepage`

To produce separate answer sheets as seen in section 2.2,

1. use the `separateanswersheet` package option.
2. use the `\AMCformBegin` command at the beginning of the answer sheet description. This command usualy follows a command to get a new page. This command can be the classical `\clearpage` for single-sided question sheets, or the `\AMCcleardoublepage` command, that

go to the next odd numbered page, so that the answer sheet is on a separate sheet even when printing in duplex mode.

3. use the `\AMCform` command to insert all boxes for all questions.

See section 2.2 for an example.

3.7 Random computation questions

One can use the LaTeX package fp to make random computation questions, as can be seen in the following example (don't forget to load package fp):

Random computation questions

Question 6 How much are 2 plus 8?

☐ 9 ■ 10 ☐ 16 ☐ -6

```
\begin{question}{simplesum}
  \FPeval\VQa{trunc(1+random*8,0)}
  \FPeval\VQb{trunc(4+random*5,0)}
  \FPeval\VQsum{clip(VQa+VQb)}
  \FPeval\VQnoA{clip(VQa+VQb-1)}
  \FPeval\VQnoB{clip(VQa*VQb)}
  \FPeval\VQnoC{clip(VQa-VQb)}
  How much are \VQa{} plus \VQb{}?
  \begin{choiceshoriz}
    \correctchoice{\VQsum}
    \wrongchoice{\VQnoA}
    \wrongchoice{\VQnoB}
    \wrongchoice{\VQnoC}
  \end{choiceshoriz}
\end{question}
```

In this example, `\VQa` and `\VQb` are used to store two random integers (the first between 1 and 8, and the second between 4 and 8). Then `\VQsum` stores the sum of these two integers, and `\VQnoA`, `\VQnoB` and `\VQnoC` are other values that will be used as distractors in the multiple choice question.

`\AMCIntervals` In some cases, command `\AMCIntervals{`$\langle x\rangle$`}{`$\langle x0\rangle$`}{`$\langle x1\rangle$`}{`$\langle delta\rangle$`}` from automultiplechoice can be useful. It adds a sequence of choices made of intervals $[x_i, x_i+\delta[$ of length $\langle delta\rangle$ covering the interval $[\langle x0\rangle, \langle x1\rangle[$, using `\correctchoice` when $\langle x\rangle$ lies in the interval, and `\wrongchoice` otherwise.

Pick the right interval

Question 7 Let X and Y be two independent random variables, following exponential laws with respective parameters 5 and 8. In which interval lies the probability $\mathrm{P}[X < Y]$?

☐ $[0, 0.1[$	☐ $[0.2, 0.3[$	☐ $[0.4, 0.5[$	☐ $[0.6, 0.7[$	☐ $[0.8, 0.9[$
☐ $[0.1, 0.2[$	■ $[0.3, 0.4[$	☐ $[0.5, 0.6[$	☐ $[0.7, 0.8[$	☐ $[0.9, 1[$

```
\begin{question}{inf-expo-indep}
  \FPeval\VQa{trunc(2 + random * 4,0)}
  \FPeval\VQb{trunc(6 + random * 5,0)}
  \FPeval\VQr{VQa/(VQa+VQb)}
  Let $X$ and $Y$ be two independent random variables, following
  exponential laws with respective parameters \VQa{} and \VQb{}.
  In which interval lies the probability $\textrm{P}[X<Y]$?
  \begin{multicols}{5}
    \begin{choices}[o]
      \AMCIntervals{\VQr}{0}{1}{0.1}
    \end{choices}
  \end{multicols}
\end{question}
```

`\AMCnumericChoices` One can also use the `\AMCnumericChoices` command to ask the student to enter a numerical value as his answer, as in the following example:

Numeric choices

Question 8
Compute $\sqrt{11}$ and round it with two digits after period.

	☐0	☐1	☐2	■3	☐4	☐5	☐6	☐7	☐8	☐9
	.									
■+	☐0	☐1	☐2	■3	☐4	☐5	☐6	☐7	☐8	☐9
☐−	☐0	☐1	■2	☐3	☐4	☐5	☐6	☐7	☐8	☐9

```
\begin{questionmultx}{sqrt}
  \FPeval\VQa{trunc(5+random*15,0)}
  \FPeval\VQs{VQa^0.5}

  Compute $\sqrt{\VQa}$ and round it with two digits after period.

  \AMCnumericChoices{\VQs}{digits=3,decimals=2,sign=true,
    borderwidth=0pt,backgroundcol=lightgray,approx=5}
\end{questionmultx}
```

Note the use of `questionmultx` environment: we need this question to be *multiple* as several boxes has to be ticked, but we can't say that *several answers are correct*, so we don't show the ♣.

Available options that can be used in the second argument of the `\AMCnumericChoices` command are the following (⟨*bool*⟩ can be `true` or `false`, and ⟨*color*⟩ must be a color known by the `xcolor` package):

`digits=`⟨*num*⟩ gives the number of digits to request (defaults to 3).

`decimals=`⟨*num*⟩ gives the number of digits after period to request (defaults to 0). Note that when `decimals` is positive, the LaTeX package `fp` must be loaded.

`base=`⟨*num*⟩ gives the base for digits and decimals (defaults to 10).

`significant=`⟨*bool*⟩ if `true`, the numbers to code are the first *significant* digits from the first argument of `\AMCnumericChoices`. For example, the right answer to `\AMCnumericChoices {56945.23}{digits=2,significant=true}` is 57.

`exponent=`⟨*num*⟩ gives the number of digits for the exponent, when requesting to enter the result in scientific notation.

`nozero=`⟨*bool*⟩ if `true`, the choice 0 is removed for all digits. May be useful when `\AMCnumericChoices` is used to get a small (< 10) positive value.

`sign=`⟨*bool*⟩ requests (or not) a signed value (default to true).

`exposign=`⟨*bool*⟩ requests (or not) a signed value of the exponent (default to true).

`strict=`⟨*bool*⟩ if `true`, a box has to be ticked for every digit and for the sign. If `false`, if some digits has no ticked box, they will be set to zero. Defaults to `false`.

`vertical=`⟨*bool*⟩ if `true`, each digit is represented on one raw. If `false` (default), each digit is represented on one line.

`expovertical=`⟨*bool*⟩ if `true`, the mantissa is above the exponent. If `false` (default), the mantissa is beside the exponent.

`reverse=`⟨*bool*⟩ if `true`, place higher values of the digits on the top in vertical mode (defaults to `true`).

`vhead=`⟨*bool*⟩ if `true`, in vertical mode, a header is placed over all digits rows, made using the command `\AMCntextVHead` that is originally defined as `\def\AMCntextVHead#1{\emph{b#1}}`. This default value is useful to number the binary digits. Default value is `false`.

`Tvhead=`⟨*text*⟩ A coma separated list as `{H,T,O,t,h,th}` for header in vhead vertical mode. Needs `vhead` to be set (defaults to the empty list `{}`).

`vheadunitindex=`⟨*num*⟩ The index of the Ones place in the `Tvhead` list, counting from the right. If zero, changed to `decimals + 1` (defaults to zero). For example, if `Tvhead={H,T,O,t,h,th}`, should be set as `vheadunitindex=4` (at least if `decimals` is not set to 3).

`hspace=`⟨*space*⟩ sets the horizontal space between boxes (defaults to `.5em`).

`vspace=`⟨*space*⟩ sets the certical space between boxes (defaults to `1ex`).

`borderwidth=`⟨*space*⟩ sets the width of the frame around all the boxes (defaults to `1mm`).

`bordercol=`⟨*color*⟩ sets the color of the frame (defaults to `lightgray`).

`backgroundcol=`⟨*color*⟩ sets the background color (defaults to `white`).

`Tsign=`⟨*text*⟩ sets the text to print at the top of the boxes to set the sign (Can also be redefined by `\def\AMCntextSign{`⟨*text*⟩`}`, and defaults to be empty).

`Tpoint=`⟨*text*⟩ sets the text for the period. Can also be redefined by `\def\AMCdecimalPoint{`⟨*text*⟩`}`, and defaults to `\raisebox{1ex}{\bf .}`.

`Texponent=`⟨*text*⟩ sets the text before the exponent. Can also be redefined by `\def\AMCexponent{`⟨*text*⟩`}`, and defaults to `$\times10$\textasciicircum`.

`scoring=`⟨*bool*⟩ if `true`, a scoring strategy is given to AMC for this question. Defaults to `true`.

`scoreexact=`⟨*num*⟩ gives the score for an exact answer (defaults to 2).

`exact=`⟨*num*⟩ sets the maximal distance to the correct integer value (value without the decimal point) for an answer to be said *exact* and be rewarded to `scoreexact` points (defaults to 0).

`scoreapprox=`⟨*num*⟩ gives the score for an approximative answer (defaults to 1).

`approx=`⟨*num*⟩ sets the maximal distance to the correct integer value (value without the decimal point) for an answer to be said *approximative* and be rewarded to `scoreapprox` points (defaults to 0).

`scorewrong=`⟨*num*⟩ gives the score for a wrong answer (defaults to 0).

`ignoreblank` can be used (only with number base 10) to ignore digits for which no box has been ticked. This way, ticking 5 for the first digit, no box for the second and 3 for the third digit will code the number 53, while this would have coded 503 without the `ignoreblank` option (because the default value for the second digit is 0).

`keepas=`⟨*name*⟩ keeps the value entered by the student in variable `{`⟨*name*⟩`}`, for future use with *alsocorrect* in another question.

`alsocorrect=`⟨*expression*⟩ gives another acceptable answer, that can be based on the values entered by the student in the previous questions.

The text added at the end of the questions using `\AMCnumericChoices` when not in the separate answer sheet (and when a separate answer sheet is requested by the `separateanswersheet` package option) can also be set redefining the `\AMCntextGoto` command, as:

```
\def\AMCntextGoto{\par{\bf\emph{Please code the answer on
                              the separate answer sheet.}}}
```

3.8 French command names

For backward compatibility, some of automultiplechoice commands, environments and package option have their French counterpart. You can always use either the English command or the French equivalent. See table 1 for details.

type	English	French
command	`\namefield`	`\champnom`
environment	`choices`	`reponses`
environment	`choiceshoriz`	`reponseshoriz`
environment	`choicescustom`	`reponsesperso`
command	`\correctchoice`	`\bonne`
command	`\wrongchoice`	`\mauvaise`
command	`\lastchoices`	`\alafin`
command	`\AMCIntervals`	`\choixIntervalles`
command	`\scoring`	`\bareme`
command	`\scoringDefaultM`	`\baremeDefautM`
command	`\scoringDefaultS`	`\baremeDefautS`
command	`\onecopy`	`\exemplaire`
environment	`examcopy`	`copieexamen`
command	`\shufflegroup`	`\melangegroupe`
command	`\insertgroup`	`\restituegroupe`
command	`\AMCform`	`\formulaire`
command	`\AMCformBegin`	`\AMCdebutFormulaire`
option	`noshuffle`	`ordre`
option	`answers`	`correc`
option	`indivanswers`	`correcindiv`
option	`box`	`bloc`
option	`separateanswersheet`	`ensemble`
option	`digits`	`chiffres`

Table 1: French equivalent commands

3.9 Customisation

3.9.1 Boxes

`\AMCboxStyle` The command `\AMCboxStyle{`⟨*style*⟩`}` can be used to specify the shape, color and dimensions of the boxes to be ticked. The argument ⟨*style*⟩ is a coma-separated list of ⟨*key*⟩=⟨*value*⟩ pairs, with the following possible ⟨*key*⟩s:

`shape` for the shape to be used: either `square` or `oval`. Note that if `oval` is used, the LaTeX package `tikz` must be loaded.

`width` for the width of the boxes.

`height` for the height of the boxes.

`size` for the size of the boxes (sets `width` and `height`).

`down` for the length the boxes are to be moved down.

`rule` for the rule width.

`outsidesep` for the distance between the box and the letter when printed outside the box.

`color` for the color (only the box that are to be filled by the students and will be used for data capture). Use something that will be understood by the `xcolor` package.

Default values are

```
\AMCboxStyle{shape=square,size=2.5ex,down=.4ex,
             rule=.5pt,outsidesep=.1em,color=black}
```

Setting the box color allows to print the boxes with some color that won't disturb too much the data capture (for example red, but some light grey can also be considered).

Boxes styling

Question 9 $2+2=$

Ⓐ 1 Ⓑ 4 Ⓒ 10

```
\AMCboxStyle{shape=oval,color=red}
\begin{question}{sum}$2+2={}$
\begin{choiceshoriz}[o]
  \wrongchoice{1}\correctchoice{4}
  \wrongchoice{10}
\end{choiceshoriz}
\end{question}
```

3.9.2 Codes

One may adapt the codes rendering from `\AMCcodeGrid` to one's needs modifying the following lengths:

- `\AMCcodeHspace` is the amount of horizontal space between two columns of digits,
- `\AMCcodeVspace` is the amount of vertical space between two rows of digits,

Default values are `\AMCcodeHspace=.5em` `\AMCcodeVspace=.5em`

3.9.3 Answers

Environment `choicescustom` will make use of the three commands `\AMCbeginAnswer` (before the first answer), `\AMCendAnswer` (after the last answer) and `\AMCanswer{`⟨*box*⟩`}{`⟨*text*⟩`}` (for each answer) to format the answers. Redefining them properly, some different answers formatting can be achieved. However, this does not seem to work with non-trivial settings...

The `choicescustom` environment

Question 10 2+2= (☐ 4 ☐ 3 ☐ 2)

```
\begin{question}{add}
  \def\AMCbeginAnswer{$\Big($}
  \def\AMCendAnswer{$\Big)$}
  \def\AMCanswer#1#2{#1  #2\hfill}
  2+2=
  \begin{choicescustom}
    \correctchoice{4}
    \wrongchoice{2}
    \wrongchoice{3}
  \end{choicescustom}
\end{question}
```

4 Implementation

This package uses the following other packages:

```
\RequirePackage{xcolor} % \fcolorbox to fill (or not) a box
\RequirePackage{fancyhdr} % \pagestyle{empty}
\@ifl@t@r\fmtversion{2020/10/01}
    {}
    {\RequirePackage{atbegshi}} % \AtBeginShipout
\RequirePackage{xkeyval} % \setkeys
\RequirePackage{rotating} % \rotatebox
\RequirePackage{fancybox} % \boxput
\RequirePackage{expl3}
\RequirePackage{csvsimple}
\RequirePackage{environ}
%   \end{macrocode}
%
% First, we read the options that can be given by AMC through the
% |jobname-config.tex| file:
%    \begin{macrocode}
\InputIfFileExists{\jobname-config.tex}%
{\message{Loading configuration file...^^J}}{}
```

`\AMC@amclog` `\AMCmessage` Informations about questions and choices will be logged to a file with extension `amc`, to be parsed later. Macro `\AMC@amclog` writes to this file.

```
\newwrite\AMC@logfile
\immediate\openout\AMC@logfile=\jobname.amc
\def\AMC@amclog#1{\immediate\write\AMC@logfile{#1}}
\def\AMCmessage#1{\AMC@amclog{\string\message{#1}}}
```

`\AMC@LR` Colours management can be faulty in right-to-left mode: in these situations, we will make use of `\LR` from package bidi to get back to left-to-right mode. `\AMC@LR` is `\LR` if bidi is loaded.

```
\AtBeginDocument{\@ifpackageloaded{bidi}{%
  \PackageInfo{automultiplechoice}{Package bidi loaded: using LR for boxes.}%
  \let\AMC@LR=\LR}%
{\let\AMC@LR=\relax}}%
```

4.1 Variables

Counters and boolean variables defined here are internal and should not be modified by the user.

The package defines the following counters:

`\AMCload@counter` number of choices already loaded for current question.

`\AMCid@quest` current question ID number (see section 4.7).

`\AMCid@etud` current student sheet number.

`\AMCid@etudstart` starting student sheet number of the current `onecopy` bloc.

`\AMCid@check` current page checking number.

`\AMCid@etudfin` last student sheet number for the exam.

`\AMCnum@copies` number of exam sheets to produce.

It also defines the following switches:

`\ifAMC@ordre` if choices are never to be shuffled.

`\ifAMC@shuffleG` if groups shuffling is allowed.

`\ifAMC@fullGroups` if groups are always fully inserted by `\insertgroup` and fully copied by `\copygroup`, irrespective to the optional parameter.

`\ifAMC@correchead` if some correction header is to be printed at the beginning.

`\ifAMC@affichekeys` if questions keys are to be printed.

`\ifAMC@keysline` if questions keys should be printed on a single line before the question text.

`\ifAMC@correc` if correct choices are to be checked on the produced document.

`\ifAMC@qbloc` if questions are to be included in LaTeX boxes (so that they can't be splitted on two different pages).

`\ifAMC@asqbloc` if questions are to be included in LaTeX boxes in the answer sheet (so that they can't be splitted on two different pages).

`\ifAMC@rbloc` if answers are to be included in LaTeX boxes (so that they can't be splitted on two different columns for example).

`\ifAMC@textPos` if questions and answers positions are to be logged.

`\ifAMC@extractOnly` if the PDF is only built to extract questions and answers images.

`\ifAMCcomplete@multi` if a choice "None of these answers are correct." is to be added to every multiple question.

`\ifAMCquestionNumber` if AMC should step up the question number for each new question.

`\ifAMC@calibration` if this LaTeX run is used to get page layouts.

`\ifAMC@plain` if automultiplechoice won't try to load useful packages (etex, environ) that extend automultiplechoice capabilities.

`\ifAMCune@bonne` if there is at least one correct answer for the current question.

`\ifAMCtype@multi` if the current question is a multiple question.

`\ifAMC@watermark` if the document is a draft, not to be used for exam.

`\ifAMC@ensemble` if answers are to be given on a separate answers sheet.

`\ifAMC@inside@box` if a letter or digit is to be printed inside all boxes.

`\ifAMC@inside@digit` if digits are to be written inside boxes instead of letters (when using a separate answer sheet for example).

`\ifAMC@outside@box` if labels for boxes are to be printed outside the box on the answer sheet.

`\ifAMCformulaire@dedans` is true for questions inside separate answer sheet.

`\ifAMC@zoneformulaire` is true for codes (made by `\AMCcodeGrid`) inside separate answer sheet.

`\ifAMC@pagelayout` is true if the AMC page layout, with signs for scan analysis, is to be used.

`\ifAMC@postcorrect` corresponds to the use of the `postcorrect` package option.

`\ifAMC@automarks` corresponds to the use of the `automarks` package option.

`\ifAMC@invisible` is true is the DVI/PDF output is not important (used for example for scoring strategy extraction).

`\ifAMC@pdfform` is true if the output is a PDF form. This PDF will not be printed but will be filled by the students with a PDF reader and sent back to the teacher.

```
\newcount\AMCload@counter
\newcount\AMCid@quest\AMCid@quest=-1
\newcount\AMCid@check
\newcount\AMCid@etud\AMCid@etud=0
\newcount\AMCid@etudstart\AMCid@etudstart=0
\newcount\AMCid@etudfin
\newcount\AMCnum@copies
```

```
\newif\ifAMC@ordre\AMC@ordrefalse
\newif\ifAMC@shuffleG\AMC@shuffleGtrue
\newif\ifAMC@fullGroups\AMC@fullGroupsfalse
\newif\ifAMC@correchead\AMC@correcheadfalse
\newif\ifAMC@affichekeys\AMC@affichekeysfalse
\newif\ifAMC@keysline\AMC@keyslinefalse
\newif\ifAMC@correc\AMC@correcfalse
\newif\ifAMC@textPos\AMC@textPosfalse
\newif\ifAMC@extractOnly\AMC@extractOnlyfalse
\newif\ifAMC@qbloc\AMC@qblocfalse
\newif\ifAMC@asqbloc\AMC@asqblocfalse
\newif\ifAMC@rbloc\AMC@rblocfalse
\newif\ifAMCcomplete@multi\AMCcomplete@multifalse
\newif\ifAMCquestionNumber\AMCquestionNumbertrue
\newif\ifAMC@calibration\AMC@calibrationfalse
\newif\ifAMC@catalog\AMC@catalogfalse
\newif\ifAMC@plain\AMC@plainfalse
\newif\ifAMCune@bonne
\newif\ifAMCtype@multi
\newif\ifAMC@watermark\AMC@watermarktrue
\newif\ifAMC@inside@box\AMC@inside@boxfalse
\newif\ifAMC@outside@box\AMC@outside@boxfalse
\newif\ifAMC@ensemble\AMC@ensemblefalse
\newif\ifAMC@inside@digit\AMC@inside@digitfalse
\newif\ifAMCformulaire@dedans\AMCformulaire@dedansfalse
\newif\ifAMC@zoneformulaire
\newif\ifAMC@pagelayout\AMC@pagelayouttrue
\newif\ifAMC@postcorrect\AMC@postcorrectfalse
\newif\ifAMC@automarks\AMC@automarksfalse
\newif\ifAMC@invisible\AMC@invisiblefalse
\newif\ifAMC@pdfform\AMC@pdfformfalse
\let\AMCcompleteMulti=\AMCcomplete@multitrue
\let\AMCnoCompleteMulti=\AMCcomplete@multifalse
```

\AMCid@name The package also defines command \AMCid@name to be the current question identifier key.

```
\def\AMCid@name{}
```

4.2 Dimensions

\AMCformVSpace
\AMCformHSpace
\AMCinterIrep
\AMCinterBrep

The following dimensions can be modified by the user to adjust questions formatting:

\AMCformVSpace is the amount of vertical space between two questions in a separate answer sheet.

\AMCformHSpace is the amount of horizontal space between two answers boxes in a separate answer sheet.

\AMCinterIrep is the amount of vertical space to be added between two answers.

\AMCinterBrep is the amount of vertical space between two boxed answers (see \AMCBoxedAnswers and \ifAMC@rbloc).

\AMCinterIquest is the amount of vertical space left after a question, in standard mode (without package option box).

\AMCinterBquest is the amount of vertical space left after a question, in 'boxed' mode (with package option box).

\AMCpostNquest is the amount of vertical space left after a numeric question.

\AMCpostOquest is the amount of vertical space left after an open question.

```
\newdimen\AMCformVSpace\AMCformVSpace=1.2ex
\newdimen\AMCformHSpace\AMCformHSpace=.3em
\newdimen\AMCinterIrep\AMCinterIrep=\z@
\newdimen\AMCinterBrep\AMCinterBrep=.5ex
\newdimen\AMCinterIquest\AMCinterIquest=\z@
\newdimen\AMCinterBquest\AMCinterBquest=3ex
\newdimen\AMCpostNquest\AMCpostNquest=1.5ex
\newdimen\AMCpostOquest\AMCpostOquest=7mm
```

4.3 Human readable sheet ID position

\AMCidsPosition The position of the human readable sheet ID, near the corresponding binary boxes, is set with the \AMCidsPosition command, in the form \AMCidsPosition{pos=⟨*position*⟩,width=⟨*width*⟩,height=⟨*height*⟩}, where ⟨*position*⟩ is one of side (default), top and none, ⟨*width*⟩ is the width of the box enclosing the ID (default value is 4cm), and ⟨*height*⟩ is the height of the box enclosing the ID (default value is 3ex).

```
\newif\ifAMCids@top
\newif\ifAMCids@side
\newdimen\AMCids@width
\newdimen\AMCids@height
\define@choicekey*{AMCids}{pos}[\AMCidsVar\AMCidsVarN]{none,top,side}{%
  \ifcase\AMCidsVarN\relax
    \AMCids@topfalse\AMCids@sidefalse
  \or
    \AMCids@toptrue\AMCids@sidefalse
  \or
```

```
    \AMCids@topfalse\AMCids@sidetrue
  \fi
}
\define@key{AMCids}{width}{\AMCids@width=#1}
\define@key{AMCids}{height}{\AMCids@height=#1}
\def\AMCidsPosition#1{\setkeys{AMCids}{#1}}
\AMCidsPosition{pos=side,width=4cm,height=3ex}
```

4.4 Localisation

In this section, some localised strings or commands are defined, for English, French ans Spanish languages.

`\AMCtext` To modify these texts, you can use command `\AMCtext`. For example, `\AMCtext{draft}{`⟨*text*⟩`}` sets the text to be printed behind each page of a draft exam.

```
\def\AMCtext#1#2{\expandafter\def\csname AMC@loc@#1\endcsname{#2}}
\def\AMClocalized#1{\csname AMC@loc@#1\endcsname}
```

4.4.1 English

Text indicating draft exams:

```
\def\AMC@loc@draft{DRAFT}
```

Message at page bottom when compiled out of AMC gui:

```
\def\AMC@loc@message{For your examination, preferably print
  documents compiled from auto-multiple-choice.}
```

Annoucing a question in a separate sheet (parameter `#1` is the question number):

```
\def\AMC@loc@qf#1{\textbf{Question #1:}}
```

Annoucing a question (parameter `#1` is the question number and pamareter `#2` can be the multiple question symbol, or be empty):

```
\def\AMC@loc@q#1#2{\textbf{Question #1} #2}
```

Headers for corrected version and catalog:

```
\def\AMC@loc@corrected{Corrected}
\def\AMC@loc@catalog{Catalog}
```

Localization text for Explanation

```
\def\AMC@loc@explain{\textit{\textbf{Explanation: }}}
```

Last choice added at the end for multiple questions when option `completemulti` is used:

```
\def\AMC@loc@none{None of these answers are correct.}
```

Word for 'question', singular and plural forms:

```
\def\AMC@loc@question{question}
\def\AMC@loc@questions{questions}
```

Default text to write in the students' name box:

```
\def\AMC@loc@namesurname{Name and surname:}
```

4.4.2 Catalan

Catalan localisation is called with option `lang=CA`.

```
\def\AMC@loc@CA{
  \def\AMC@loc@draft{PROJECTE}
  \def\AMC@loc@message{Pel vostre examen, imprimiu preferiblement
    els documents compilats amb l'ajuda de auto-multiple-choice.}
  \def\AMC@loc@qf##1{\textbf{Pregunta ##1 :}}
  \def\AMC@loc@q##1##2{\textbf{Pregunta ##1} ##2}
  \def\AMC@loc@corrected{Correcci\'o}
  \def\AMC@loc@catalog{Cat\`aleg}
  \def\AMC@loc@explain{\textit{\textbf{Explicaci\'o : }}}
  \def\AMC@loc@none{Cap de les respostes \'es correcte.}
  \def\AMC@loc@question{pregunta}
  \def\AMC@loc@questions{preguntes}
  \def\AMC@loc@namesurname{Nom i cognoms:}
}
```

4.4.3 Dutch

Dutch localisation is called with option `lang=NL`.

```
\def\AMC@loc@NL{
  \def\AMC@loc@draft{Ontwerp}
  \def\AMC@loc@message{Gebruik bij uw proefwerk bij voorkeur die
    documenten welke door auto-multiple-choice zijn aangemaakt.}
  \def\AMC@loc@qf##1{\textbf{Vraag ##1 :}}
  \def\AMC@loc@q##1##2{\textbf{Vraag ##1} ##2}
  \def\AMC@loc@corrected{Correctie}
  \def\AMC@loc@catalog{Catalogus}
  \def\AMC@loc@none{Geen van de antwoorden is juist.}
  \def\AMC@loc@question{vraag}
  \def\AMC@loc@questions{vragen}
  \def\AMC@loc@namesurname{Achternaam en voornaam:}
}
```

4.4.4 French

French localisation is called with option `francais`, or `lang=FR`.

```
\def\AMC@loc@FR{
  \def\AMC@loc@draft{PROJET}
  \def\AMC@loc@message{Pour votre examen, imprimez de pr\'ef\'erence
    les documents compil\'es \`a l'aide de auto-multiple-choice.}
  \def\AMC@loc@qf##1{\textbf{Question ##1 :}}
  \def\AMC@loc@q##1##2{\textbf{Question ##1} ##2}
  \def\AMC@loc@corrected{Correction}
  \def\AMC@loc@catalog{Catalogue}
  \def\AMC@loc@explain{\textit{\textbf{Explication : }}}
  \def\AMC@loc@none{Aucune de ces r\'eponses n'est correcte.}
  \def\AMC@loc@question{question}
  \def\AMC@loc@questions{questions}
  \def\AMC@loc@namesurname{Nom et pr\'enom :}
}
```

4.4.5 German

German localisation is called with option `lang=DE`.

```
\def\AMC@loc@DE{
  \def\AMC@loc@draft{ENTWURF}
  \def\AMC@loc@message{Benutzen Sie f\"ur Ihre Pr\"ufung bevorzugt Dokumente die mit
    auto-multiple-choice erstellt wurden.}
  \def\AMC@loc@qf##1{\textbf{Frage ##1 :}}
  \def\AMC@loc@q##1##2{\textbf{Frage ##1} ##2}
  \def\AMC@loc@corrected{Korrektur}
  \def\AMC@loc@catalog{Katalog}
  \def\AMC@loc@explain{\textit{\textbf{Erkl\"arung : }}}
  \def\AMC@loc@none{Keine dieser Antworten ist korrekt.}
  \def\AMC@loc@question{Frage}
  \def\AMC@loc@questions{Fragen}
  \def\AMC@loc@namesurname{Vor- und Nachname:}
}
```

4.4.6 Italian

Italian localisation is called with option `lang=IT`.

```
\def\AMC@loc@IT{
  \def\AMC@loc@draft{BOZZA}
  \def\AMC@loc@message{Per l'esame, \`e preferibile stampare i documenti
     a partire da auto-multiple-choice.}
  \def\AMC@loc@qf##1{\textbf{Domanda ##1:}}
  \def\AMC@loc@q##1##2{\textbf{Domanda ##1} ##2}
  \def\AMC@loc@corrected{Correzione}
  \def\AMC@loc@catalog{Catalogo}
  \def\AMC@loc@none{Nessuna risposta \`e giusta.}
  \def\AMC@loc@question{domanda}
  \def\AMC@loc@questions{domande}
  \def\AMC@loc@namesurname{Nome e cognome:}
}
```

4.4.7 Norwegian

Norwegian localisation is called with option `lang=NO`.

```
\def\AMC@loc@NO{
  \def\AMC@loc@draft{UTKAST}
  \def\AMC@loc@message{Det anbefales {\aa} skrive ut dokumentet
  for gjennomgang \\direkte fra auto-multiple-choice.}
  \def\AMC@loc@qf##1{\textbf{Oppgave ##1 :}}
  \def\AMC@loc@q##1##2{\textbf{Oppgave ##1} ##2}
  \def\AMC@loc@corrected{Rettet}
  \def\AMC@loc@catalog{Katalog}
  \def\AMC@loc@none{Ingen svar er riktige.}
  \def\AMC@loc@question{oppgave}
  \def\AMC@loc@questions{oppgave}
  \def\AMC@loc@namesurname{Etternavn og fornavn:}
}
```

4.4.8 Portuguese

Portuguese localisation is called with option `lang=PT`.

```
\def\AMC@loc@PT{
  \def\AMC@loc@draft{RASCUNHO}
  \def\AMC@loc@message{Para o seu exame, use preferencialmente documentos compilados do auto-multiple-ch
  \def\AMC@loc@qf##1{\textbf{Quest\~ao ##1:}}
  \def\AMC@loc@q##1##2{\textbf{Quest\~ao ##1} ##2}
  \def\AMC@loc@corrected{Corrigido}
  \def\AMC@loc@catalog{Cat\'alogo}
  \def\AMC@loc@explain{\textit{\textbf{Justifique: }}}
  \def\AMC@loc@none{Nenhuma das respostas apresentadas est\'a correta.}
  \def\AMC@loc@question{Quest\~ao}
  \def\AMC@loc@questions{Quest\~oes}
  \def\AMC@loc@namesurname{Nome e apelido:}
}
```

4.4.9 Spanish

Spanish localisation is called with option `lang=ES`.

```
\def\AMC@loc@ES{
  \def\AMC@loc@draft{BORRADOR}
  \def\AMC@loc@message{Para revisi\'on, preferentemente imprimir documentos compilados
    desde auto-multiple-choice.}
  \def\AMC@loc@qf##1{\textbf{Pregunta ##1 :}}
  \def\AMC@loc@q##1##2{\textbf{Pregunta ##1} ##2}
  \def\AMC@loc@corrected{Correcci\'on}
  \def\AMC@loc@catalog{Cat\'alogo}
  \def\AMC@loc@none{Ninguna de estas preguntas son correctas.}
  \def\AMC@loc@question{pregunta}
  \def\AMC@loc@questions{preguntas}
  \def\AMC@loc@namesurname{Nombre y apellidos:}
}
```

4.4.10 Japanese

Japanese localisation is called with option `lang=JA`. It includes UTF8 encoded Japanese characters which are shown as ◇ here (look at the `.sty` file to see them).

```
\def\AMC@loc@JA{
  \def\AMC@loc@draft{◇◇◇◇}
  \def\AMC@loc@message{◇◇◇◇◇◇◇◇auto-multiple-choice◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇}
  \def\AMC@loc@qf##1{\textbf{◇##1:}}
  \def\AMC@loc@q##1##2{\textbf{◇##1} ##2}
  \def\AMC@loc@corrected{◇◇◇◇}
  \def\AMC@loc@catalog{◇◇◇◇◇◇}
  \def\AMC@loc@explain{\textit{\textbf{◇◇: }}}
  \def\AMC@loc@none{◇◇◇◇◇}
  \def\AMC@loc@question{◇}
  \def\AMC@loc@questions{◇}
}
```

4.4.11 Other languages

Other languages can be integrated to automultiplechoice package upon request to the author.

4.5 Interaction with other packages

4.5.1 cleveref

For references to questions:

```
\AtBeginDocument{\@ifpackageloaded{cleveref}{%
    \message{AMC/cleveref integration loaded^^J}%
    \crefalias{AMCquestionaff}{question}%
    \crefname{question}{\AMC@loc@question}{\AMC@loc@questions}%
}{}}%
```

4.6 Random

4.6.1 Random pseudo-generator

The package uses the pseudo-random bit generator from *TuGBoat* 1994, vol 15:1:

```
\ifx\AMC@SR\undefined\newcount\AMC@SR\fi
\providecommand\AMC@SRconst{2097152}
\providecommand\AMC@SRset[1]{\global\AMC@SR#1 \ignorespaces}
\providecommand\AMC@SRadvance{%
  \begingroup%
    \ifnum\AMC@SR<\AMC@SRconst\relax\AMC@SR@count\z@\else\AMC@SR@count\@ne\fi%
    \ifodd\AMC@SR\advance\AMC@SR@count\@ne\fi%
    \global\divide\AMC@SR\tw@%
    \ifodd\AMC@SR@count\global\advance\AMC@SR\AMC@SRconst\relax\fi%
  \endgroup}
\providecommand\AMC@SRbit{\AMC@SRadvance\ifodd\AMC@SR1\else0\fi}
\providecommand\AMC@SRtest[2]{\AMC@SRadvance%
  \ifodd\AMC@SR#2\else#1\fi\ignorespaces}
\providecommand\AMC@SRvalue{\number\AMC@SR}
```

`\AMCrandomseed` The seed of this generator is set to 1515, but another value can be given using the command `\AMCrandomseed{`⟨*seed*⟩`}`.

```
\AMC@SRset{1515}
\def\AMCrandomseed#1{\AMC@SRset{#1}}
```

4.6.2 Uniform random deviates

`\AMC@SRnextByte` `\AMC@SRmax` This generator is used to build first a 20-bit uniform integer generator (macro `\AMC@SRnextByte`). Then, using modulo, a (nearly) uniform generator on $\{0, \ldots, n-1\}$ is built: command `\AMC@SRmax{`n`}` puts in `\AMC@SR@count` the random deviate.

```
\newcount\AMC@SR@count
\def\AMC@SR@time{\AMC@SRset{\time}}
\newcount\AMC@SRnum
\def\AMC@SRnextByte{\AMC@SRnum=\z@%
  \AMC@SR@count=20%
  \loop\multiply\AMC@SRnum\tw@%
     \AMC@SRtest{\advance\AMC@SRnum\@ne}{}%
  \ifnum\AMC@SR@count>\@ne\advance\AMC@SR@count\m@ne\repeat%
```

```
}
\newcommand\AMC@SRmax[1]{\AMC@SRnextByte%
  \AMC@SR@count=\AMC@SRnum%
  \divide\AMC@SR@count by #1\relax%
  \multiply\AMC@SR@count by #1\relax%
  \advance\AMC@SRnum by -\AMC@SR@count%
}
```

4.6.3 Tokens shuffling

`\AMCsw@p` `\AMC@shuffletoks` The package defines the macro `\AMCsw@p` to swap the values of two token registers given as parameters.

After defining n token registers `\foo@i`, `\foo@ii`, `\foo@iii`, `\foo@iv` and so on, you can shuffle them using `\AMC@shuffletoks[`$\langle a\rangle$`]{`$\langle n\rangle$`}{`$\langle foo\rangle$`}`. With optional argument $\langle a\rangle$, registers are shuffled from number $\langle a\rangle$ to $\langle n\rangle$ (default value for $\langle a\rangle$ is 1).

```
\newcount\AMC@sti
\newcount\AMC@stil
\newtoks\AMCsw@p@
\newcommand\AMCsw@p[2]{%
  \global\AMCsw@p@=#1%
  \global#1=#2%
  \global#2=\AMCsw@p@}
\newcommand{\AMC@shuffletoks}[3][\@ne]{%
  \AMC@sti=#2\relax%
  \AMC@stil=#2\relax%
  \advance\AMC@stil\@ne%
  \advance\AMC@stil -#1\relax%
  \@whilenum\AMC@sti>#1\do{%
    \AMC@SRmax{\AMC@stil}\advance\AMC@SRnum #1\relax%
    \AMCsw@p{\csname #3\romannumeral\AMC@SRnum\endcsname}%
            {\csname #3\romannumeral\AMC@sti\endcsname}%
    \advance\AMC@sti\m@ne\relax%
    \advance\AMC@stil\m@ne\relax%
  }}
```

4.7 Keys numbering

`\AMC@unnumero` `\AMC@affecte` This package allocates a unique integer ID to each question key from the questionnary. The counter `\AMC@numerotation` keeps track of the number of keys which already had an ID. Command `\AMC@definitnumero{`n`}{key}` allocates ID n to the key `key`. Command `\AMC@prepare{key}` looks if an ID had already been associated to `key`, and, if not, makes a new ID allocation for `key`. Command `\AMC@unnumero{key}` returns the ID associated with `key` (creating one if necessary). Command `\AMC@affecte{key}{\cnt}` give to counter `\cnt` the value of the ID associated to `key` (creating one if necessary).

```
\newcount\AMC@numerotation\AMC@numerotation=\z@%
\def\AMC@definitnumero#1#2{\AMCmessage{NUM=#1=#2}%
  \expandafter\global\expandafter\def\csname AMC@numtab@#2\endcsname{#1}}
\def\AMC@prepare#1{\expandafter\ifx\csname AMC@numtab@#1\endcsname\relax%
  \global\advance\AMC@numerotation\@ne%
  \expandafter\AMC@definitnumero\expandafter{\the\AMC@numerotation}{#1}\fi}
\def\AMC@unnumero#1{\AMC@prepare{#1}\csname AMC@numtab@#1\endcsname}
\def\AMC@affecte#1#2{\AMC@prepare{#1}\global#2=\csname AMC@numtab@#1\endcsname}
```

4.8 Boxes

4.8.1 Character logging

\AMC@logchar The command \AMC@logchar{⟨*char*⟩}{⟨*key*⟩} logs the character written in the box referenced as ⟨*key*⟩ in the .amc file. This is used in catalog mode, to get understandable references to answers from the statistics tables of the ODS export.

```
\def\AMC@logchar#1#2{%
  \protected@write\AMC@logfile{}{%
    \string\answer%
    {\the\AMCid@etud/\thepage:#2}%
    {#1}}%
}
```

4.8.2 Position logging

\AMC@tracebox \AMC@pagepos Command \AMC@tracebox{⟨*trace*⟩}{⟨*key*⟩}{⟨*content*⟩} makes a LaTeX box around ⟨*content*⟩, and, if ⟨*trace*⟩ is not empty, logs to the .xy file informations to be able to compute exact location of this box on the page, attached to the box identification ⟨*key*⟩.

Command \AMC@pagepos logs page and page size informations at the beginning of each page.

```
\def\AMC@shapename@{\ifAMC@invisible none\else\AMC@shapename\fi}
\def\AMC@tracepos#1#2{%
  \ifAMC@calibration\ifx\@empty#1\@empty\else%
  \pdfsavepos\protected@write\AMC@XYFILE{}{%
    \string\tracepos%
    {\the\AMCid@etud/\thepage:#2}%
    {\noexpand\number\pdflastxpos sp}%
    {\noexpand\number\pdflastypos sp}%
    {\AMC@shapename}}%
  \fi\fi}
\def\AMC@traceposx#1#2{%
  \ifAMC@calibration\ifx\@empty#1\@empty\else%
  \pdfsavepos\protected@write\AMC@XYFILE{}{%
    \string\tracepos%
    {\the\AMCid@etud/\thepage:#2}%
    {\noexpand\number\pdflastxpos sp}%
    {0sp}%
    {\AMC@shapename}}%
  \fi\fi}
\def\AMC@traceposy#1#2{%
  \ifAMC@calibration\ifx\@empty#1\@empty\else%
  \pdfsavepos\protected@write\AMC@XYFILE{}{%
    \string\tracepos%
    {\the\AMCid@etud/\thepage:#2}%
    {0sp}%
    {\noexpand\number\pdflastypos sp}%
    {\AMC@shapename}}%
\fi\fi}
\newcommand\AMC@tracebox[3]{%
  \vbox{\AMC@traceposy{#1}{#2}%
    \hbox{\AMC@traceposx{#1}{#2}#3\AMC@traceposx{#1}{#2}}%
    \AMC@traceposy{#1}{#2}}}
\def\AMC@pagepos{%
```

```
  \ifAMC@calibration\protected@write\AMC@XYFILE{}{%
    \string\page%
    {\the\AMCid@etud/\thepage/\the\AMCid@check}%
    {\the\paperwidth}{\the\paperheight}%
    {\the\pdfpagewidth}{\the\pdfpageheight}}\fi}
```

`\AMCdontScan` `\AMCdontAnnotate` `\AMCreTick` The commands `\AMCdontScan`, `\AMCdontAnnotate` and `\AMCreTick` write into the `xy` file instructions related to the current question.

```
\newcommand{\AMCdontScan}{\ifAMC@calibration\immediate\write\AMC@XYFILE{\string\dontscan{\the\AMCid@etud
\newcommand{\AMCdontAnnotate}{\ifAMC@calibration\immediate\write\AMC@XYFILE{\string\dontannotate{\the\AM
\newcommand{\AMCreTick}{\ifAMC@calibration\immediate\write\AMC@XYFILE{\string\retick{\the\AMCid@etud,\th
%
```

`\AMC@tracechar` The macro `\AMC@tracechar{`⟨*char*⟩`}{`⟨*unused*⟩`}{`⟨*trace*⟩`}{`⟨*key*⟩`}` is used to log (for further processing with AMC), into to `.xy` file, the character used to identify the box.

```
\newcommand\AMC@tracechar[4]{%
  \ifAMC@calibration\ifx\@empty#3\@empty\else%
    \protected@write\AMC@XYFILE{}{%
      \string\boxchar{\the\AMCid@etud/\thepage:#4}{#1}%
    }%
  \fi\fi%
}
```

`amcxyfile` (*env.*) The following lines defines an environment to tag positions outputs for a particular part of the document. This is used mainly for documentation or testing.

```
\newenvironment{amcxyfile}[1]{%
  \protected@write\AMC@XYFILE{}{\string\xyopen{#1}}%
}{%
  \protected@write\AMC@XYFILE{}{\string\xyclose{}}%
}
```

`\AMCzone` The `\AMCzone[`⟨*flags*⟩`]{`⟨*zone name*⟩`}{`⟨*zone content*⟩`}` is a simple call to `\AMC@tracebox`:

```
\newcommand{\AMCzone}[3][]{\AMC@tracebox{1}{__zone:#1:#2}{#3}}
```

`\namefield` The `\namefield{`⟨*name field content*⟩`}` is a simple call to `\AMCzone`:

```
\newcommand{\namefield}[2][id]{\AMCzone[#1]{__n}{#2}}
```

It is used to enclose the page region where students are to write their names, so as te retreive it easily from the scans.

`\namefielddots` The command `\namefielddots` can be used to fill a line with dots (printed sheets) or use a text field in PDF forms:

```
\newcommand{\namefielddots}{%
  \noindent%
  \ifAMC@pdfform%
    \hspace*{\fill}%
    \TextField[name={\the\AMCid@etud:namefield},width=.95\linewidth,bordercolor=0 0 0]{}%
    \hspace*{\fill}
  \else%
    \dotfill
  \fi%
}
```

As an example,

```
\namefield{\fbox{%
  \begin{minipage}{5cm}
    Name:

    \vspace*{.5cm}
    \namefielddots
    \vspace{2mm}
  \end{minipage}}}
```

produces the following box:

Name:

. . . . . .

and outputs information about the position of the box in the `.xy` file, as seen in section 5.1.

4.8.3 Boxes to be checked by students

`\AMC@answerBox@` There are two styles for boxes to be checked by the students. The first one is an empty box, printed beside the answer. The sencond is a box with a character in it. It is mainly used when answers are to be given on a separate answer sheet.

These boxes can be drawn using command `\AMC@answerBox@{`⟨*char*⟩`}{`⟨*answer*⟩`}{`⟨*trace*⟩`}{`⟨*key*⟩`}`: ⟨*char*⟩ is the character to print inside the box, ⟨*trace*⟩ is non-empty if you want to log the box position in the `.xy` file, ⟨*key*⟩ is the box identification, and ⟨*answer*⟩ is an answer to be written in the box (or `\AMC@checkedbox` for filling the box).

Depending onthe required shape for the boxes, the corresponding

`\AMC@shape@xxx{`⟨*char*⟩`}{`⟨*answer*⟩`}{`⟨*trace*⟩`}{`⟨*key*⟩`}`

command is used.

- `\AMC@answerBox@{K}{}{1}{test}` produce the box K, writing the lines in the `.xy` file shown in section 5.2.
- `\AMC@answerBox@{K}{\AMC@checkedbox}{}{}` produces ■
- `\AMC@answerBox@{}{8}{}{}` produces 8
- `\AMC@answerBox@{K}{8}{1}{testb}` produces K with `\AMCboxStyle{shape=oval,color=red}`

```
\def\AMC@checkedbox{}
\let\AMC@new@savebox=\newsavebox
\let\AMC@save@box=\savebox
\let\AMC@use@box=\usebox
\newif\ifAMC@draw@cross
```

The `\AMC@smashcentered{`⟨*text*⟩`}` command shows the ⟨*text*⟩ centered at point.

```
\newbox\AMC@smashbox
\newdimen\AMC@smashboxheight
\newcommand{\AMC@smashcentered}[1]{{%
```

```
    \setbox\AMC@smashbox\hbox{#1}%
    \AMC@smashboxheight=\ht\AMC@smashbox%
    \advance\AMC@smashboxheight by \dp\AMC@smashbox%
    \vfuzz=\AMC@smashboxheight\hfuzz=\wd\AMC@smashbox%
    \hspace*{-.5\wd\AMC@smashbox}\hbox to .5\wd\AMC@smashbox{%
      \vbox to 0pt{%
        \vspace*{-.5\AMC@smashboxheight}\vbox to .5\AMC@smashboxheight{%
          \box\AMC@smashbox}}}%
  }}
```

`\AMC@setcolors@{`⟨*trace*⟩`}{`⟨*answer*⟩`}` sets colours `\AMC@boxcolor@` and `\AMC@fillcolor@` according to its arguments. It also sets the `\ifAMC@draw@cross` switch if AMC should draw a cross instead of filling the box.

```
\newcommand\AMC@setcolors@[2]{%
  \def\AMC@boxcolor@{\AMC@boxcolor}%
  \ifx\@empty#1\@empty \def\AMC@boxcolor@{black}\fi%
  \ifAMC@correc\def\AMC@boxcolor@{black}\fi%
  \def\AMC@fillcolor@{\ifx #2\AMC@checkedbox%
    \AMC@boxcolor@\else white\fi}%
  \AMC@draw@crossfalse%
  \ifKV@AMCdim@cross\ifx #2\AMC@checkedbox%
    \AMC@draw@crosstrue\fi\fi%
}
\newcommand\AMC@answerBox@[4]{%
  \ifAMC@catalog%
    \AMC@logchar{#1}{#4}%
  \fi%
  \AMC@LR{\hspace{0pt}%
    \lower\AMC@boxeddown\hbox{\csname AMC@shape@\AMC@shapename@\endcsname%
      {\AMCchoiceLabelFormat{#1}}{#2}{#3}{#4}}}%
}
\newcommand\AMC@shapeprepare@square{}
\newcommand\AMC@shape@square[4]{%
  \fboxsep=\z@\fboxrule=\AMC@boxedrule%
  \AMC@setcolors@{#3}{#2}%
  \ifKV@AMCdim@cross\def\AMC@fillcolor@{white}\fi%
  \fcolorbox{\AMC@boxcolor@}{\AMC@fillcolor@}%
  {%
    \boxput*(0,0){%
      \ifAMC@draw@cross\AMC@crosschar\fi%
    }{%
    \vbox to \AMC@boxedheight{%
      \AMC@tracepos{#3}{#4}%
      \vfill%
      \hbox to \AMC@boxedwidth{\hfill%
        \AMC@smashcentered{\textcolor{\AMC@boxcolor@}{#1}}%
        \AMC@smashcentered{#2}%
        \hfill}\vfill}}%
    \AMC@tracepos{#3}{#4}}%
}
```

`\AMC@makeovalbox{`⟨*trace*⟩`}{`⟨*answer*⟩`}{`⟨*box*⟩`}` prepares an oval frame in the LaTeX box ⟨*box*⟩.

```
\newcommand\AMC@makeovalbox[3]{%
  \AMC@setcolors@{#1}{#2}%
```

```
  \ifKV@AMCdim@cross\def\AMC@fillcolor@{white}\fi%
  \AMC@save@box{#3}{%
    \begin{tikzpicture}%
      \useasboundingbox (-0.5\AMC@boxedwidth-0.5\AMC@boxedrule,0.5\AMC@boxedheight+0.5\AMC@boxedrule)
      rectangle (0.5\AMC@boxedwidth+0.5\AMC@boxedrule,-0.5\AMC@boxedheight-0.5\AMC@boxedrule);
      \draw[\AMC@boxcolor@,fill=\AMC@fillcolor@,line width=\AMC@boxedrule,rounded corners=\AMC@oval@radi
      (-0.5\AMC@boxedwidth,0.5\AMC@boxedheight)
      rectangle (0.5\AMC@boxedwidth,-0.5\AMC@boxedheight);
      \ifAMC@draw@cross
        \draw[\AMC@boxcolor@,line width=\AMC@crossrule]
        (-0.5\AMC@boxedwidth,0.5\AMC@boxedheight) -- (0.5\AMC@boxedwidth,-0.5\AMC@boxedheight)
        (0.5\AMC@boxedwidth,0.5\AMC@boxedheight) -- (-0.5\AMC@boxedwidth,-0.5\AMC@boxedheight);
      \fi
    \end{tikzpicture}}%
}
\newcommand\AMC@shapeprepare@oval{%
  \ifx\AMC@ovalbox@R\@undefined\else%
    \AMC@makeovalbox{1}{}{\AMC@ovalbox@R}%
    \AMC@makeovalbox{1}{\AMC@checkedbox}{\AMC@ovalbox@RF}%
    \AMC@makeovalbox{}{}{\AMC@ovalbox@}%
    \AMC@makeovalbox{}{\AMC@checkedbox}{\AMC@ovalbox@F}%
  \fi%
}
\newcommand\AMC@shape@oval[4]{%
  \AMC@setcolors@{#3}{#2}%
  \AMC@tracebox{#3}{#4}{\boxput*(0,0){%
      \AMC@smashcentered{\textcolor{\AMC@boxcolor@}{#1}}%
      \AMC@smashcentered{#2}%
    }{%
      \ifx\@empty#3\@empty%
        \ifx #2\AMC@checkedbox%
          \AMC@use@box{\AMC@ovalbox@F}%
        \else%
          \AMC@use@box{\AMC@ovalbox@}%
        \fi%
      \else%
        \ifx #2\AMC@checkedbox%
          \AMC@use@box{\AMC@ovalbox@RF}%
        \else%
          \AMC@use@box{\AMC@ovalbox@R}%
        \fi%
      \fi%
    }}%
}
\newcommand\AMC@shapeprepare@form{}
\newcommand\AMC@shape@form@base[5]{%
  \ifx #2\AMC@checkedbox%
    \def\AMC@shape@form@ticked{true}%
  \else%
    \def\AMC@shape@form@ticked{false}%
  \fi%
  \AMC@tracebox{#3}{#4}{%
    \CheckBox[checked=\AMC@shape@form@ticked,%
```

```
              checkboxsymbol=\ding{110},name={#5},%
              bordercolor=0 0 0,
              width=\AMC@boxedwidth,height=\AMC@boxedheight]{}{}%
  }%
}
\newcommand\AMC@shape@form[4]{%
  \AMC@shape@form@base{#1}{#2}{#3}{#4}{\the\AMCid@etud:#4}%
}
\newcommand\AMC@shapeprepare@none{}
\newcommand\AMC@shape@none[4]{ #1 }
```

\AMC@answerBox
\AMCchoiceLabel
CchoiceLabelFormat

Command `\AMC@answerBox` is the same as `\AMC@answerBox@`, but if ⟨*char*⟩ is empty, it is replaced by an arabic or alphabetical counter, depending on the use of the `digits` package option.

To use another way to label the choices boxes, the user can redefine the `\AMCchoiceLabel` macro, which takes as argument the name of the counter used to number the choices. One can for example use `\def\AMCchoiceLabel#1{\alph{#1}}` to ask for lowercase letters.

To write these labels with another font, size, or so, the user can redefine the `\AMCchoiceLabelFormat` macro, which takes as argument the label. One can for example get sans serif bold labels with `\def\AMCchoiceLabelFormat#1{{\textsf{\textsf{#1}}}}`.

```
\def\AMCchoiceLabel#1{%
  \ifAMC@inside@digit\arabic{#1}%
  \else\Alph{#1}\fi%
}
\def\AMCchoiceLabelFormat#1{#1}
\newcounter{AMC@ncase}
\setcounter{AMC@ncase}{0}
\newcommand\AMC@answerBox[4]{%
  \AMC@answerBox@{\ifx\@empty#1\@empty%
    \AMCchoiceLabel{AMC@ncase}%
    \else #1\fi}{#2}{#3}{#4}}
```

\AMCboxStyle

The dimensions of these box are managed by `\AMCboxDimensions{`⟨*sizes*⟩`}`, where ⟨*sizes*⟩ is a coma separated list of ⟨*name*⟩=⟨*dimension*⟩ constructs. Here, ⟨*name*⟩ can be `size` for the box size, `rule` for the box rule width, `down` for moving the box down, `color` for the box color and `outsidesep` for the distance between the box and the letter (when outside the box).

The ⟨*color*⟩ value given to `color` is a color that should be defined for the `xcolor` package. This color is used only in the case the box will be used for data capture: it is not used on the corrected answer sheet (`answers` or `indivanswers` package option), and not used on the subject part of an exam with a separate answer sheet (`separateanswersheet` package option).

The `\AMCboxColor{`⟨*color*⟩`}` command is defined as an alias to `\AMCboxStyle{color=`⟨*color*⟩`}`, and `\AMCboxDimensions` as an alias to `\AMCboxStyle`, for backward compatibility.

```
\newlength\AMC@boxedrule
\newlength\AMC@crossrule
\newlength\AMC@boxeddown
\newlength\AMC@boxedwidth
\newlength\AMC@boxedheight
\newlength\AMC@oval@radius
\newlength\AMC@outside@sep
\define@choicekey{AMCdim}{shape}{square,oval,form,none}{\def\AMC@shapename{#1}}
\define@key{AMCdim}{size}{\AMC@boxedwidth=#1\AMC@boxedheight=#1}
\define@key{AMCdim}{height}{\AMC@boxedheight=#1}
\define@key{AMCdim}{width}{\AMC@boxedwidth=#1}
```

```
\define@key{AMCdim}{rule}{\AMC@boxedrule=#1}
\define@key{AMCdim}{outsidesep}{\AMC@outside@sep=#1}
\define@key{AMCdim}{down}{\AMC@boxeddown=#1}
\define@key{AMCdim}{color}{\def\AMC@boxcolor{#1}}
\define@boolkey{AMCdim}{cross}[false]{}
\define@key{AMCdim}{crosschar}[\textbf{\textsf{X}}]{\def\AMC@crosschar{#1}}
\define@key{AMCdim}{crossrule}[1.5pt]{\AMC@crossrule=#1}
\def\AMC@shapeprepare{\csname AMC@shapeprepare@\AMC@shapename@ \endcsname}
\def\AMCboxStyle#1{%
  \setkeys{AMCdim}{#1}%
  \ifnum\AMC@boxedwidth<\AMC@boxedheight%
    \AMC@oval@radius=\AMC@boxedwidth\divide\AMC@oval@radius\tw@%
  \else%
    \AMC@oval@radius=\AMC@boxedheight\divide\AMC@oval@radius\tw@%
  \fi%
  \AMC@shapeprepare%
}
\AMCboxStyle{shape=square,size=2.5ex,down=.4ex,rule=.5pt,outsidesep=.1em,color=black,cross,crosschar,cro
\newcommand\AMCboxColor[1]{\AMCboxStyle{color=#1}}
\let\AMCboxDimensions=\AMCboxStyle
```

`\AMCboxOutsideLetter` `\AMC@box` `\AMC@formBox@` `\AMC@formBox` `\AMCoutsideLabelFormat`

Command `\AMC@box{`⟨*char*⟩`}{`⟨*answer*⟩`}` prints a box with character ⟨*char*⟩ inside, showing answer ⟨*answer*⟩ (`\AMC@checkedbox` to get a filled box), using global variables to identify the box (question and choice).

It calls `\AMC@formBox@{`⟨*char*⟩`}{`⟨*answer*⟩`}{`⟨*trace*⟩`}{`⟨*key*⟩`}` to actually render the box.

Command `\AMC@formBox` simply sets the first argument when empty before calling `\AMC@formBox@`.

The command `\AMCboxOutsideLetter{`⟨*box*⟩`}{`⟨*char*⟩`}` is called to print the box *and* the character ⟨*char*⟩ outside (and next to) it. The character is formatted using `\AMCoutsideLabelFormat` first: if you need bold characters, redifine it with `\def\AMCoutsideLabelFormat#1{\textbf{#1}}`

`\AMC@keyBox@` is used instead of `\AMCformBox@` when the text that corresponds to the answer is the letter/character inside the box itself (see `\AMCcodeGrid` and `\AMCnumericChoices`.

```
\def\AMCoutsideLabelFormat#1{#1}
\newcommand\AMCboxOutsideLetter[2]{#1\nobreak\hspace{.1em}\AMCoutsideLabelFormat{#2}}
\newif\ifAMC@printformoutside@%
\newcommand\ifAMC@printformoutside{%
  \AMC@printformoutside@false%
  \ifAMC@ensemble\ifAMC@outside@box%
    \ifAMCformulaire@dedans\AMC@printformoutside@true\fi%
    \ifAMC@zoneformulaire\AMC@printformoutside@true\fi%
  \fi\fi%
  \ifAMC@printformoutside@%
}
\newcommand\AMC@formBox@[4]{%
  \ifAMC@printformoutside% letter to be written outside the box
    \AMCboxOutsideLetter{\AMC@answerBox@{}{#2}{#3}{#4}}{#1}%
  \else%
    \AMC@answerBox@{#1}{#2}{#3}{#4}%
  \fi%
  \AMC@tracechar{#1}{#2}{#3}{#4}%
}
\newif\ifAMC@printkeyoutside@%
\newcommand\ifAMC@printkeyoutside{%
```

```
  \AMC@printkeyoutside@false%
  \ifAMC@ensemble%
    \ifAMC@outside@box\AMC@printkeyoutside@true\fi%
  \else%
    \ifAMC@inside@box\else\AMC@printkeyoutside@true\fi%
  \fi%
  \ifAMC@printkeyoutside@%
}
\newcommand\AMC@keyBox@[4]{%
  \ifAMC@printkeyoutside%
    \AMCboxOutsideLetter{\AMC@answerBox@{}{#2}{#3}{#4}}{#1}%
  \else%
    \AMC@answerBox@{#1}{#2}{#3}{#4}%
  \fi%
  \AMC@tracechar{#1}{#2}{#3}{#4}%
}
\newcommand\AMC@formBox[4]{%
  \AMC@formBox@{\ifx\@empty#1\@empty%
    \AMCchoiceLabel{AMC@ncase}%
    \else #1\fi}{#2}{#3}{#4}%
}
\newcommand{\AMC@box}[2]{%
  \ifAMC@ensemble%
    \ifAMC@zoneformulaire% for codes inside form sheet
      \protect\AMC@formBox{#1}{#2}{1}{case:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}%
    \else%
      \ifAMCformulaire@dedans% for answer boxes inside form sheet
        \protect\AMC@formBox{#1}{#2}{1}{case:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}%
      \else% outside form sheet: not to be read during data capture
        \AMC@formBox{#1}{#2}{1}{casequestion:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}%
    \fi\fi%
  \else% no separate sheet for answers: always read
    \ifAMC@inside@box%
      \AMC@formBox{#1}{#2}{1}{case:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}%
    \else%
      \AMC@formBox@{}{#2}{1}{case:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}%
    \fi%
  \fi%
}
```

4.8.4 Scoring zones

`\AMCscoreZone` The source file can define zones that will be used to print scores when annotating the completed answer sheets. The command `\AMCscoreZone{`⟨*zone*⟩`}` logs these zones positions on the page.

```
\newif\ifAMCsz@logged\AMCsz@loggedfalse
\newcommand{\AMCscoreZone}[1]{%
  \ifAMC@ensemble%
    \ifAMCformulaire@dedans%
      \AMC@tracebox{1}{score::\the\AMCid@quest,-1}{#1}%
    \else%
      \AMC@tracebox{1}{scorequestion::\the\AMCid@quest,-1}{#1}%
    \fi%
  \else%
```

```
    \AMC@tracebox{1}{score::\the\AMCid@quest,-1}{#1}%
  \fi%
  \ifAMCsz@logged\else%
    \AMCmessage{VAR:scorezones=1}%
    \global\AMCsz@loggedtrue%
  \fi%
}
```

4.8.5 Binary boxes

The package prints on each page some boxes that code (like binary digits) student sheet number, page number and a check number, so as to be read easily from scans after exam.

`\AMCid@checkmax` `\AMC@NCBetud` `\AMC@NCBpage` `\AMC@NCBcheck`

The check number is just decreased each page. Its maximum value is `\AMCid@checkmax`. The number of binary digits used to print student sheet number, page and check number are `\AMC@NCBetud`, `\AMC@NCBpage` and `\AMC@NCBcheck`. The number of the first page is `\AMC@premierecopie`.

The length of zone reserved for binary boxes is `\AMC@CBtaille`.

```
\def\AMCid@checkmax{60}
\def\AMC@NCBetud{12}
\def\AMC@NCBpage{6}
\def\AMC@NCBcheck{6}
\newlength{\AMC@CBtaille}\setlength{\AMC@CBtaille}{5cm}
\def\AMC@premierecopie{1}
```

`\AMC@binaryCode` The command `\AMC@binaryCode{`⟨*options*⟩`}{`⟨*n*⟩`}` prints boxes to represent the number ⟨*n*⟩ in its binary form. Options from ⟨*options*⟩ include:

`ndigits=`⟨*ndigits*⟩ for the number of digits to be shown.

`id=`⟨*id*⟩ for an ID of the number role (1 for the student number, 2 for the page number, 3 for the checking value).

`hsep=`⟨*hsep*⟩ for the space between boxes.

`style=`⟨*style*⟩ for some box style options.

`\AMCbin@one` and `\AMCbin@zero` print individual digit-boxes.

For example, `\AMC@binaryCode{ndigits=12}{367}` shows $367 = 000101101111_2$ using 12 boxes:

```
\newtoks\AMCbin@sequence
\newcount\AMCbin@number
\newcount\AMCbin@digit
\newcount\AMCbin@id
\newcount\AMCbin@did
\newcount\AMCbin@ndigits
\newdimen\AMCbin@hsep
\define@key{AMCbin}{ndigits}{\AMCbin@ndigits=#1}
\define@key{AMCbin}{id}{\AMCbin@id=#1}
\define@key{AMCbin}{hsep}{\AMCbin@hsep=#1}
\define@key{AMCbin}{style}[]{\def\AMCbin@style{#1}}
```

```
\def\AMCbin@one{%
  \ifnum\AMCbin@did>\z@%
  \hspace{\AMCbin@hsep}%
  \fi%
  \advance\AMCbin@did\@ne%
  \ifnum\AMCbin@id>0%
  \AMC@answerBox@{}{\AMC@checkedbox}{1}{chiffre:\the\AMCbin@id,\the\AMCbin@did}%
  \else%
  \AMC@answerBox@{}{\AMC@checkedbox}{1}{}%
  \fi}
\def\AMCbin@zero{%
  \ifnum\AMCbin@did>\z@%
  \hspace{\AMCbin@hsep}%
  \fi%
  \advance\AMCbin@did\@ne%
  \ifnum\AMCbin@id>0%
  \AMC@answerBox@{}{}{1}{chiffre:\the\AMCbin@id,\the\AMCbin@did}%
  \else%
  \AMC@answerBox@{}{}{1}{}%
  \fi}
\newcommand{\AMC@binaryCode}[2]{%
\setkeys{AMCbin}{ndigits=1,hsep=0pt,style}\setkeys{AMCbin}{#1}%
\AMCbin@did=\z@%
{\AMCboxDimensions{shape=square,size=.32cm,down=0pt,rule=.2pt,cross=false}\expandafter\AMCboxDimensions\
\AMCbin@digit=\z@%
\loop%
\ifnum\AMCbin@number>\z@%
\advance\AMCbin@digit\@ne%
\ifodd\AMCbin@number\AMCbin@sequence=\expandafter{\expandafter\AMCbin@one\the\AMCbin@sequence}%
\else\AMCbin@sequence=\expandafter{\expandafter\AMCbin@zero\the\AMCbin@sequence}\fi%
\divide\AMCbin@number\tw@%
\repeat%
\loop\relax%
\ifnum\AMCbin@digit<\AMCbin@ndigits\advance\AMCbin@digit\@ne%
\AMCbin@sequence=\expandafter{\expandafter\AMCbin@zero\the\AMCbin@sequence}\repeat%
\the\AMCbin@sequence%
\ifnum\AMCbin@digit>\AMCbin@ndigits\PackageError{automultiplechoice}{Too low AMC@NCB value (got \the\AMC
}}
```

The commands `\AMCbin@begin` and `\AMC@binaryBoxes` are now unused and are defined for backward compatibility.

```
\def\AMCbin@begin#1{\setkeys{AMCbin}{id=#1}}
\newcommand{\AMC@binaryBoxes}[2][1]{%
\AMC@binaryCode{ndigits=#1}{#2}%
}
```

4.9 Checking Environment

`\AMCcurrentenv` Sets the current environment as document.

```
\def\AMCcurrentenv{document}
```

`\AMCif@env` Checks for the current environment.

```
\def\AMCif@env#1{
```

```
    \def\AMC@tempenv{#1}%
    \ifx\AMC@tempenv\AMCcurrentenv
        \expandafter\@firstoftwo
    \else
        \expandafter\@secondoftwo
    \fi
}
```

4.10 Handling groups of questions

The package allows to handle groups of questions, so as to be able to shuffle them before printing them to the sheets.

`\nouveaugroupe` `\element` Command `\nouveaugroupe{`⟨*group-name*⟩`}{`⟨*n*⟩`}` creates a new (empty) group with name ⟨*group-name*⟩ (argument ⟨*n*⟩ is present only for compatibility reasons and is ignored). Command `\element{`⟨*group-name*⟩`}{`⟨*text*⟩`}` adds to group ⟨*group-name*⟩ a new element that contains ⟨*text*⟩. ⟨*text*⟩ can be a `question` environment, ore two successive `question`s to be kept together, or anything else. Calling command `\nouveaugroupe` is not compulsory, as `\element` calls it if necessary.

```
\newcount\AMCtok@k
\newcount\AMCtok@max
\newcount\AMCtok@size
\newcommand{\nouveaugroupe}[2]{%
  \expandafter\ifx\csname #1@k\endcsname\relax%
    \expandafter\newcount\csname #1@k\endcsname%
    \expandafter\newcount\csname AMC#1@j\endcsname%
    \csname #1@k\endcsname=\z@\relax%
    \csname AMC#1@j\endcsname=\z@\relax%
    \setgroupmode{#1}{\AMCdefault@groupmode}%
  \fi%
}
\newcommand\AMC@prepare@element[1]{%
  \nouveaugroupe{#1}{}%
  \global\advance\csname #1@k\endcsname\@ne\relax%
  \AMCtok@k=\csname #1@k\endcsname%
  \expandafter\ifx\csname #1@\romannumeral\AMCtok@k\endcsname\relax%
    \expandafter\newtoks\csname #1@\romannumeral\AMCtok@k\endcsname\fi%
}
\newcommand{\element}[2]{%
  \AMC@prepare@element{#1}%
  \global\csname #1@\romannumeral\AMCtok@k\endcsname={#2}%
}
```

`\setgroupmode` `\setdefaultgroupmode` Command `\setgroupmode{`⟨*group-name*⟩`}{`⟨*mode*⟩`}` sets the group mode to ⟨*mode*⟩ for group ⟨*group-name*⟩. This mode setup the behaviour of `\insertgroup` and `\copygroup` for this group:

1. With mode `fixed`, group's elements will be taken from the beginning.
2. With mode `cyclic`, the elements will be taken from the group following the last call group's use, recycling if necessary.
3. Mode `withreplacement` is the same as `fixed`, but the group is shuffled before each use.

4. Mode `withoutreplacement` is like `cyclic`, adding some shuffling when comming back to the beginning of the group.

The command `\setdefaultgroupmode{`⟨*mode*⟩`}` sets the group mode to be used for the following created groups (a group is created at the first `\element{`⟨*group*⟩`}` call). When no `\setdefaultgroupmode` is used, `fixed` is the default mode.

```
\def\AMCdefault@groupmode{fixed}
\newcommand{\setdefaultgroupmode}[1]{\def\AMCdefault@groupmode{#1}}
\newcommand{\setgroupmode}[2]{%
  \expandafter\ifx\csname AMCgrouppre@#2\endcsname\relax%
    \PackageError{automultiplechoice}{Unknown group mode for #1 : #2}%
       {You asked to set group '#1' mode to '#2',
         but '#2' is not a valid group mode}%
  \else%
    \expandafter\global\expandafter\def\csname AMC#1@mode\endcsname{#2}%
  \fi%
}
```

The functions `\AMCgrouppre@xxx{`⟨*group-name*⟩`}{`⟨*n*⟩`}{`⟨*i*⟩`}` are called before using ⟨*n*⟩ elements from group ⟨*group-name*⟩ starting from index ⟨*i*⟩ (negative value for ⟨*i*⟩ stands for the current value of the group index), either with `\insertgroup` or `\copygroup`.

For mode **fixed**, the group index is set to ⟨*i*⟩, or 0 if ⟨*i*⟩ is negative (take elements from the beginning).

```
\newcommand{\AMCgrouppre@fixed}[3]{%
  \ifnum#3<\z@%
    \csname AMC#1@j\endcsname=\z@%
  \else%
    \csname AMC#1@j\endcsname=#3%
  \fi%
}
```

For mode **withreplacement**, the group is shuffled and the group index is set to ⟨*i*⟩ or 0 (take elements from the beginning) if negative.

```
\newcommand{\AMCgrouppre@withreplacement}[3]{%
  \ifnum#3<\z@%
    \csname AMC#1@j\endcsname=\z@%
  \else%
    \csname AMC#1@j\endcsname=#3%
  \fi%
  \shufflegroup{#1}%
}
```

For mode **withoutreplacement**, the group index is set to ⟨*i*⟩, or left unchanged if ⟨*i*⟩ is negative. If there is not enough elements left in the group, the elements before the index and the elements after the index are shuffled.

```
\newcount\AMC@imax
\newcommand{\AMCgrouppre@withoutreplacement}[3]{%
  \ifnum#3<\z@%
  \else%
    \csname AMC#1@j\endcsname=#3%
  \fi%
  \ifnum\AMCtok@ik=\AMCloop@k%
    \AMCtok@ik=\z@%
  \fi%
```

```
  \ifnum\AMCtok@ik=\z@%
    \shufflegroup{#1}%
  \else%
    \AMC@imax=\AMCloop@k%
    \advance\AMC@imax -#2\relax%
    \ifnum\AMCtok@ik>\AMC@imax%
      \shufflegroupslice{#1}{\@ne}{\AMCtok@ik}%
      \ifnum\AMCtok@ik<\AMCloop@k%
        \advance\AMCtok@ik\@ne%
        \shufflegroupslice{#1}{\AMCtok@ik}{\AMCloop@k}%
      \fi%
    \fi%
  \fi%
}
```

For mode **cyclic**, nothing has to be done, except setting the group index if non-negative.

```
\newcommand{\AMCgrouppre@cyclic}[3]{%
  \ifnum#3<\z@%
  \else%
    \csname AMC#1@j\endcsname=#3%
  \fi%
}
```

The function `\AMCgroup@pre{`⟨*mode*⟩`}{`⟨*group-name*⟩`}{`⟨*n*⟩`}{`⟨*i*⟩`}` calls the right `\AMCgrouppre@`xxx command.

```
\newcommand{\AMCgroup@pre}[4]{%
  \csname AMCgrouppre@#1\endcsname{#2}{#3}{#4}%
}
```

`\shufflegroup` `\insertgroup` `\insertgroupfrom`

Command `\shufflegroup{`⟨*group-name*⟩`}` shuffles the elements of group ⟨*group-name*⟩, and `\shufflegroupslice{`⟨*group-name*⟩`}{`⟨*a*⟩`}{`⟨*b*⟩`}` shuffles elements ⟨*a*⟩ to ⟨*b*⟩ from group ⟨*group-name*⟩. It can be called at each student sheet in order to get different student sheets and avoid cheating.

Command `\insertgroup[`⟨*n*⟩`]{`⟨*groupname*⟩`}` inserts all the elements of group ⟨*groupname*⟩, or only the first ⟨*n*⟩ elements if ⟨*n*⟩ is given. `\insertgroupfrom[`⟨*n*⟩`]{`⟨*groupname*⟩`}{`⟨*i*⟩`}` inserts all the elements of group ⟨*groupname*⟩ starting from index ⟨*i*⟩ (the index of the first element is 0), or only the first ⟨*n*⟩ elements if ⟨*n*⟩ is given.

```
\newcommand{\shufflegroup}[1]{%
  \ifAMC@shuffleG{\AMC@shuffletoks{\number\csname #1@k\endcsname}{#1@}}\fi%
}
\newcommand{\shufflegroupslice}[3]{%
  \ifAMC@shuffleG{\AMC@shuffletoks[#2]{#3}{#1@}}\fi%
}
\newcount\AMCtok@ik
\newcount\AMCloop@k
\newcommand{\AMCgrouploop@prep}[3]{%
  \AMCtok@size=#1\relax%
  \ifAMC@fullGroups\AMCtok@size=\m@ne\fi%
  \ifnum\AMCtok@size<\z@%
    \AMCtok@size=\csname #2@k\endcsname%
  \fi%
  \AMCtok@ik=\csname AMC#2@j\endcsname%
  \AMCloop@k=\csname #2@k\endcsname%
  \expandafter\ifx\csname AMC#2@mode\endcsname\relax%
    \PackageError{automultiplechoice}{No group mode for #2}%
```

```
      {No mode has been defined for group '#2'. This should not occur...}%
  \fi%
  \AMCgroup@pre{\csname AMC#2@mode\endcsname}{#2}{\the\AMCtok@size}{#3}%
}
\newcommand{\AMCgrouploop@next}[1]{%
  \global\advance\csname AMC#1@j\endcsname\@ne\relax%
  \expandafter\ifnum\csname AMC#1@j\endcsname>\AMCloop@k\relax%
    \global\csname AMC#1@j\endcsname=\@ne%
  \fi%
  \AMCtok@ik=\csname AMC#1@j\endcsname%
  \advance\AMCtok@size\m@ne%
}
\newcommand{\insertgroupfrom}[3][-1]{%
  \ifnum#1=0%
  \else%
    \AMCgrouploop@prep{#1}{#2}{#3}%
    {\loop%
      \AMCgrouploop@next{#2}%
      {\the\csname #2@\romannumeral\AMCtok@ik\endcsname}%
    \ifnum\AMCtok@size>\z@\repeat}%
  \fi%
}
\newcommand{\insertgroup}[2][-1]{%
  \insertgroupfrom[#1]{#2}{-1}%
}
```

`\cleargroup` `\copygroup` `\copygroupfrom`

The commands `\cleargroup` and `\copygroup` can also be used to make more complex questions combinations in the exams, allowing for example to ask the package to shuffle 3 questions taken at random from group `groupa` and 5 questions taken at random from group `groupb`.

`\cleargroup{`⟨*group*⟩`}` clears the group ⟨*group*⟩, ereasing all of its elements.

`\copygroup[`⟨*n*⟩`]{`⟨*from*⟩`}{`⟨*to*⟩`}` copies ⟨*n*⟩ elements from group ⟨*from*⟩ to group ⟨*to*⟩. If optional parameter ⟨*n*⟩ is not given, all the questions from group ⟨*from*⟩ are copied. `\copygroupfrom[`⟨*n*⟩`]{`⟨*from*⟩`}{`⟨*to*⟩`}{`⟨*i*⟩`}` copies ⟨*n*⟩ elements from group ⟨*from*⟩ to group ⟨*to*⟩, starting from element at index ⟨*i*⟩ (the index of the first element is 0). If optional parameter ⟨*n*⟩ is not given, all the questions from group ⟨*from*⟩ are copied.

See section 3.4 for an illustration for these commands.

```
\newcommand{\cleargroup}[1]{%
  \nouveaugroupe{#1}{}%
  \csname #1@k\endcsname=\z@\relax%
  \csname AMC#1@j\endcsname=\z@\relax%
}
\newcommand{\copygroupfrom}[4][-1]{%
  \ifnum#1=0%
  \else%
    \AMCgrouploop@prep{#1}{#2}{#4}%
    {\loop%
      \AMCgrouploop@next{#2}%
      \AMC@prepare@element{#3}%
      \global\csname #3@\romannumeral\AMCtok@k\endcsname=\csname #2@\romannumeral\AMCtok@ik\endcsname%
    \ifnum\AMCtok@size>\z@\repeat}%
  \fi%
}
\newcommand{\copygroup}[3][-1]{%
```

```
  \copygroupfrom[#1]{#2}{#3}{-1}%
}
```

4.11 Questions

To manage multiple choice questions, first set some counters and token registers to handle answers. Token registers `\reponse@i`, `\reponse@ii` and so on will be used for answers – we restrict the number of answers of a single questions to `\AMCload@counter` = 199.

```
\newcount\AMCrep@count
\newcount\AMCrep@nn
\newcount\AMCrep@nnmax
\AMCload@counter=199
\@whilenum\AMCload@counter>0\do{%
  \expandafter\newtoks\csname reponse@\romannumeral\AMCload@counter\endcsname%
  \advance\AMCload@counter\m@ne%
}
```

`\AMCload@reponse` `\AMCrien@deux`

Command `\AMCload@reponse{`⟨*n*⟩`}{`⟨*text*⟩`}` will be used to add answer number ⟨*n*⟩ with text ⟨*text*⟩ (⟨*text*⟩ will include the box to be ticked and all the layout commands) to the set of answers (in a token register `\reponse@`*xxx* – counter `\AMCload@counter` keeps track of the number of answers), in order to shuffle them when all answers will be loaded.

When answers are not to be shuffled, command `\AMCrien@deux{`⟨*n*⟩`}{`⟨*text*⟩`}` will be used instead, only printing ⟨*text*⟩.

```
\newcommand\AMCload@reponse[2]{%
  \global\advance\AMCload@counter\@ne\relax%
  \global\csname reponse@\romannumeral\AMCload@counter\endcsname%
  =\expandafter{\expandafter\AMCrep@count\expandafter=#2 #1}%
}
\newcommand\AMCrien@deux[2]{#1}
```

`\shuffle@it` `\AMCdump@reponses`

After loading all answers, commands `\shuffle@it` will be used to shuffle them, and `\AMCdump@reponses` to print them.

```
\def\shuffle@it{\AMC@shuffletoks{\number\AMCload@counter}{reponse@}}
\newcount\AMCnum@questions
\newcommand\AMCdump@reponses{%
  \global\AMCnum@questions=\AMCload@counter%
  \@whilenum\AMCload@counter>0\do{%
    \the\csname reponse@\romannumeral\AMCload@counter\endcsname%
    \advance\AMCload@counter\m@ne}}
```

4.11.1 Managing answers

`\lastchoices` `\AMCrep@init` `\AMC@fin@rep`

Command `\AMCrep@init{`⟨*mode*⟩`}` is called for each question before reading answers. ⟨*mode*⟩ is `r` for suffled answers, and `o` if answers are not to be shuffled. It sets the number of answers counter to zero, and calls `\AMCrep@o` or `\AMCrep@r` depending on ⟨*mode*⟩. These commands sets `\AMCload@@reponse` and `\AMCrep@fini` that will be called for each answer and after the last answer respectively, depending on ⟨*mode*⟩:

- If ⟨*mode*⟩=`r`, `\AMCload@@reponse` is `\AMCload@reponse` (loads answer to token register) and `\AMCrep@fini` calls `\shuffle@it` and `\AMCdump@reponses`;
- If ⟨*mode*⟩=`o`, `\AMCload@@reponse` is `\AMCrien@deux` (prints answer directly) and `\AMCrep@fini` does nothing.

Command \lastchoices is called before giving answers that are to be printed at the end (even when shuffling answers). It closes the answers list calling \AMCrep@fini and opens another one in ordered mode. Note that it also saves the value of \AMCrep@count, which is the number of the current answer among all answers given in the subject source for the current question.

Command \AMC@fin@rep is to be called after the last answer: it adds a "None of these answers are correct." answer if necessary (package option completemulti) with answer number zero, and calls \AMCrep@fini.

```
\newcommand\AMCrep@init[1]{%
  \ifAMC@ordre\AMCrep@o\else%
    \csname AMCrep@#1\endcsname\fi\AMCload@counter=\z@}
\newcommand\AMCrep@o{%
  \def\AMCload@@reponse{\AMCrien@deux}\def\AMCrep@fini{}}
\newcommand\AMCrep@r{%
  \def\AMCload@@reponse{\AMCload@reponse}%
  \def\AMCrep@fini{\shuffle@it\AMCdump@reponses}}
\newcount\AMCrep@@count
\newcommand\lastchoices{%
  \AMCrep@@count=\AMCrep@count%
  \AMCrep@fini\AMCrep@init{o}%
  \AMCrep@count=\AMCrep@@count}
\newcommand\@aucune{\emph{\AMC@loc@none}}
\newcommand\AMC@fin@rep{%
  \ifAMCcomplete@multi\ifAMCtype@multi%
    \lastchoices\AMCrep@count=-1%
    \ifAMCune@bonne\wrongchoice{\@aucune}\else%
      \ifAMC@postcorrect\wrongchoice{\@aucune}\else\correctchoice{\@aucune}\fi%
    \fi\fi\fi\AMCrep@fini}
```

4.11.2 Separate answer sheet

This package needs some memory to print questions/answers boxes again on a separate answer sheet.

\AMCformQuestion
\AMCformAnswer

First define commands that will announce questions and answers on the separate answer sheet (these commands can be modified by the user): \AMCformQuestion{⟨*number*⟩} is responsible for announcing question, and \AMCformAnswer{⟨*box*⟩} is responsible for printing the box to be ticked, given as argument ⟨*box*⟩.

Commands \AMCformQuestionA and \AMCformAnswerA set up counter \AMC@ncase value before calling their counterparts.

```
\def\AMCformBeforeQuestion{\vspace{\AMCformVSpace}\par}
\def\AMCformAfterQuestion{\ifAMC@asqbloc\egroup\fi}
\def\AMCformQuestion#1{\AMC@loc@qf{#1}}
\def\AMCformQuestionN{\AMCformQuestion{\AMC@qaff}}
\def\AMCformQuestionA{%
  \setcounter{AMC@ncase}{0}%
  \AMCformBeforeQuestion%
  \ifAMC@asqbloc\vbox\bgroup\fi%
  \ifx\@empty\AMC@sza@callout\@empty\else%
    \csname\AMC@sza@callout\endcsname%
  \fi%
  \AMCformQuestionN%
  \ifx\@empty\AMC@sza@callin\@empty\else%
```

```
    \csname\AMC@sza@callin\endcsname%
  \fi%
}
\def\AMCformAnswer#1{\hspace{\AMCformHSpace} #1}
\def\AMCformAnswerA#1{\addtocounter{AMC@ncase}{1}\AMCformAnswer{#1}}
```

C@mem@add@ifneeded
\AMCformBegin
\AMCform
\AMCformS

These are commands to manage memory for separate answer sheet. `\AMC@mem@add@ifneeded{`⟨*code*⟩`}` adds ⟨*code*⟩ to this memory. `\AMC@mem@answer{`⟨*code*⟩`}` adds to memory answer code ⟨*code*⟩, and `\AMC@mem@openQuestion` adds to memory question code to announce current question.

The command `\AMCformBegin` defines the beginning of the separate answer sheet for the current student sheet, and `\AMCform` prints the whole memory: questions and answers boxes.

`\AMCformS` is a `\AMCform` variant that does not clear the list of answer boxes. It can be used to make the same exact subject for all students, displaying the questions before (outside) `onecopy`, so that `onecopy` contains only the answer sheet.

```
\ExplSyntaxOn

\prg_set_conditional:Nnn \amc_if_separate_question: { p , T } {
  \ifAMC@ensemble
    \ifAMC@zoneformulaire
      \prg_return_false:
    \else
      \prg_return_true:
    \fi
  \else
    \prg_return_false:
  \fi
}
\cs_new_eq:NN \AMC@if@separate@question \amc_if_separate_question:T

\int_new:N \amc_memory_elts_count

\cs_new:Nn \amc_clear_memory: { \int_gzero:N \amc_memory_elts_count }
\cs_new_eq:NN \AMC@mem@clear \amc_clear_memory:

\cs_new:Npn \amc_memory_elt_i:n #1 {
  amc_memory_elts_ \int_to_alph:n { #1 }
}
\cs_new:Nn \amc_memory_current_elt: {
  \amc_memory_elt_i:n \amc_memory_elts_count
}
\cs_new:Npn \amc_memory_vars_i:n #1 {
  amc_memory_vars_ \int_to_alph:n { #1 }
}
\cs_new:Nn \amc_memory_current_vars: {
  \amc_memory_vars_i:n \amc_memory_elts_count
}

\cs_new:Nn \amc_add_memory_elt: {
  \int_gincr:N \amc_memory_elts_count
  \tl_gclear_new:c { \amc_memory_current_elt: }
  \tl_gclear_new:c { \amc_memory_current_vars: }
}
```

```
\cs_new_eq:NN \AMC@mem@next \amc_add_memory_elt:

\cs_new:Npn \amc_add_to_memory:n #1 {
  \tl_gput_right:cn { \amc_memory_current_elt: } { #1 }
}
\cs_new_eq:NN \AMC@mem@add \amc_add_to_memory:n

\cs_new:Npn \amc_add_to_vars:n #1 {
  \tl_gput_right:cn { \amc_memory_current_vars: } { #1 }
}
\cs_new_eq:NN \AMC@mem@addvar \amc_add_to_vars:n

\cs_new:Npn \amc_add_qidaffname:nnn #1#2#3 {
  \amc_add_to_vars:n {\AMCid@quest=#1\setcounter{AMCquestionaff}{#2}%
    \global\def\AMCid@name{#3}}
}
\cs_generate_variant:Nn \amc_add_qidaffname:nnn { xxx }
\cs_new_eq:NN \AMC@mem@qidaffname \amc_add_qidaffname:xxx

\cs_new:Npn \amc_mem_elt_cat:n #1 {
  \amc_add_to_vars:n { \def\AMCmem@elt@cat{ #1 } }
}
\cs_generate_variant:Nn \amc_mem_elt_cat:n { x }
\cs_new_eq:NN \AMC@mem@category \amc_mem_elt_cat:x

\cs_new:Npn \amc_add_aid:n #1 {
  \amc_add_to_memory:n {\AMCrep@count=#1}
}
\cs_generate_variant:Nn \amc_add_aid:n { x }
\cs_new_eq:NN \AMC@mem@aid \amc_add_aid:x

\cs_new:Npn \amc_if_category_is_p:n #1 {
  \str_if_eq_p:on { \AMCmem@elt@cat } { #1 }
}
\cs_new:Npn \amc_use_memory:n #1 {
  \int_step_inline:nnnn { 1 } { 1 } \amc_memory_elts_count {
    \def\AMCmem@elt@cat{ plain }
    \tl_use:c { \amc_memory_vars_i:n { ##1 } }
    \bool_if:nTF { #1 } {
      \tl_use:c { \amc_memory_elt_i:n { ##1 } }
    } { }
  }
}
\cs_new:Nn \amc_use_memory: { \amc_use_memory:n { \c_true_bool } }
\cs_new_eq:NN \AMC@mem@show \amc_use_memory:
\cs_new_eq:NN \AMC@mem@show@filter \amc_use_memory:n
\cs_new_eq:NN \AMCifcategory \amc_if_category_is_p:n

\ExplSyntaxOff
\newcommand\AMC@mem@add@ifneeded[1]{%
  \AMC@if@separate@question{%
    \AMC@mem@add{#1}%
  }%
```

```
}
\newcommand\AMC@mem@addsingle@ifneeded[2]{%
  \AMC@if@separate@question{%
    \AMC@mem@next%
    \AMC@mem@category{#2}%
    \AMC@mem@add{#1}%
  }%
}
\newcommand\AMC@mem@answer[1]{%
  \addtocounter{AMC@ncase}{1}%
  \AMC@if@separate@question{%
    \AMC@mem@aid{\the\AMCrep@count}%
    \AMC@mem@add{\AMCformAnswerA{#1}}%
  }%
}
\newcommand\AMC@mem@openQuestion{%
  \AMC@if@separate@question{%
    \AMC@mem@next%
    \AMC@mem@qidaffname{\the\AMCid@quest}{\arabic{AMCquestionaff}}{\AMCid@name}%
    \AMC@mem@add{\AMCformQuestionA}%
  }%
}
\def\AMCformBegin{%
  \AMC@zoneformulairetrue\setcounter{section}{0}%
  \ifAMC@ensemble\ifAMC@automarks\pagestyle{AMCpageFull}\fi\fi%
}
\newcommand\AMCform{%
  \ifAMC@ensemble\AMCformulaire@dedanstrue%
    \AMC@mem@show%
  \fi}
\newcommand\AMCformFilter[1]{%
  \ifAMC@ensemble\AMCformulaire@dedanstrue%
    \AMC@mem@show@filter{#1}%
  \fi}
\newif\ifAMC@keepmemory
\newcommand\AMCformS{%
  \ifAMC@ensemble\AMCformulaire@dedanstrue%
  \AMCmessage{BR=0}\AMC@mem@show%
  \global\AMC@keepmemorytrue%
  \fi}
```

`\AMCtableForm` `\AMCtableForm[`⟨*options*⟩`]` is a variant of `\AMCform` that displays the boxes as a multi-column table.

The argument ⟨*options*⟩ is a key-value list, where:

- `nanswers=`⟨*n*⟩ gives the numer of answers that are labeled in the table (defaults to the maximal number of answers of the subject).
- `ncols=`⟨*n*⟩ gives the number of columns to use (defaults to the largest possible number of columns).
- `idtext=`⟨*text*⟩ gives a text to be inserted before each question number.
- `idtitle=`⟨*text*⟩ gives a text to be used as a column title for question numbers.

- `columnsep=`⟨*dim*⟩ and `columnseprule=`⟨*dim*⟩ are passed to the `multicols` environment.
- `headers=`⟨*bool*⟩ tells if the headers with boxes letters are to be added at the top of the columns (defaults to true).
- `inside=`⟨*bool*⟩ tells if letters are to be written inside the boxes (defaults to false).

```
\newcounter{AMC@specsi}
\def\AMC@formspecs#1{%
  \noindent\hbox to \AMCformTH{\hspace*{\fill}\textbf{\AMC@tf@idtitle}\hspace{\AMCformHSpace}}%
  \setcounter{AMC@specsi}{0}%
  \loop%
    \addtocounter{AMC@specsi}{1}%
    \AMCformAnswer{\bf\AMCchoiceLabel{AMC@specsi}}%
  \ifnum\value{AMC@specsi}<#1\repeat%
}
\newcount\AMC@coli
\def\AMC@formcolspecs#1#2{{%
  \columnseprule=0pt%
  \begin{multicols}{#1}
  \AMC@coli=\z@%
  \@whilenum\AMC@coli<#1\do{
    \advance\AMC@coli\@ne%
    \AMC@formspecs{#2}\par
  }
  \end{multicols}
}}
\newcount\AMC@tf@ncols
\newdimen\AMCformTH
\newdimen\AMC@tfcolw
\newdimen\AMC@tfaw
\def\AMC@tf@idtext{}
\def\AMC@tf@idtitle{}
\newbox\AMC@tfbox
\define@key{AMCtf}{nanswers}[0]{\def\AMC@tf@nanswers{#1}}
\define@key{AMCtf}{ncols}[0]{\AMC@tf@ncols=#1}
\define@key{AMCtf}{idtext}{\def\AMC@tf@idtext{#1}}
\define@key{AMCtf}{idtitle}{\def\AMC@tf@idtitle{#1}}
\define@key{AMCtf}{idwidth}[0pt]{\AMCformTH=#1}
\define@key{AMCtf}{columnsep}[0.5em]{\columnsep=#1}
\define@key{AMCtf}{columnseprule}[0.5pt]{\columnseprule=#1}
\define@boolkey{AMCtf}{headers}[true]{}
\define@boolkey{AMCtf}{inside}[false]{}
\setkeys{AMCtf}{nanswers,ncols,idwidth,headers,inside}
\newcommand\AMCtableForm[1][]{%
  \setkeys{AMCtf}{columnsep,columnseprule}
  \setkeys{AMCtf}{#1}%
  \ifnum\AMC@tf@nanswers=0%
    \def\AMC@tf@nanswers{\the\AMCrep@nnmax}%
  \fi%
  \newdimen\AMC@tf@colW\AMC@tf@colW=\AMCformHSpace%
  \advance\AMC@tf@colW by \AMC@boxedwidth%
  \def\AMCformBeforeQuestion{\vspace{\AMCformVSpace}\par}%
  \def\AMCformQuestion##1{\noindent\hbox to \AMCformTH{\hspace*{\fill}\textbf{\AMC@tf@idtext{}##1}\hspac
```

```
  \def\AMCformAnswer##1{\hbox to \AMC@tf@colW{\hspace*{\fill}##1\hspace*{\fill}}}%
  \ifnum\AMCformTH=0%
    \setbox\AMC@tfbox=\hbox{\textbf{\AMC@tf@idtext{}999}}%
    \AMCformTH=\wd\AMC@tfbox%
  \fi%
  \ifnum\AMC@tf@ncols=0%
    \AMC@tfcolw=\AMC@tf@nanswers\AMC@tf@colW%
    \advance\AMC@tfcolw by \AMCformTH%
    \advance\AMC@tfcolw by \columnsep%
    \AMC@tfaw=\linewidth%
    \loop%
    \ifnum\AMC@tfaw>\AMC@tfcolw%
      \advance\AMC@tfaw by -\AMC@tfcolw%
      \advance\AMC@tf@ncols\@ne%
    \repeat%
  \fi%
  \ifKV@AMCtf@headers%
    \AMC@formcolspecs{\the\AMC@tf@ncols}{\AMC@tf@nanswers}%
    \vspace{\AMCformVSpace}\vspace{-2\multicolsep}
  \fi%
  {\raggedcolumns
    \begin{multicols}{\the\AMC@tf@ncols}
      \ifKV@AMCtf@inside\else\def\AMCchoiceLabel##1{}\fi%
      \AMCform
    \end{multicols}}
}
```

`\AMCsection` `\AMCsubsection` The `\AMCsection` and `\AMCsubsection` commands issue their standard counterparts (`\section` and `\subsection` with the same argument, both in the subject *and* in the separate answer sheet.

```
\newcommand{\AMCsectionNumbered}[1]{%
  \section{#1}\AMC@mem@addsingle@ifneeded{\section{#1}}{section}}
\newcommand{\AMCsubsectionNumbered}[1]{%
  \subsection{#1}\AMC@mem@addsingle@ifneeded{\subsection{#1}}{subsection}}
\newcommand{\AMCsectionStar}[1]{%
  \section*{#1}\AMC@mem@addsingle@ifneeded{\section*{#1}}{section}}
\newcommand{\AMCsubsectionStar}[1]{%
  \subsection*{#1}\AMC@mem@addsingle@ifneeded{\subsection*{#1}}{subsection}}
\def\AMCsection{\@ifstar\AMCsectionStar\AMCsectionNumbered}
\def\AMCsubsection{\@ifstar\AMCsubsectionStar\AMCsubsectionNumbered}
```

4.11.3 Formatting answers

`choices` (*env.*) `choiceshoriz` (*env.*) `choicescustom` (*env.*) `tikz-single` (*env.*) `tikz-multi` (*env.*) `\AMCBoxedAnswers` Answers have to be included in an environment `choices` (standard), `choiceshoriz` (answers on one line) or `choicescustom` (user defined) depending on the desired formatting.

Use `\AMCBoxedAnswers` to request all answers to be included in LaTeX boxes; this can be useful for example when using multicolumn answers formatting.

```
\def\AMCBoxedAnswers{\AMC@rbloctrue}
\newcommand{\start@Answers}{%
  \global\AMCrep@count=\z@%
  \global\AMCrep@nn=\z@%
}
\newenvironment{choices}[1][r]{%
  \def\une@rep{\AMCrep@itemize}%
```

```
  \ifAMC@rbloc\def\une@rep{\AMCrep@bloc}%
  \else\begin{itemize}\setlength{\itemsep}{\AMCinterIrep}\fi%
    \AMCrep@init{#1}}%
  {\AMC@fin@rep\ifAMC@rbloc\else\end{itemize}\fi}
\newenvironment{choiceshoriz}[1][r]{%
  \def\une@rep{\AMCrep@ligne}\AMCrep@init{#1}%
  \par\begin{center}}%
  {\AMC@fin@rep\end{center}}
\newenvironment{choicescustom}[1][r]{%
  \def\une@rep{\AMCrep@perso}\AMCrep@init{#1}%
  \AMCbeginAnswer\ignorespaces}%
  {\AMC@fin@rep\AMCendAnswer}
\newenvironment{tikz-single}[1][r]{
\def\une@rep{\AMCrep@tikz}\AMCrep@init{#1}%
\begin{tikzpicture}}{\AMC@fin@rep\end{tikzpicture}}
\newenvironment{tikz-multi}[1][r]{
\def\une@rep{\AMCrep@tikzmat}\AMCrep@init{#1}%
\begin{tikzpicture}[remember picture]}{\AMC@fin@rep\end{tikzpicture}}
```

\AMCrep@bloc
\AMCrep@tikz
\AMCrep@tikzmat
\AMCrep@itemize
\AMCrep@ligne
\AMCrep@perso

For each of these styles, a corresponding `\AMCrep@`*xxx*`{`⟨*box*⟩`}{`⟨*text*⟩`}` is defined, which will format the answer with a box given in ⟨*box*⟩ and text ⟨*text*⟩. `\AMCrep@bloc` is also defined and used in standard formatting when the user wants to put answers inside a LaTeX box.

```
\newcommand\AMCrep@bloc[2]{\AMC@mem@answer{#1}%
  \par%
  \ifAMC@textPos\vbox\bgroup\AMC@tracepos{1}{atext:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}\hbox\
    \noindent\begin{minipage}{\linewidth}%
      \begin{itemize}\item[#1] #2\end{itemize}\end{minipage}%
    \ifAMC@textPos\AMC@tracepos{1}{atext:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}\egroup\AMC@trac
  \vspace{\AMCinterBrep}}
 \newcommand\AMCrep@tikz[5]{\AMC@mem@answer{#1}\node[#4] (lab\thecsvrow) at (#3) {#2} node[#5] (box\thec
\newcommand\AMCrep@tikzmat[5]{\AMC@mem@answer{#1}\node[#5] (box\thecsvrow) at (#3) {#1} node[#4] (lab\th
\newcommand\AMCrep@itemize[2]{\AMC@mem@answer{#1}\item[#1] #2}
\newlength\AMChorizAnswerSep
\setlength{\AMChorizAnswerSep}{3em plus 4em}
\newlength\AMChorizBoxSep
\setlength{\AMChorizBoxSep}{1em}
\newcommand\AMCrep@ligne[2]{\AMC@mem@answer{#1}%
  \ifAMC@textPos%
  \mbox{\AMC@tracebox{1}{atext:\AMCid@name:\the\AMCid@quest,\the\AMCrep@count}{#1\hspace*{\AMChorizBoxSe
  \else%
  \mbox{#1\hspace*{\AMChorizBoxSep}#2}%
  \fi\hspace{\AMChorizAnswerSep}}
\newcommand\AMCrep@perso[2]{\AMC@mem@answer{#1}\AMCanswer{#1}{#2}}
```

\AMCbeginAnswer
\AMCendAnswer
\AMCanswer

The `custom` style will use user-defined commands to format answers: `\AMCbeginAnswer` is called once before answers, `\AMCanswer{`⟨*box*⟩`}{`⟨*text*⟩`}` is called for each answer (⟨*box*⟩ beeing the box to be ticked and ⟨*text*⟩ the text associated with the proposed answer), and `\AMCendAnswer` is called after all answers.

```
\def\AMCbeginAnswer{}
\def\AMCanswer#1#2{#1 #2}
\def\AMCendAnswer{}
```

\answer
\correctchoice
\wrongchoice

The commands `\correctchoice` and `\wrongchoice` are used inside `choices`-like environments

to give the proposed answers and specify if they are to be tocked by the students or not.

```
\newcommand{\next@Answer}{%
  \global\advance\AMCrep@count\@ne\relax%
  \global\advance\AMCrep@nn\@ne\relax%
  \ifnum\AMCrep@nn>\AMCrep@nnmax%
    \global\AMCrep@nnmax=\AMCrep@nn%
  \fi%
}
\newcommand{\correctchoice}[2][]{%
  \next@Answer%
  \ifAMC@calibration\AMCmessage{REP=\the\AMCrep@count:B}\fi%
  \global\AMCune@bonnetrue%
  \AMCload@@reponse{\une@rep{\ifAMC@correc\AMC@box{#1}{\AMC@checkedbox}%
      \else\AMC@box{#1}{}\fi}{#2}}{\the\AMCrep@count}\ignorespaces}
\newcommand{\wrongchoice}[2][]{%
  \next@Answer%
  \ifAMC@calibration\AMCmessage{REP=\the\AMCrep@count:M}\fi%
  \AMCload@@reponse{\une@rep{\AMC@box{#1}{}}{#2}}{\the\AMCrep@count}%
  \ignorespaces}
```

4.11.4 Score zones

\AMCscoreZone
oreZoneAnswerSheet

The position of the scores on the annotated answer sheets can be defined in the LaTeX source file using `\AMCsetScoreZone{`⟨*options*⟩`}` (or `\AMCsetScoreZoneAnswerSheet{`⟨*options*⟩`}` for the answer sheets when the separate answer sheet option is used).

First begin with some helpers: `\AMCemptybox{`⟨*width*⟩`}{`⟨*height*⟩`}{`⟨*depth*⟩`}` draws an empty box with specified dimensions, and `\AMCmarginNote{`⟨*note*⟩`}` (code from one of `sgmoye`'s comments on `tex.stackexchange.com`) prints a marginal note in the left or right margin, depending on current the position (usefull in `multicols` environment).

```
\newcommand{\AMCemptybox}[3]{{%
    \sbox0{}\wd0=#1\ht0=#2\dp0=#3\relax\box0}}
\newlength\AMC@mn@test
\newlength\AMC@mn@sep\AMC@mn@sep=4mm
\newlength\AMC@mn@leftmargin
\newlength\AMC@mn@rightmargin
\newcommand\AMCmarginNote[1]{%
  \begin{tikzpicture}[remember picture,overlay]%
    \coordinate (here) at (0,0);%
    \pgfextractx{\AMC@mn@test}{\pgfpointdiff{\pgfpointorigin}%
      {\pgfpointanchor{current page}{center}}}%
    \ifodd\thepage%
      \AMC@mn@leftmargin=\oddsidemargin%
      \AMC@mn@rightmargin=\evensidemargin%
    \else
      \AMC@mn@leftmargin=\evensidemargin%
      \AMC@mn@rightmargin=\oddsidemargin%
    \fi
    \ifdim\AMC@mn@test < 1cm%
      \draw (current page.east |- here)+(-\AMC@mn@rightmargin-1in+\AMC@mn@sep,0pt) node[anchor=text,alig
    \else%
      \draw (current page.west |- here)+(0cm,0pt) node[anchor=text,align=right,text width=\AMC@mn@leftma
    \fi%
```

```
  \end{tikzpicture}%
}
```

Define now different ways to place the score zone:

`none` nowhere

`question` right after the question heading

`margin` in the margin, using `marginpar` (this does not work with `multicols` environment)

`margins` in the left or right margin, depending on the current position (needs `tikz` package)

```
\newcommand{\AMC@sz@box}{\AMCemptybox{\AMC@sz@width}{\AMC@sz@height}{\AMC@sz@depth}}
%
\newcommand{\AMC@sz@callin@question}{\AMCscoreZone{\AMC@sz@box}}
%
\newcommand{\AMC@sz@callout@margin}{\hspace{0pt}\marginpar{\AMCscoreZone{\AMC@sz@box}}}
%
\newcommand{\AMC@sz@init@margins}{\PackageWarning{automultiplechoice}{Please run twice to get proper mar
\newcommand{\AMC@sz@callout@margins}{\hspace{0pt}\AMCmarginNote{\AMCscoreZone{\AMC@sz@box}}}
```

Let us now set up options handling.

```
\newlength\AMC@sz@width
\newlength\AMC@sz@height
\newlength\AMC@sz@depth
\def\AMC@sz@callout{}
\def\AMC@sz@callin{}
\define@key{AMCsz}{width}{\AMC@sz@width=#1}
\define@key{AMCsz}{height}{\AMC@sz@height=#1}
\define@key{AMCsz}{depth}{\AMC@sz@depth=#1}
\define@key{AMCsz}{calloutside}{\def\AMC@sz@callout{#1}}
\define@key{AMCsz}{callinside}{\def\AMC@sz@callin{#1}}
\define@choicekey{AMCsz}{position}{none,question,margin,margins}{%
  \ifcsname AMC@sz@callout@#1\endcsname%
    \def\AMC@sz@callout{AMC@sz@callout@#1}%
  \else%
    \def\AMC@sz@callout{}%
  \fi%
  \ifcsname AMC@sz@callin@#1\endcsname%
    \def\AMC@sz@callin{AMC@sz@callin@#1}%
  \else%
    \def\AMC@sz@callin{}%
  \fi%
  \ifcsname AMC@sz@init@#1\endcsname%
    \csname AMC@sz@init@#1\endcsname%
  \fi%
}
\newcommand{\AMCsetScoreZone}[1]{\setkeys{AMCsz}{#1}}
\AMCsetScoreZone{width=1.5em,height=1.5ex,depth=.5ex,position=none}
```

And do the same for `\AMCsetScoreZoneAnswerSheet`...

```
\newcommand{\AMC@sza@box}{\AMCemptybox{\AMC@sza@width}{\AMC@sza@height}{\AMC@sza@depth}}
%
\newcommand{\AMC@sza@init@none}{}
```

```
\newcommand{\AMC@sza@callout@none}{}
\newcommand{\AMC@sza@callin@none}{}
%
\newcommand{\AMC@sza@init@question}{}
\newcommand{\AMC@sza@callout@question}{}
\newcommand{\AMC@sza@callin@question}{\AMCscoreZone{\AMC@sza@box}}
%
\newcommand{\AMC@sza@init@margin}{}
\newcommand{\AMC@sza@callout@margin}{\hspace{0pt}\marginpar{\AMCscoreZone{\AMC@sza@box}}}
\newcommand{\AMC@sza@callin@margin}{}
%
\newcommand{\AMC@sza@init@margins}{\PackageWarning{automultiplechoice}{Please run twice to get proper ma
\newcommand{\AMC@sza@callout@margins}{\hspace{0pt}\AMCmarginNote{\AMCscoreZone{\AMC@sz@box}}}
\newcommand{\AMC@sza@callin@margins}{}
%
\newlength\AMC@sza@width
\newlength\AMC@sza@height
\newlength\AMC@sza@depth
\def\AMC@sza@callout{}
\def\AMC@sza@callin{}
\define@key{AMCsza}{width}{\AMC@sza@width=#1}
\define@key{AMCsza}{height}{\AMC@sza@height=#1}
\define@key{AMCsza}{depth}{\AMC@sza@depth=#1}
\define@key{AMCsza}{calloutside}{\def\AMC@sza@callout{#1}}
\define@key{AMCsza}{callinside}{\def\AMC@sza@callin{#1}}
\define@choicekey{AMCsza}{position}{none,question,margin,margins}{%
  \ifcsname AMC@sza@callout@#1\endcsname%
    \def\AMC@sza@callout{AMC@sza@callout@#1}%
  \else%
    \def\AMC@sza@callout{}%
  \fi%
  \ifcsname AMC@sza@callin@#1\endcsname%
    \def\AMC@sza@callin{AMC@sza@callin@#1}%
  \else%
    \def\AMC@sza@callin{}%
  \fi%
  \ifcsname AMC@sza@init@#1\endcsname%
    \csname AMC@sza@init@#1\endcsname%
  \fi%
}
\newcommand{\AMCsetScoreZoneAnswerSheet}[1]{\setkeys{AMCsza}{#1}}
\AMCsetScoreZoneAnswerSheet{width=1.5em,height=1.5ex,depth=.5ex,position=none}
\newcommand{\AMCnoScoreZone}{\AMCsetScoreZone{position=none}\AMCsetScoreZoneAnswerSheet{position=none}}
```

4.11.5 Formatting questions

\AMCquestionaff
\AMC@stepQuestion
\AMC@qaff

The counter \AMCquestionaff keeps track of the current question number. It can be redefined by the user, for example to print several questions without a number, and then print questions with a number starting at one.

\AMC@stepQuestion will increase this counter and \AMC@qaff will format the question number out.

```
\newcounter{AMCquestionaff}
\newcommand{\AMCnumero}[1]{\setcounter{AMCquestionaff}{#1}\addtocounter{AMCquestionaff}{-1}}
```

```
\AtBeginDocument{%
  \ifx\@skiphyperreftrue\@undefined%
    \expandafter\newif\csname if@skiphyperref\endcsname%
  \fi%
}
\newcommand\AMC@stepQuestion{\ifAMCquestionNumber\@skiphyperreftrue\refstepcounter{AMCquestionaff}\@skip
\newcommand\AMC@qaff{\arabic{AMCquestionaff}}
```

\AMCbeforeQuestion
\AMCbeginQuestion
\multiSymbole

The command \AMCbeforeQuestion opens a new question. The command \AMCbeginQuestion{⟨*n*⟩}{⟨*sign*⟩} will format the question header, where ⟨*n*⟩ is the question number and ⟨*sign*⟩ beeing \multiSymbole in case of a multiple question, and empty in case of a simple one. \AMCbeforeQuestion, \AMCbeginQuestion and \multiSymbole can be user-redifined.

```
\def\AMCbeforeQuestion{\ifAMC@qbloc\else\par\noindent\fi}
\def\AMCbeginQuestion#1#2{\noindent\AMC@loc@q{#1}{#2}%
  \ifx\@empty\AMC@sz@callin\@empty\hspace*{1em}\fi%
}
\def\multiSymbole{$\clubsuit$}
```

question (*env.*)
questionmult (*env.*)
riable-single (*env.*)
riable-multi (*env.*)
stionouverte (*env.*)
\ouverte@vs

Environment {question}{⟨*key*⟩} encloses a simple question (with one and only one correct choice) with associated unique key ⟨*key*⟩ and the proposed answers.

Environment {questionmult}{⟨*key*⟩} is the same for multiple questions (with none, one or several correct choices).

Environment {questionmultx}{⟨*key*⟩} is the same as questionmult, but with no use of \multiSymbole.

Environment {questionouverte}[⟨*width*⟩] is used for open questions (that won't be marked automatically!), with width given as an optional argument (defaults to 3 cm).

The command \AMCexternalQuestion{⟨*id*⟩}{⟨*maxscore*⟩} allows to declare a question that will be scored outside AMC, with a maximal score ⟨*maxscore*⟩. When you use this command, you can manage the question number and question text freely (AMC won't handle this).

```
\ifx\question\undefined\else\let\question\undefined\fi
\def\AMCnobloc{\AMC@qblocfalse}
\def\AMCbloc{\AMC@qbloctrue}
\newcommand\AMCstartWithQuestion[1]{%
  \global\def\AMCid@name{#1}\AMC@affecte{#1}{\AMCid@quest}%
  \ifAMC@calibration%
    \AMCmessage{Q=\the\AMCid@quest}%
    \immediate\write\AMC@XYFILE{\string\question{\the\AMCid@quest}{\AMCid@name}}%
  \fi%
  \start@Answers%
}
\newcommand\AMCexternalQuestion[2]{%
  \AMCstartWithQuestion{#1}%
  \ifAMC@calibration%
    \AMCmessage{B=MAX=#2}%
    \AMCmessage{MULT}%
    \AMCmessage{FQ}%
  \fi%
}
\newenvironment{question}[2][]{%
  \def\AMCcurrentenv{question}%
  \AMC@stepQuestion%
  \AMCstartWithQuestion{#2}%
```

```
  \AMCbeforeQuestion%
  \ifx\@empty\AMC@sz@callout\@empty\else%
    \csname\AMC@sz@callout\endcsname%
  \fi%
  \AMCtype@multifalse\ifAMC@qbloc\ifAMC@textPos\vbox\bgroup\AMC@tracepos{1}{qtext:#2:\the\AMCid@quest,0}
  \ifAMC@affichekeys\index{\texttt{#2}}\ifAMC@keysline[\texttt{#2}]\newline\fi\fi%
  \AMCbeginQuestion{\ifAMC@affichekeys\ifAMC@ensemble\AMC@qaff\ \fi\ifAMC@keysline\else[\texttt{#2}]\fi\
  \ifAMC@calibration\immediate\write\AMC@XYFILE{\string\questionnum{\the\AMCid@etud}{\the\AMCid@quest}{\
  \ifx\@empty\AMC@sz@callin\@empty\else%
    \csname\AMC@sz@callin\endcsname%
  \fi%
  \AMCformulaire@dedansfalse\setcounter{AMC@ncase}{0}%
  \AMC@mem@openQuestion}%
{\ifAMC@qbloc\end{minipage}\ifAMC@textPos\AMC@tracepos{1}{qtext:\AMCid@name:\the\AMCid@quest,0}\egroup\A
\newenvironment{questionmult}[1]{%
  \AMCune@bonnefalse\begin{question}[{{\multiSymbole}}]{#1}%
  \AMCtype@multitrue\ifAMC@calibration%
  \AMCmessage{MULT}\fi}%
{\end{question}}
\newif\ifSurveySingleAnswer\SurveySingleAnswerfalse
\newenvironment{variable-single}[2]
{\def\AMCbeginQuestion##1##2{}
\begin{question}{#1}\scoring{v=#2}\QuestionIndicative\ifSurveySingleAnswer\else\AMCmessage{MULT}\fi
\begin{tikz-single}[o]}
{\end{tikz-single}
\end{question}}
\newenvironment{variable-multi}[4]
{\def\AMCbeginQuestion##1##2{}
\begin{question}{#1}\scoring{v=#4}\QuestionIndicative\ifSurveySingleAnswer\else\AMCmessage{MULT}\fi
\begin{tikz-multi}[o]
\node[#3] (var) at (0,0) {#2};}
{\end{tikz-multi}
\end{question}}
\newenvironment{questionmultx}[1]{%
  \begingroup\def\multiSymbole{}\begin{questionmult}{#1}}%
{\end{questionmult}\endgroup}
\newdimen\ouverte@vs
\newenvironment{questionouverte}[1][3cm]{%
  \AMC@stepQuestion%
  \AMCtype@multifalse\ouverte@vs=#1%
  \ifAMC@qbloc\noindent\begin{minipage}{\linewidth}\fi%
  \AMCbeginQuestion{\AMC@qaff}{}}%
{\vspace*{\ouverte@vs}\ifAMC@qbloc\end{minipage}\vspace{3ex}\fi}
```

4.11.6 Explanations

\explain The command \explain{⟨*text*⟩} is used inside question-like environments to give the explanation for the answers of a question. The command \explaincontext{⟨*text*⟩} inserts its argument only in the corrected paper.

```
\newcommand{\explain}[1]{%
\ifAMC@correc%
  \AMCif@env{question}{\par\noindent{\AMC@loc@explain #1}}{\AMC@error@explain}\vspace{1ex}%
\else%
```

```
  \AMCif@env{question}{}{\AMC@error@explain}%
\fi%
}
\newcommand{\explaincontext}[1]{%
\ifAMC@correc%
#1%
\fi%
}
```

4.12 Scoring

`\scoring` `\scoringDefaultS` `\scoringDefaultM` `\QuestionIndicative`

Scoring strategies are simply transmitted to the `.amc` file for later analysis.

`\scoring{`⟨*scrore*⟩`}` details the scoring strategy for current question or current answer, `\scoringDefaultS{`⟨*score*⟩`}` and `\scoringDefaultM{`⟨*score*⟩`}` gives default scoring strategy for simple and multiple questions, and `\QuestionIndicative` tells that the current question is not no be taken into account in the global mark.

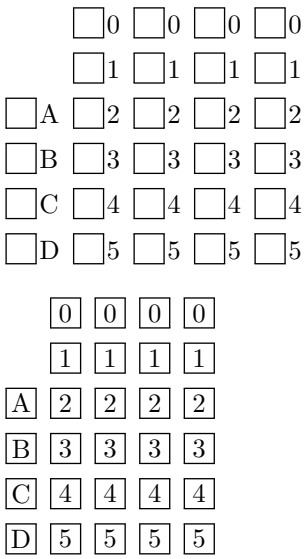
```
\def\scoring#1{\ifAMC@calibration\AMCmessage{B=#1}\fi\ignorespaces}
\def\scoringDefaultS#1{\ifAMC@calibration\AMCmessage{BDS=#1}\fi}
\def\scoringDefaultM#1{\ifAMC@calibration\AMCmessage{BDM=#1}\fi}
\def\QuestionIndicative{\ifAMC@calibration\AMCmessage{INDIC}\fi}
```

4.13 Numerical data

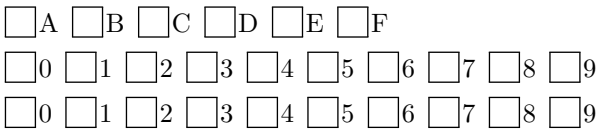
4.13.1 Codes

`\AMCcodeGrid` `\AMCcodeGridInt`

Students can code some numerical information (such as student number) through special questions, which can be formatted easily with the command `\AMCcodeGrid[`⟨*opts*⟩`]{`⟨*key*⟩`}{`⟨*descr*⟩`}`, where ⟨*key*⟩ is a key prefix and ⟨*descr*⟩ is a coma-separated list of character pools to offer. The characters entered by the student will be available through the questions ⟨*key*⟩`[1]`, . . . ,⟨*key*⟩`[`⟨*length(descr)*⟩`]`.
As an example,
`\AMCcodeGrid{code}{ABCD,012345,012345,012345,012345}` produces the opposite boxes (two results are show here: without or with `separateanswersheet` option), and trace positions of all the boxes in the `.xy` file with the `code` identifier: the first digit is represented by question with key `code[6]`, the second by question with key `code[5]`, and so on.
Positions of the boxes are logged in the `.xy` file, as shown in section 5.3 for the first set of boxes (without `separateanswersheet`, with digits outside boxes).



The "horizontal" version can also be considered using option `h`, especially with a small number of digits. See opposite for the result of `\AMCcodeGrid[h]{code}{ABCDEF,0123456789,0123456789}`.



The `\AMCcodeGridInt[`⟨*opts*⟩`]{`⟨*key*⟩`}{`⟨*n*⟩`}` is a shorcut for calling `\AMCcodeGrid` with ⟨*n*⟩ digits from 0 to 9. This allows to create grids for ⟨*n*⟩-digits integers easily.

These two commands supports the following options (given as a comma-separated list optional argument ⟨*opts*⟩):

- `vertical=true` or `false` to indicate the direction to be used (default is `true`);
- `h` is a shortcut for `vertical=false`;
- `v` is a shortcut for `vertical=true`;
- `top` to request top-aligned columns in vertical direction.
- `multi` for codes that are repeated on each page.

```
\newcount\AMC@chiffres
\newdimen\AMCcodeHspace\AMCcodeHspace=.5em
\newdimen\AMCcodeVspace\AMCcodeVspace=.5em
\newcommand\AMCcodeID@squarebrackets[2]{#1[#2]}
\newcommand\AMCcodeID@dot[2]{#1.#2}
\newcommand\AMCcodeID@@[1]{%
  \expandafter\def\expandafter\AMCcodeID\expandafter{\csname AMCcodeID@#1\endcsname}%
  \def\AMCcodeID@mode{#1}%
}
\AMCcodeID@@{squarebrackets}
\ExplSyntaxOn

\clist_new:N \amc_code_descr_clist
\seq_new:N \amc_code_digits_seq
\int_new:N \amc_code_digit_n_int
\bool_new:N \amc_code_vertical_bool
\bool_new:N \amc_code_top_bool
\bool_new:N \amc_code_multi_bool
\clist_new:N \amc__multi_clist

\cs_new:Npn \amc_code_init:N #1 {
  \def\AMCbeginQuestion##1##2{}
  \def\AMCbeforeQuestion{}
  \AMCnoScoreZone
  \AMCquestionNumberfalse
  \setlength{\parindent}{0pt}
  \AMCnobloc
  \int_set:Nn \amc_code_digit_n_int { \clist_count:N #1 }
}

\cs_new:Nn \amc_code_digit_init: {
  \QuestionIndicative
  \global\AMCrep@count=\z@
}

\cs_new:Npn \amc_code_digit:n #1 {
  \global\advance\AMCrep@count\@ne\relax
  \ifAMC@calibration\AMCmessage{ REP = \the\AMCrep@count : M }\fi
  \hbox{\AMC@keyBox@{#1}{}{1}{case : \AMCid@name : \the\AMCid@quest , \the\AMCrep@count}}
  \bool_if:NTF \amc_code_vertical_bool {
```

```
    \vspace{\AMCcodeVspace}
  }{
    \hspace{\AMCcodeHspace}
  }
}

\keys_define:nn { amccode } {
  vertical .bool_set:N = \amc_code_vertical_bool,
  vertical .initial:n = { true },
  vertical .default:n = { true },
  v .code:n = { \bool_set_true:N \amc_code_vertical_bool },
  h .code:n = { \bool_set_false:N \amc_code_vertical_bool },
  top .bool_set:N = \amc_code_top_bool,
  top .initial:n = { false },
  top .default:n = { true },
  multi .bool_set:N = \amc_code_multi_bool,
  multi .initial:n = { false },
  multi .default:n = { true }
}

\cs_new_nopar:Nn \amc_multi_report: {
  \ifAMC@calibration
  \immediate\write\AMC@XYFILE{\string\with{multi=\clist_use:Nn\amc__multi_clist{,}}}
  \fi
}
\cs_new_eq:NN \AMC@multi@report \amc_multi_report:
\int_new:N \amc_multi_count_int
\cs_new_nopar:Nn \amc_multi_clear: {
  \int_gzero:N \amc_multi_count_int
}
\cs_new_eq:NN \AMC@multiclear \amc_multi_clear:

\cs_new:Npn \amc_code_generate:nNn #1#2#3 {
  { \keys_set:nn { amccode } { #3 }
    \bool_if:NTF \amc_code_multi_bool {
      \clist_gset:Nn \amc__multi_clist { #1 }
    } {}
    \bool_if:NTF \amc_code_multi_bool { \int_gincr:N \amc_multi_count_int } {}
    \amc_code_init:N #2
    \clist_map_inline:Nn #2 { % iterates over 'digits'
      \begin{question}{
          \AMCcodeID{ #1 \bool_if:NTF
            \amc_code_multi_bool
            { * \int_use:N \amc_multi_count_int } {} }
          { \int_use:N \amc_code_digit_n_int }
        }
        \amc_code_digit_init:
        \seq_set_split:Nnn \amc_code_digits_seq {} { ##1 }
        \bool_if:NTF \amc_code_vertical_bool {
          \hspace{0pt}
          \bool_if:NTF \amc_code_top_bool { \vtop } { \vbox }
          \bgroup
        }{
```

```
          \hbox\bgroup
        }
        \seq_map_inline:Nn \amc_code_digits_seq {
          % iterates over available characters for 'digit'
          \amc_code_digit:n { ####1 }
        }
        \bool_if:NTF \amc_code_vertical_bool {
          \vspace{-\AMCcodeVspace}\egroup
          \hspace{\AMCcodeHspace}
        }{
          \egroup\vspace{\AMCcodeVspace}
          \par
        }
      \end{question}
      \int_decr:N \amc_code_digit_n_int
    }
  }
}

\cs_new:Npn \amc_code_generate:nnn #1#2#3 {
  \clist_set:Nn \amc_code_descr_clist { #2 }
  \amc_code_generate:nNn { #1 } \amc_code_descr_clist { #3 }
}
\cs_generate_variant:Nn \amc_code_generate:nnn { xxx }
\newcommand{\AMCcodeGrid}[3][]{
  \amc_code_generate:xxx { #2 } { #3 } { #1 }
}

\cs_new:Npn \amc_code_generate_integer:nnn #1#2#3 {
  \clist_clear:N \amc_code_descr_clist
  \prg_replicate:nn { #2 } { \clist_put_right:Nn \amc_code_descr_clist { 0123456789 } }
  \amc_code_generate:nNn { #1 } \amc_code_descr_clist { #3 }
}
\cs_generate_variant:Nn \amc_code_generate_integer:nnn { xxx }
\newcommand{\AMCcodeGridInt}[3][]{
  \amc_code_generate_integer:xxx { #2 } { #3 } { #1 }
}

\cs_new:Npn \amc_code_generate_integer_v:nn #1#2 {
  \amc_code_generate_integer:nnn { #1 } { #2 } { v }
}
\cs_new:Npn \amc_code_generate_integer_h:nn #1#2 {
  \amc_code_generate_integer:nnn { #1 } { #2 } { h }
}
\cs_generate_variant:Nn \amc_code_generate_integer_v:nn { xx }
\cs_generate_variant:Nn \amc_code_generate_integer_h:nn { xx }
\cs_new_eq:NN \AMCcode \amc_code_generate_integer_v:xx
\cs_new_eq:NN \AMCcodeH \amc_code_generate_integer_h:xx

\ExplSyntaxOff
```

4.13.2 Numerical questions

`\AMCnumericChoices` The command `\AMCnumericChoices{`⟨*correct*⟩`}{`⟨*options*⟩`}` can be used as a replacement for the `choices` environment when the questions asks for a numeric value to code on the answer sheet.

As an example,

```
\begin{question}{product}
  What is the value of $7\times 5$?
  \AMCnumericChoices{35}{digits=2,sign=false}
\end{question}
```

produces (in correction mode):

Question 11 What is the value of 7×5?

☐0 ☐1 ☐2 ■3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9
☐0 ☐1 ☐2 ☐3 ☐4 ■5 ☐6 ☐7 ☐8 ☐9

and these boxes are only shown on the separate answer sheet if the `separateanswersheet` option is used.

This command uses the `\AMCformatChoices{`⟨*showcommand*⟩`}{`⟨*hidecommand*⟩`}{`⟨*arg1*⟩`}{`⟨*arg2*⟩`}` command, that calls either ⟨*hidecommand*⟩`{`⟨*arg1*⟩`}{`⟨*arg2*⟩`}` if the `separateanswersheet` option is used and if we are currently in the question part (not in the answer sheet), or ⟨*showcommand*⟩`{`⟨*arg1*⟩`}{`⟨*arg2*⟩`}` when all the boxes are to be produced.

```
\newcommand\AMCformatChoices[4]{%
  \global\AMCrep@count=\z@%
  \AMC@if@separate@question{%
    \AMC@mem@add{\global\AMCrep@count=\z@%
                 #1{#3}{#4}}%
  }%
  \ifAMC@ensemble%
    #2{#3}{#4}%
    \AMCmessage{QPART}%
  \else%
    #1{#3}{#4}%
  \fi%
}
```

Some computation commands are now defined. The command `\amc_fp_decompose:NNn{`⟨*fp var*⟩`}{`⟨*int var*⟩`}{`⟨x⟩`}` sets ⟨*fp var*⟩ to be the *mantissa* and ⟨*int var*⟩ the *exponent* of the flaoting point number ⟨x⟩. For example, `\amc_fp_decompose:NNn\mant_fp\expo_int{123.456}` give the value `1.23456` to `\mant_fp` and `2` to `\expo_int` (because $123.456 = 1.23456 \times 10^2$).

The command `\amc_fp_to_digits:Nnnn{`⟨*clist*⟩`}`⟨x⟩⟨*n digits*⟩⟨*base*⟩ rounds the floating point number ⟨x⟩ and populates the comma separated list ⟨*clist*⟩ with its ⟨*n digits*⟩ digits in base ⟨*base*⟩. An error is issued if ⟨x⟩ would have required more digits.

```
\ExplSyntaxOn

\cs_generate_variant:Nn \tl_replace_once:Nnn { Nxn }

\tl_new:N \amc_ee_tl
\seq_new:N \amc_ee_seq
```

Note that with some versions of `l3fp-convert` (prior to 2017-09-18), `\fp_to_scientific` leads to a 'e' with catcode 12 (*other*). We convert it to catcode *letter* before splitting.

```
\group_begin:
\char_set_catcode_other:N E
\tex_lowercase:D
{
  \cs_new:Npn \amc_read_scientific:NNn #1 #2 #3 {
    \tl_set:Nn \amc_ee_tl { #3 }
    \tl_replace_once:Nxn \amc_ee_tl { E } { e }
    \seq_set_split:NnV \amc_ee_seq e \amc_ee_tl
    \fp_set:Nn #1 { \seq_item:Nn \amc_ee_seq 1 }
    \int_set:Nn #2 { \seq_item:Nn \amc_ee_seq 2 }
  }
}
\group_end:

\cs_generate_variant:Nn \amc_read_scientific:NNn { NNf, NNx }

\fp_new:N \amc_fulls_fp
\cs_new:Npn \amc_fp_decompose:NNn #1 #2 #3 {
  \fp_set:Nn \amc_fulls_fp { #3 }
```

Note that with some versions of `l3fp-convert`, the exponent part is omited for some values, so that we add `e 0`.

```
  \amc_read_scientific:NNx #1 #2
  { \fp_to_scientific:N \amc_fulls_fp e 0 }
}
\cs_generate_variant:Nn \amc_fp_decompose:NNn { NNx }

\fp_new:N \amc_num_mantissa_fp
\int_new:N \amc_num_exponent_int
\cs_new:Npn \amc_fp_n_significant_digits:Nnn #1 #2 #3 {
  \amc_fp_decompose:NNn \amc_num_mantissa_fp \amc_num_exponent_int
  { #2 }
  \fp_set:Nn #1
  { round(\amc_num_mantissa_fp * 10^((#3)-1)) }
  \fp_compare:nTF { abs(#1) >= 10^(#3) }
  {
    \fp_set:Nn #1 { #1 / 10 }
  } { }
}

\fp_new:N \amc_num_nsig_fp
\cs_new:Npn \amc_fp_show_n_significant_digits:nn #1 #2 {
  \amc_fp_n_significant_digits:Nnn \amc_num_nsig_fp { #1 } { #2 }
}
\cs_new_eq:NN \AMCsignificantDigits \amc_fp_show_n_significant_digits:nn

\cs_new:Npn \amc_fp_show_significant_digits: {
  \fp_use:N \amc_num_nsig_fp
}
\cs_new_eq:NN \AMCshowSignificantDigits \amc_fp_show_significant_digits:

\cs_new:Npn \amc_fp_n_digits:Nnn #1 #2 #3 {
```

```
  \fp_set:Nn #1
    { round((#2) * \amc_num_base_int^(#3)) }
}

\int_new:N \amc_todigits_int
\cs_new:Npn \amc_fp_to_digits:Nnnn #1 #2 #3 #4 {
  \clist_clear:N #1
  \int_set:Nn \amc_todigits_int { \fp_eval:n { abs(round(#2)) } }
  \prg_replicate:nn { #3 } {
    \clist_put_left:Nx #1 { \int_mod:nn \amc_todigits_int { #4 } }
    \int_set:Nn \amc_todigits_int
    { \int_div_truncate:nn \amc_todigits_int { #4 } }
  }
  \int_compare:nNnTF \amc_todigits_int = 0 { } {
    \message{^^J!~Error:~number~too~large,
      ~some~digits~will~be~discarded^^J}
  }
}

\cs_new:Npn \amc_invalid_digits:Nn #1 #2 {
  \clist_clear:N #1
  \prg_replicate:nn { #2 } { \clist_put_left:Nx #1 { -1 } }
}

\cs_new:Npn \amc_get_fp_sign:Nn #1 #2 {
  \fp_compare:nNnTF #2 < 0 {
    \int_set:Nn #1 { -1 }
  }{
    \fp_compare:nNnTF #2 > 0 {
      \int_set:Nn #1 { 1 }
    }{
      \int_set:Nn #1 { 0 }
    }
  }
}

\cs_new:Npn \amc_get_int_sign:Nn #1 #2 {
  \int_compare:nNnTF #2 < 0 {
    \int_set:Nn #1 { -1 }
  }{
    \int_compare:nNnTF #2 > 0 {
      \int_set:Nn #1 { 1 }
    }{
      \int_set:Nn #1 { 0 }
    }
  }
}

\ExplSyntaxOff
```

The command `\AMCnumericShow{`⟨*value*⟩`}{`⟨*opts*⟩`}` is called to draw all necessary boxes to code a numerical value ⟨*value*⟩ with options given as a comma separated list ⟨*opts*⟩. `\AMCnumericOpts{`⟨*opts*⟩`}` can be used to set some default values for these options.

Begin with the available options:

```
\def\AMCntextGoto{}
\def\AMCntextVHead#1{\emph{b#1}}
\newdimen\AMCnumeric@Hspace\AMCnumeric@Hspace=.5em
\newdimen\AMCnumeric@Vspace\AMCnumeric@Vspace=1ex
\ExplSyntaxOn

\keys_define:nn { amcnumeric } {
  Tsign .code:n = {\def\AMCntextSign{#1}},
  Tsign .initial:n = {},
  Tpoint .code:n = {\def\AMCdecimalPoint{#1}},
  Tpoint .initial:n = { \raisebox{1ex}{\bf .} },
  Texponent .code:n = {\def\AMCexponent{#1}},
  Texponent .initial:n = { $\times10$\textasciicircum },
  vspace .code:n = {\AMCnumeric@Vspace=#1},
  hspace .code:n = {\AMCnumeric@Hspace=#1},
  bordercol .code:n = {\def\AMCncol@Border{#1}},
  bordercol .initial:n = { lightgray },
  borderwidth .code:n = {\def\AMCncol@BorderWidth{#1}},
  borderwidth .initial:n = { 1mm },
  backgroundcol .code:n = {\def\AMCncol@Background{#1}},
  backgroundcol .initial:n = { white },
  digits .int_set:N = \amc_num_ndigits_int,
  digits .initial:n = { 3 },
  decimals .int_set:N = \amc_num_decd_int,
  decimals .initial:n = { 0 },
  exponent .int_set:N = \amc_num_expo_int,
  exponent .initial:n = { 0 },
  base .int_set:N = \amc_num_base_int,
  base .initial:n = { 10 },
  sign .bool_set:N = \amc_num_sign_bool,
  sign .initial:n = { true },
  sign .default:n = { true },
  exposign .bool_set:N = \amc_num_exposign_bool,
  exposign .initial:n = { true },
  exposign .default:n = { true },
  strict .bool_set:N = \amc_num_strict_bool,
  strict .initial:n = { false },
  strict .default:n = { true },
  scoring .bool_set:N = \amc_num_scoring_bool,
  scoring .initial:n = { true },
  scoring .default:n = { true },
  ignoreblank .bool_set:N = \amc_num_ignoreblank_bool,
  ignoreblank .initial:n = { false },
  ignoreblank .default:n = { true },
  vertical .bool_set:N = \amc_num_vertical_bool,
  vertical .initial:n = { false },
  vertical .default:n = { true },
  expovertical .bool_set:N = \amc_num_expovertical_bool,
  expovertical .initial:n = { false },
  expovertical .default:n = { true },
  reverse .bool_set:N = \amc_num_reverse_bool,
  reverse .initial:n = { false },
```

```
  reverse .default:n = { true },
  vhead .bool_set:N = \amc_num_vhead_bool,
  vhead .initial:n = { false },
  vhead .default:n = { true },
  Tvhead .code:n = {\clist_set:Nx \amc_tvhead_clist {\clist_reverse:n {#1}}},
  Tvhead .initial:n = {}, % \c_empty_clist does not work with \clist_reverse:n
  vheadunitindex .int_set:N = \amc_vheadunitindex_int,
  vheadunitindex .initial:n = 0,
  nozero .bool_set:N = \amc_num_nozero_bool,
  nozero .initial:n = { false },
  nozero .default:n = { true },
  significant .bool_set:N = \amc_num_significant_bool,
  significant .initial:n = { false },
  significant .default:n = { true },
  scoreexact .code:n = {\def\AMC@numeric@scoreexact{#1}},
  scoreexact .initial:n = { 2 },
  scoreapprox .code:n = {\def\AMC@numeric@scoreapprox{#1}},
  scoreapprox .initial:n = { 1 },
  scorewrong .code:n = {\def\AMC@numeric@scorewrong{#1}},
  scorewrong .initial:n = { 0 },
  exact .int_set:N = \amc_num_exact_int,
  exact .initial:n = { 0 },
  approx .int_set:N = \amc_num_approx_int,
  approx .initial:n = { 0 },
  keepas .code:n = {\def\AMC@numeric@keepas{#1}},
  keepas .initial:n = {},
  alsocorrect .code:n = {\def\AMC@numeric@alsocorrect{#1}},
  alsocorrect .initial:n = {}
}

\cs_new:Npn \amc_num_setopts #1 {
  \keys_set:nn { amcnumeric } { #1 }
}

\cs_new_nopar:Nn \amc_num_check_score_opts: {
  \bool_if:NTF \amc_num_ignoreblank_bool {
    \int_compare:nNnTF \amc_num_base_int = { 10 } { } {
      \message{^^J!~Error:~ignoreblank~can~only~be~used~with~number~base~10^^J}
    }
  } {}
}

\cs_new_eq:NN \AMCnumericOpts \amc_num_setopts

```

The command `\amc_num_char:nn{`⟨*inside*⟩`}{`⟨*answer*⟩`}` draw a box with content ⟨*inside*⟩ (only if needed), where ⟨*answer*⟩ is `\AMC@checkedbox` if the corresponding choice is correct and empty if not.

```
\cs_new:Npn \amc_num_char:nn #1 #2 {
  \global\advance\AMCrep@count\@ne\relax
  \AMCmessage{REP= \the\AMCrep@count :
    \ifx#2\AMC@checkedbox B\else M\fi }
  \ifAMC@correc
    \protect\AMC@keyBox@{#1}{#2}{1}{case : \AMCid@name :
```

```
      \the\AMCid@quest , \the\AMCrep@count}
  \else
    \protect\AMC@keyBox@{#1}{}{1}{case : \AMCid@name :
      \the\AMCid@quest , \the\AMCrep@count}
  \fi
}
```

The command `\amc_num_digit_box:nn{`⟨*i*⟩`}{`⟨*j*⟩`}` draws a box for current digit value ⟨*i*⟩, where ⟨*j*⟩ is the correct value for the current digit. If ⟨*i*⟩ is greater than 9, it is converted to a character from the English alphabet (A for 10, B for 11...)

```
\int_new:N \amc_num_digit_value_int
\tl_new:N \amc_num_digit_value_tl
\cs_new:Npn \amc_num_digit_box:nn #1 #2 {
  \int_set:Nn \amc_num_digit_value_int { #1 }
  \tl_set:Nn \amc_num_digit_value_tl {
    \int_compare:nNnTF { \amc_num_digit_value_int } < { 10 }
    { \int_to_arabic:n { \amc_num_digit_value_int } }
    { \int_to_Alph:n { \amc_num_digit_value_int - 9 } }
  }
  \int_compare:nNnTF { #1 } = { #2 } {
    \amc_num_char:nn{ \tl_use:N \amc_num_digit_value_tl }
                    {\AMC@checkedbox}
  } {
    \amc_num_char:nn{ \tl_use:N \amc_num_digit_value_tl }
                    {}
  }
}
```

The command `\amc_num_sign_boxes:Nn{`⟨*sign*⟩`}{`⟨*prefix*⟩`}` draws two boxes for the students to code the sign (with a right value given by the boolean ⟨*negative*⟩).

```
\cs_new:Npn \amc_num_sign_boxes:Nn #1 #2 {
  \int_case:nn { #1 } {
    { -1 } {
      \hbox{\amc_num_char:nn{$+$}{}}
      \vspace{\AMCnumeric@Vspace}
      \AMCmessage{B=set. sign #2 =1}
      \hbox{\amc_num_char:nn{$-$}{\AMC@checkedbox}}
      \AMCmessage{B=set. sign #2 =-1}
    }
    { 1 } {
      \hbox{\amc_num_char:nn{$+$}{\AMC@checkedbox}}
      \vspace{\AMCnumeric@Vspace}
      \AMCmessage{B=set. sign #2 =1}
      \hbox{\amc_num_char:nn{$-$}{}}
      \AMCmessage{B=set. sign #2 =-1}
    }
    { 0 } {
      \hbox{\amc_num_char:nn{$+$}{}}
      \vspace{\AMCnumeric@Vspace}
      \AMCmessage{B=set. sign #2 =1}
      \hbox{\amc_num_char:nn{$-$}{}}
      \AMCmessage{B=set. sign #2 =-1}
    }
  }
```

```
}
```

The command `\amc_num_digit_boxes_h:nnn{`⟨*varname*⟩`}{`⟨*correct*⟩`}{`⟨*maxdigit*⟩`}` draws a serie of boxes for all possible values of a digit (from 0 to ⟨*maxdigit*⟩), where the correct value is ⟨*correct*⟩, transmitting scoring data to AMC so that the vaiable ⟨*varname*⟩ will be set to the value chosen by the student.

```
\cs_new:Npn \amc_num_digit_boxes_h:nnn #1 #2 #3 {
  \int_step_inline:nnnn
  { \bool_if:NTF \amc_num_nozero_bool { 1 } { 0 } }
  { 1 } { #3 - 1 } {
    \amc_num_digit_box:nn { ##1 }{ #2 }
    \AMCmessage{B= set. #1 = ##1}
    \hspace{\AMCnumeric@Hspace}
  }
  \hspace{-\AMCnumeric@Hspace}
}

\cs_new:Npn \amc_num_digit_boxes_v:nnn #1 #2 #3 {
  \int_step_inline:nnnn
  { \bool_if:NTF \amc_num_nozero_bool { 1 } { 0 } }
  { 1 } { #3 - 1 } {
    \vbox{\hbox{
        \amc_num_digit_box:nn { ##1 }{ #2 }
      }}
    \AMCmessage{B= set. #1 = ##1}
    \int_compare:nNnTF { ##1 } < { #3 - 1 } {
      \vspace{\AMCnumeric@Vspace}
    } {}
  }
}

\int_new:N \amc_num_first_digit_int
\cs_new:Npn \amc_num_digit_boxes_vr:nnn #1 #2 #3 {
  \int_set:Nn \amc_num_first_digit_int
  { \bool_if:NTF \amc_num_nozero_bool { 1 } { 0 } }
  \int_step_inline:nnnn { #3 - 1 } { -1 }
  \amc_num_first_digit_int {
    \vbox{\hbox{
        \amc_num_digit_box:nn { ##1 }{ #2 }
      }}
    \AMCmessage{B= set. #1 = ##1}
    \int_compare:nNnTF { ##1 } > \amc_num_first_digit_int {
      \vspace{\AMCnumeric@Vspace}
      } {}
  }
}
```

The command `\amc_num_integer_boxes_v:Nnn{`⟨*correct digits*⟩`}{`⟨*prefix*⟩`}`⟨*decimals*⟩ draws boxes for integer entry, without the sign.

```
\cs_new:Npn \amc_num_integer_boxes_v:Nnn #1 #2 #3 {
```

begin a loop over all digits,

```
  \int_set_eq:NN \amc_num_digit_int { \clist_count:N #1 }
  \clist_map_inline:Nn #1 {
```

place the decimal point if necessary,

```
    \int_compare:nNnTF \amc_num_digit_int = { #3 } {
      \hbox{ \AMCdecimalPoint }\hspace{\AMCnumeric@Hspace}
    } { }
```

draw the box for this digit,

```
    \hbox{\vbox{
        \bool_if:NTF \amc_num_vhead_bool {
          \vbox{\hbox to \AMC@boxedwidth{\hfill\AMCntextVHead{ \int_eval:n
                { \amc_num_digit_int - 1 } }\hfill}}
          \vspace{\AMCnumeric@Vspace}
        } { }
        \bool_if:NTF \amc_num_reverse_bool {
          \amc_num_digit_boxes_vr:nnn { #2
            \int_to_Alph:n \amc_num_digit_int }
          { ##1 } { \amc_num_base_int }
        } {
          \amc_num_digit_boxes_v:nnn { #2
            \int_to_Alph:n \amc_num_digit_int }
          { ##1 } { \amc_num_base_int }
        }
      }}
```

and end the loop over digits, adding space if this is not the last one.

```
    \int_compare:nNnTF \amc_num_digit_int > 1 {
      \hspace{\AMCnumeric@Hspace}
    } { }
    \int_decr:N \amc_num_digit_int
  }
}

```

The command `\amc_num_integer_boxes_h:Nnn{`⟨*correct digits*⟩`}{`⟨*prefix*⟩`}`⟨*decimals*⟩ does the same, in horizontal mode.

```

\cs_new:Npn \amc_num_integer_boxes_h:Nnn #1 #2 #3 {
  \vbox{
    \int_set_eq:NN \amc_num_digit_int { \clist_count:N #1 }
    \clist_map_inline:Nn #1 {
      \int_compare:nNnTF
      \amc_num_digit_int = { #3 } {
        \hbox{ \AMCdecimalPoint }
      } { }
      \hbox{
        \amc_num_digit_boxes_h:nnn { #2
          \int_to_Alph:n \amc_num_digit_int }
        { ##1 } \amc_num_base_int
      }
      \int_compare:nNnTF \amc_num_digit_int > 1 {
        \vspace{\AMCnumeric@Vspace}
      } { }
      \int_decr:N \amc_num_digit_int
    }}
}

```

Finally, `\amc_num_integer_boxes:NnnNN{`⟨*correct digits*⟩`}{`⟨*prefix*⟩`}{`⟨*decimals*⟩`}{`⟨*sign bool*⟩`}{`⟨*sign*⟩`}` draws boxes for integer entry, including the sign if ⟨*sign bool*⟩ is true. When using the `strict` option, check the + box for a null value.

```
\cs_new:Npn \amc_num_integer_boxes:NnnNN #1 #2 #3 #4 #5 {
  \hbox{
    \bool_if:NTF { #4 } {
      \vbox{
        \ifx\AMCntextSign\@empty\@empty\else
        \hbox{\AMCntextSign}\vspace{\AMCnumeric@Vspace}\fi
        \bool_if:NTF \amc_num_strict_bool {
          \int_compare:nNnTF { #5 } = 0 {
            \amc_num_sign_boxes:Nn { 1 } { #2 }
          }{
            \amc_num_sign_boxes:Nn { #5 } { #2 }
          }
        }{
          \amc_num_sign_boxes:Nn { #5 } { #2 }
        }
      }
      \hspace{.5em}
      \vrule
      \hspace{.5em}
    } { }
    \hbox{
      \bool_if:NTF \amc_num_vertical_bool
      \amc_num_integer_boxes_v:Nnn \amc_num_integer_boxes_h:Nnn
      #1 { #2 } { #3 }
    }
  }
}
```

The command `\amc_num_build_integer_scoring:Nnnnn{`⟨*tl var*⟩`}{`⟨*sign bool*⟩`}{`⟨*prefix*⟩`}{`⟨*n*⟩`}{`⟨*decimals*⟩`}` builds a scoring to compute an integer from a serie of ⟨*n*⟩-digits boxes (from which ⟨*decimals*⟩ are for decimals), with name prefix ⟨*prefix*⟩, using a sign variable if ⟨*sign bool*⟩ is true.

```
\cs_new:Npn \amc_num_build_integer_scoring:Nnnnn #1 #2 #3 #4 #5 {
  \amc_num_check_score_opts:
  \tl_clear:N #1
  \int_set_eq:NN \amc_num_digit_int { #4 }
  \int_while_do:nNnn \amc_num_digit_int > 0 {
    \bool_if:NTF \amc_num_strict_bool {
      \AMCmessage{B=requires. #3
        \int_to_Alph:n \amc_num_digit_int = 1}
    } {
      \AMCmessage{B=default. #3
        \int_to_Alph:n \amc_num_digit_int =
        \bool_if:NTF \amc_num_ignoreblank_bool { } { 0 }
        }
    }
    \int_compare:nNnTF \amc_num_digit_int = #4 { } {
      \bool_if:NTF \amc_num_ignoreblank_bool {
```

```
        \tl_put_right:Nx #1 { ~.~ }
        \int_compare:nNnTF \amc_num_digit_int = #5 {
          \tl_put_right:Nx #1 { "." ~.~ }
        } { }
      } {
        \tl_put_left:Nn #1 { ( }
        \tl_put_right:Nx #1 { ) *
          \int_use:N \amc_num_base_int + }
      }
    }
    \tl_put_right:Nx #1
    { #3 \int_to_Alph:n \amc_num_digit_int }
    \int_decr:N \amc_num_digit_int
  }
  \bool_if:NTF \amc_num_ignoreblank_bool {
    \tl_put_left:Nn #1 { ( 0 + ( }
    \tl_put_right:Nn #1 { ) ) }
    \int_compare:nNnTF \amc_num_decd_int > 0 {
      \tl_put_right:Nx #1 { * ( \int_use:N \amc_num_base_int ** \int_eval:n { #5 } ) }
    } { }
  } {
    \tl_put_left:Nn #1 { ( }
    \tl_put_right:Nn #1 { ) }
  }
  \bool_if:NTF { #2 } {
    \bool_if:NTF \amc_num_strict_bool {
      \AMCmessage{B=requires. sign #3 =1}
    } {
      \AMCmessage{B=default. sign #3 =1}
    }
    \tl_put_right:Nx #1 { * ( sign #3 ) }
  } { }
}

```

Then the command `\AMCnumericShow{`⟨*x*⟩`}{`⟨*options*⟩`}` itself:

```

\fp_new:N \amc_num_result_fp
\fp_new:N \amc_num_correct_fp
\clist_new:N \amc_num_digits_clist
\clist_new:N \amc_num_expo_digits_clist
\int_new:N \amc_num_digit_int
\int_new:N \amc_num_sign_int
\int_new:N \amc_num_expo_sign_int
\tl_new:N \amc_num_compute_tl
\tl_new:N \amc_num_expo_tl
\int_new:N \amc_num_correct_expo_int

\cs_new:Npn \amc_numeric_show:nn #1 #2 {
```

We have to tell AMC that the scoring we will give concerns this question:

```
  \ifAMC@ensemble\ifAMCformulaire@dedans
    \AMCmessage{Q=\the\AMCid@quest}
  \fi\fi
```

Then we parse the options from ⟨*opts*⟩:

```
{\keys_set:nn { amcnumeric } { #2 }
```

if Tvhead is set, we adapt the AMCntextVHead macro

```
\int_compare:nTF {\amc_vheadunitindex_int = 0}
 {\int_set:Nn \amc_vheadunitindex_int {\amc_num_decd_int + 1}}
 {}

\clist_if_empty:NTF \amc_tvhead_clist
 {}
 {
  \def\AMCntextVHead##1{
     \int_set:Nn \l_tmpa_int
     {\int_max:nn
   {(##1) - \amc_num_decd_int + \amc_vheadunitindex_int}
   {0}}
  \emph{\clist_item:Nn \amc_tvhead_clist {\l_tmpa_int}}}}

   \bool_if:nTF { \bool_if_p:N\amc_num_significant_bool
     && \int_compare_p:n { \amc_num_base_int != 10 } } {
     \message{^^J!~AMCnumeric~Error:~significant=true~can't~be~used~with~base!=10.^^J}
   } {}
   \bool_if:nTF { \int_compare_p:n { \amc_num_expo_int != 0 }
     && \int_compare_p:n { \amc_num_base_int != 10 } } {
     \message{^^J!~AMCnumeric~Error:~scientific~notation~can't~be~used~with~base!=10.^^J}
   } {}
```

Convert the floating point correct value to integer, taking into account the parameters `significant`, `exponent` and `decimals`:

```
  \ifx\@empty#1\@empty
    \fp_set:Nn \amc_num_correct_fp { 0 }
    \fp_set:Nn \amc_num_mantissa_fp { 0 }
    \int_set:Nn \amc_num_correct_expo_int { 0 }
  \else
  \bool_if:NTF \amc_num_significant_bool {
    \amc_fp_n_significant_digits:Nnn \amc_num_correct_fp { #1 } \amc_num_ndigits_int
  } {
    \int_compare:nNnTF \amc_num_expo_int > 0 {
      \amc_fp_decompose:NNn \amc_num_mantissa_fp \amc_num_correct_expo_int { #1 }
      \int_compare:nNnTF { \amc_num_ndigits_int - \amc_num_decd_int } > 1 {
        \fp_set:Nn \amc_num_mantissa_fp {
          \amc_num_mantissa_fp * \amc_num_base_int^( \amc_num_ndigits_int - \amc_num_decd_int - 1 )
        }
        \int_set:Nn \amc_num_correct_expo_int {
          \amc_num_correct_expo_int - ( \amc_num_ndigits_int - \amc_num_decd_int - 1 )
        }
      } {}
      \amc_fp_n_digits:Nnn \amc_num_correct_fp \amc_num_mantissa_fp \amc_num_decd_int
    } {
      \amc_fp_n_digits:Nnn \amc_num_correct_fp { #1 } \amc_num_decd_int
    }
  }
  \fi
```

Now extracts the required digits:

```
\ifx\@empty#1\@empty
  \amc_invalid_digits:Nn \amc_num_digits_clist \amc_num_ndigits_int
  \amc_invalid_digits:Nn \amc_num_expo_digits_clist \amc_num_expo_int
  \int_set:Nn \amc_num_sign_int { 0 }
  \int_set:Nn \amc_num_expo_sign_int { 0 }
\else
  \amc_fp_to_digits:Nnnn \amc_num_digits_clist \amc_num_correct_fp
    \amc_num_ndigits_int \amc_num_base_int
  \amc_get_fp_sign:Nn \amc_num_sign_int \amc_num_correct_fp
  \int_compare:nNnTF \amc_num_expo_int > 0 {
    \amc_fp_to_digits:Nnnn \amc_num_expo_digits_clist \amc_num_correct_expo_int
      \amc_num_expo_int \amc_num_base_int
    \amc_get_int_sign:Nn \amc_num_expo_sign_int \amc_num_correct_expo_int
  } {}
\fi
```

The question scoring is given to AMC (if requested by the `scoring=true` option). Note that the variable `intV` refers to the correct value, and `intX` to the value entered by the student.

```
\fp_set:Nn \amc_num_result_fp { #1 }
\AMCmessage{B=numval=\fp_to_scientific:N \amc_num_result_fp ,
  numex=\int_use:N \amc_num_exact_int,
  numapp=\int_use:N \amc_num_approx_int,
  numsex=\AMC@numeric@scoreexact,
  numsapp=\AMC@numeric@scoreapprox
}
\bool_if:NTF \amc_num_scoring_bool {
  \AMCmessage{B=haut=,mz=,d=undef,p=undef,
    formula=(Vdifference <= \int_use:N \amc_num_exact_int ?
    \AMC@numeric@scoreexact :
    \int_compare:nNnTF \amc_num_approx_int = 0 {
      \AMC@numeric@scorewrong
    } {
      (Vdifference <= \int_use:N\amc_num_approx_int ?
        \AMC@numeric@scoreapprox : \AMC@numeric@scorewrong)
    }
    )}
} {}
\amc_num_build_integer_scoring:Nnnnn
  \amc_num_compute_tl \amc_num_sign_bool { digit } \amc_num_ndigits_int
  \amc_num_decd_int
\int_compare:nNnTF \amc_num_expo_int > 0 {
  \amc_num_build_integer_scoring:Nnnnn
    \amc_num_expo_tl \amc_num_exposign_bool { expo } \amc_num_expo_int  { 0 }
  \AMCmessage{B= set. intE = \amc_num_expo_tl}
} {}
\AMCmessage{B= set.intV = \fp_to_int:N\amc_num_correct_fp ,
  set.intXX = \amc_num_compute_tl }
\int_compare:nNnTF \amc_num_expo_int > 0 {
  \AMCmessage{B= set.intX = intXX * \int_use:N\amc_num_base_int **( intE - (\int_use:N\amc_num_correc
}{
  \AMCmessage{B= set.intX = intXX}
}
\int_compare:nNnTF \amc_num_expo_int > 0 {
  \AMCmessage{B= set.valueX = intXX * \int_use:N\amc_num_base_int ** (intE - \int_use:N\amc_num_decd_
```

```
}{
  \AMCmessage{B= set.valueX = intXX * \int_use:N\amc_num_base_int ** (- \int_use:N\amc_num_decd_int)}
}
\ifx\@empty\AMC@numeric@keepas\@empty\else
\AMCmessage{B= setglobal.\AMC@numeric@keepas = valueX}
\fi
\ifx\@empty#1\@empty
\bool_if:NTF \amc_num_significant_bool {
  \AMCmessage{B=set.Vdifference=0}
}{
  \ifx\@empty\AMC@numeric@alsocorrect\@empty
    \AMCmessage{B=set.Vdifference=0}
  \else
    \AMCmessage{B="set.Vdifference =
      amcvdifference( \AMC@numeric@alsocorrect, valueX, \int_use:N\amc_num_decd_int, \int_use:N\amc_n
      "}
  \fi
}
\else
\bool_if:NTF \amc_num_significant_bool {
  \AMCmessage{B=set.Vdifference="min( abs((intV)-(intX)) ,
    abs(\int_use:N\amc_num_base_int * (intV) - (intX)) ,
    abs((intV) - \int_use:N\amc_num_base_int * (intX)) )"}
} {
  \ifx\@empty\AMC@numeric@alsocorrect\@empty
  \AMCmessage{B=set.Vdifference=abs((intV)-(intX))}
  \else
  \AMCmessage{B="set.Vdifference =
    min( amcvdifference( \AMC@numeric@alsocorrect, valueX, \int_use:N\amc_num_decd_int, \int_use:N\am
    abs((intV)-(intX)) )"}
  \fi
}
\fi
```

Begin now with the frame around all the boxes:

```
\ifAMC@extractOnly\else
\vspace{1.5ex}\par{
  \fboxrule=\AMCncol@BorderWidth
  \fcolorbox{\AMCncol@Border}{\AMCncol@Background}{
    \bool_if:NTF \amc_num_expovertical_bool {
      \hbox{\vbox{
        \vbox{\amc_num_integer_boxes:NnnNN
        \amc_num_digits_clist { digit } \amc_num_decd_int \amc_num_sign_bool
        \amc_num_sign_int}
        \int_compare:nNnTF \amc_num_expo_int > 0 {
          \vspace{\AMCnumeric@Vspace}
          \vbox{\hbox{\AMCexponent}}
          \vspace{\AMCnumeric@Vspace}
          \vbox{\amc_num_integer_boxes:NnnNN
            \amc_num_expo_digits_clist { expo } { 0 } \amc_num_exposign_bool
            \amc_num_expo_sign_int}
        } {}
      }}
    } {
```

```
        \amc_num_integer_boxes:NnnNN
        \amc_num_digits_clist { digit } \amc_num_decd_int \amc_num_sign_bool
        \amc_num_sign_int
        \int_compare:nNnTF \amc_num_expo_int > 0 {
          \hspace{\AMCnumeric@Hspace}\AMCexponent\hspace{\AMCnumeric@Hspace}
          \amc_num_integer_boxes:NnnNN
          \amc_num_expo_digits_clist { expo } { 0 } \amc_num_exposign_bool
          \amc_num_expo_sign_int
        } {}
      }
    }
  }
  \fi
```

And tell AMC that we finished with this question:

```
  \ifAMC@ensemble\else\vspace{\AMCpostNquest}\par\fi
  \ifAMC@ensemble\ifAMCformulaire@dedans
    \AMCmessage{FQ}
  \fi\fi
  }
}

\cs_new_eq:NN \AMCnumericShow \amc_numeric_show:nn

```

`\AMCnumericHide` is called when the boxes are not to be drawn (in the question sheets for separate answer sheet layout), and `\AMCnumericChoices{`⟨*value*⟩`}{`⟨*options*⟩`}` is the function to be used in the LaTeX source code of the exam.

```
\cs_new:Npn \amc_numeric_hide:nn #1 #2 {
  \keys_set:nn { amcnumeric } { #2 }
  \AMCntextGoto
  \ifAMC@qbloc\else\vspace{1.5ex}\par\fi
}

\cs_new_eq:NN \AMCnumericHide \amc_numeric_hide:nn

\ExplSyntaxOff
\def\AMCnumericChoicesPlain{%
  \AMC@if@separate@question{\AMC@mem@category{numeric}}%
  \AMCformatChoices{\AMCnumericShow}{\AMCnumericHide}%
}
```

The `{`⟨*value*⟩`}` argument is often given as a macro, that is to be expanded before calling `\AMCnumericChoicesPlain`, so that its value will be the same in the separate answer sheet...

```
\ExplSyntaxOn

\cs_new:Npn \amc_numeric_choices:nn #1#2 {
  \AMCnumericChoicesPlain{#1}{#2}
}
\cs_generate_variant:Nn \amc_numeric_choices:nn { xn }
\cs_new_eq:NN \AMCnumericChoices \amc_numeric_choices:xn

\ExplSyntaxOff
```

4.13.3 Intervals

\AMCIntervals The command \AMCIntervals{⟨*x*⟩}{⟨*x0*⟩}{⟨*x1*⟩}{⟨*delta*⟩} can be used to present answers as intervals $[x_i, x_i + \delta[$ covering $[\langle x0\rangle, \langle x1\rangle[$, such that the only interval containing $\langle x\rangle$ is declared as \correctchoice, and the other as \wrongchoice.

For this command to work, one has to load package fp.

As an example,

```
\begin{question}{quarter}
  In which interval falls $1/4$?
  \begin{multicols}{5}
    \begin{choices}[o]
      \AMCIntervals{0.25}{0}{1}{0.1}
    \end{choices}
  \end{multicols}
\end{question}
```

produces (in correction mode):

Question 12 In which interval falls 1/4?

☐ $[0, 0.1[$	■ $[0.2, 0.3[$	☐ $[0.4, 0.5[$	☐ $[0.6, 0.7[$	☐ $[0.8, 0.9[$
☐ $[0.1, 0.2[$	☐ $[0.3, 0.4[$	☐ $[0.5, 0.6[$	☐ $[0.7, 0.8[$	☐ $[0.9, 1[$

Note that the interval formatting can be changed redefining the \AMCintervalFormat command, which is originally defined as

```
\def\AMCIntervalFormat#1#2{[#1,\,#2[}
```

to follow local conventions (writting $[a, b)$ instead of $[a, b[$ is for example a common usage).

```
\ExplSyntaxOn

\fp_new:N \amc_interv_a
\fp_new:N \amc_interv_b
\cs_new:Npn \amc_intervals:nnnn #1 #2 #3 #4 {
  \fp_set:Nn \amc_interv_a { #2 }
  \fp_do_while:nn { \amc_interv_a < #3 } {
    \fp_set:Nn \amc_interv_b { \amc_interv_a + #4 }
    \fp_compare:nTF { \amc_interv_a <= #1 < \amc_interv_b }
      \correctchoice \wrongchoice
    {\AMCIntervalFormat{\fp_use:N \amc_interv_a}{\fp_use:N \amc_interv_b}}
    \fp_set:Nn \amc_interv_a \amc_interv_b
  }
}
\cs_new_eq:NN \AMCIntervals \amc_intervals:nnnn

\ExplSyntaxOff
```

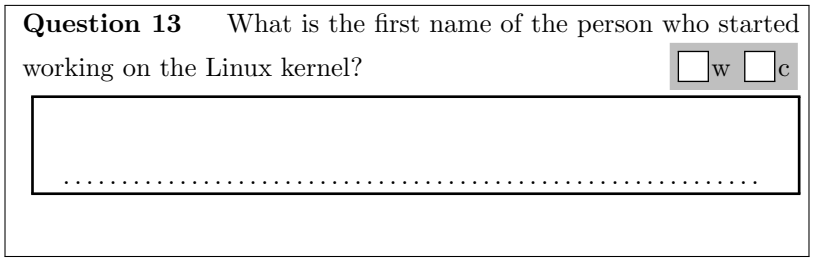
4.14 Open questions

\AMCOpen The command \AMCOpen{⟨*options*⟩}{⟨*choices*⟩} can be used as a replacement for the choices environment when asking the student to write some answer by hand. The teacher will correct and mark this answer either on the paper before scanning, or with manual data capture, thanks to the scoring boxes.

As an example,

```
\begin{question}{Linux}
  What is the first name of the person who started working on the Linux kernel?
  \AMCOpen{}{\wrongchoice[w]{w}\scoring{0}\correctchoice[c]{c}\scoring{2}}
\end{question}
```

shows:



The teacher will have to tick the 'w' box for wrong answers, and the 'c' box for correct answers.

Begin with the options definitions:

```
\def\AMCotextGoto{}
\def\AMCotextReserved{}
\def\AMCocol@Background{lightgray}
\def\AMCocol@BoxFrameRule{white}
\def\AMCocol@FrameRule{black}
\def\AMCocol@Foreground{}
\def\AMCopen@answer{}
\def\AMCopen@question{}
\def\AMCopen@lineuptext{}
\define@key{AMCOpen}{backgroundcol}{\def\AMCocol@Background{#1}}
\define@key{AMCOpen}{foregroundcol}{\def\AMCocol@Foreground{#1}}
\define@key{AMCOpen}{Treserved}{\def\AMCotextReserved{#1}}
\define@key{AMCOpen}{question}[\AMCid@name]{\def\AMCopen@question{#1}}
\define@key{AMCOpen}{answer}{\def\AMCopen@answer{#1}}
\define@key{AMCOpen}{contentcommand}[AMCopen@lines]{\def\AMCopen@contentcommand{#1}}
\newdimen\AMCopen@Hspace\AMCopen@Hspace=.5em
\define@key{AMCOpen}{hspace}{\AMCopen@Hspace=#1}
\def\AMCopen@Width{.95\linewidth}
\define@key{AMCOpen}{width}{\def\AMCopen@Width{#1}}
\newdimen\AMCopen@LineHeight\AMCopen@LineHeight=1cm
\define@key{AMCOpen}{lineheight}{\AMCopen@LineHeight=#1}
\newcount\AMCopen@Lines\AMCopen@Lines=1
\define@key{AMCOpen}{lines}{\AMCopen@Lines=#1}
\newdimen\AMCopen@boxmargin\AMCopen@boxmargin=3pt
\define@key{AMCOpen}{boxmargin}{\AMCopen@boxmargin=#1}
\newdimen\AMCopen@boxframerule\AMCopen@boxframerule=1pt
\define@key{AMCOpen}{boxframerule}{\AMCopen@boxframerule=#1}
\define@key{AMCOpen}{boxframerulecol}{\def\AMCocol@BoxFrameRule{#1}}
\define@key{AMCOpen}{framerulecol}{\def\AMCocol@FrameRule{#1}}
\newdimen\AMCopen@framerule\AMCopen@framerule=1pt
\define@key{AMCOpen}{framerule}{\AMCopen@framerule=#1}
\define@key{AMCOpen}{lineuptext}{\def\AMCopen@lineuptext{#1}}
\define@boolkey{AMCOpen}{dots}[true]{}
\define@boolkey{AMCOpen}{scan}[true]{}
\define@boolkey{AMCOpen}{retick}[true]{}
```

```
\define@boolkey{AMCOpen}{annotate}[false]{}
\define@boolkey{AMCOpen}{lineup}[false]{}
\setkeys{AMCOpen}{dots,scan,retick,annotate,lineup,contentcommand}
\newcommand\AMCopenOpts[1]{\setkeys{AMCOpen}{#1}}
```

The command \AMCOpen is similar to \AMCnumericChoices, calling either \AMCopenShow or \AMCopenHide.

```
\newcommand\AMCopen@lines{%
  \begin{minipage}{\AMCopen@Width}%
    \loop\vspace{\AMCopen@LineHeight}
    \hspace*{.5em}\ifAMC@correc\smash{\AMCopen@answer}\def\AMCopen@answer{}\fi%
    \ifKV@AMCOpen@dots%
    \dotfill\hspace*{.5em}
    \fi
    \ifnum\AMCopen@Lines>\@ne\par\advance\AMCopen@Lines\m@ne\repeat%
  \end{minipage}
}
\newcommand\AMCopenShow[2]{
  \ifAMC@ensemble\ifAMCformulaire@dedans%
    \AMCmessage{Q=\the\AMCid@quest}%
  \fi\fi%
  {\setkeys{AMCOpen}{#1}%
    \ifKV@AMCOpen@lineup%
      \ifAMC@ensemble\else%
        \ifx\@empty\AMCopen@lineuptext\@empty\fi%
      \fi%
      \ifAMC@correc\smash{\AMCopen@answer}\fi%
      \ifx\@empty\AMCopen@lineuptext\@empty%
        \dotfill%
      \else%
        \AMCopen@lineuptext\hfill%
      \fi%
    \else%
      \hspace*{.5em}\linebreak[1]\hspace*{\fill}%
    \fi%
    {\AMCnoCompleteMulti%
      \def\AMCbeginAnswer{}\def\AMCendAnswer{}%
      \def\AMCanswer##1##2{\ifAMC@ensemble ##1\else%
          \ifAMC@inside@box ##1\else{\AMCboxOutsideLetter{##1}{##2}}\fi\fi%
        \hspace{\AMCopen@Hspace}}%
      \fboxsep=\AMCopen@boxmargin%
      \fboxrule=\AMCopen@boxframerule%
      \fcolorbox{\AMCocol@BoxFrameRule}{\AMCocol@Background}{%
        \ifAMC@ensemble\AMCopen@question%
          \ifx\@empty\AMCopen@question\@empty\else\hspace{\AMCopen@Hspace}\fi%
        \fi%
        \begin{choicescustom}[o]%
          \ifx\AMCocol@Foreground\@empty\@empty\else%
            \def\AMC@boxcolor{\AMCocol@Foreground}%
          \fi%
          #2%
          \ifKV@AMCOpen@scan\else\AMCdontScan\fi%
          \ifKV@AMCOpen@retick\AMCreTick\fi%
          \ifKV@AMCOpen@annotate\else\AMCdontAnnotate\fi%
```

```
        \end{choicescustom}%
        \ifx\@empty\AMCotextReserved\@empty%
          \hspace{-\AMCopen@Hspace}%
        \else%
          \ifx\AMCocol@Foreground\@empty\@empty%
            \AMCotextReserved%
          \else%
            \textcolor{\AMCocol@Foreground}{\AMCotextReserved}%
          \fi%
        \fi%
      }}%
    \ifKV@AMCOpen@lineup\else%
      \par\nobreak\noindent%
      \hspace*{\fill}{%
        \fboxrule=\AMCopen@framerule%
        \fcolorbox{\AMCocol@FrameRule}{white}{%
          \csname\AMCopen@contentcommand\endcsname
        }}%
      \vspace{\AMCpostOquest}\par%
    \fi%
  }%
  \ifAMC@ensemble\ifAMCformulaire@dedans%
  \AMCmessage{FQ}%
  \fi\fi%
}
\newcommand\AMCopenHide[2]{%
  \AMCotextGoto%
  \ifAMC@qbloc\else\vspace{1.5ex}\par\fi%
}
\def\AMCOpen{%
  \AMC@if@separate@question{\AMC@mem@category{open}}%
  \AMCformatChoices{\AMCopenShow}{\AMCopenHide}%
}
```

4.15 Boxes with letters only

`\AMCBoxOnly` Sometimes the letters printed in the boxes (or just after them) are enough to describe the answers. In such cases, printing the boxes both on the question and on the answer sheet is not necessary. The `\AMCBoxOnly{`⟨*options*⟩`}{`⟨*choices*⟩`}` can be used as a replacement for the `choices` environment:

```
\begin{question}{arm}
  Which letter shows the \textit{arm} on the diagram?
  \AMCBoxOnly{ordered=true}{\wrongchoice[A]{}\correctchoice[B]{}%
    \wrongchoice[C]{}\wrongchoice[D]{}}
\end{question}
```

```
\def\AMCbotextGoto{}
\def\AMCbo@help{}
\define@key{AMCBoxOnly}{help}{\def\AMCbo@help{#1}}
\define@boolkey{AMCBoxOnly}{ordered}[false]{}
\setkeys{AMCBoxOnly}{ordered}
\newcommand\AMCboOpts[1]{\setkeys{AMCBoxOnly}{#1}}
```

```
\newcommand\AMCboShow[2]{%
  \ifAMC@ensemble\ifAMCformulaire@dedans%
    \AMCmessage{Q=\the\AMCid@quest}%
  \fi\fi%
  {\setkeys{AMCBoxOnly}{#1}%
    \def\AMCbeginAnswer{}\def\AMCendAnswer{}%
    \def\AMCanswer##1##2{\hspace{\AMCformHSpace} \ifAMC@ensemble ##1\else%
      \ifAMC@inside@box ##1\else{\AMCboxOutsideLetter{##1}{##2}}\fi\fi%
      }%
    \ifAMC@ensemble\AMCbo@help\fi%
    \ifKV@AMCBoxOnly@ordered%
      \begin{choicescustom}[o]%
    \else%
      \begin{choicescustom}%
    \fi%
      #2
    \end{choicescustom}%
  }%
  \ifAMC@ensemble\ifAMCformulaire@dedans%
  \AMCmessage{FQ}%
  \fi\fi%
}
\newcommand\AMCboHide[2]{
  \AMCbotextGoto%
  \ifAMC@qbloc\else\vspace{1.5ex}\par\fi%
}
\def\AMCBoxOnly{%
  \AMC@if@separate@question{\AMC@mem@category{box}}%
  \AMCformatChoices{\AMCboShow}{\AMCboHide}%
}
```

4.16 Page formatting

4.16.1 Watermark

`\AMCw@termark` `\AMCw@terprint` These commands are used to print a grey "DRAFT" under each page, so as to prevent from printing old versions of the subject.

```
\DeclareFontShape{OT1}{cmr}{b}{n}{<35->cmr17}{}
\def\AMC@watertext{\AMC@loc@draft}
\newcommand\AMCw@termark{%
  \setlength{\@tempdimb}{.5\paperwidth}%
  \setlength{\@tempdimc}{-.5\paperheight}%
  \put(\strip@pt\@tempdimb,\strip@pt\@tempdimc){%
    \makebox(0,0){\rotatebox{45}{\AMC@LR{%
        \textcolor[gray]{0.8}{
          \fontencoding{OT1}\fontfamily{cmr}
          \fontseries{b}\fontshape{n}
          \fontsize{90pt}{120pt}
          \selectfont
          \AMC@watertext}}}}}
\newcommand\AMCw@terprint[1]{%
  \setbox\@tempboxa\vbox to \z@{%
    \vbox{%
```

```
      \hbox to \z@{%
        #1\hss}}\vss}
  \dp\@tempboxa\z@
  \box\@tempboxa}
```

4.16.2 Signs for scan analysis

The following code sets up all the signs to be printed on the pages so as to be able to recognize the position of the boxes on the scans. Four circles ● are printed on the corners (see `\m@rqueCalage`), and binary boxes show the student sheet number (see `\AMCIDBoxesA`), the page (see `\AMCIDBoxesB`) and a checking number (see `\AMCIDBoxesC`).

`\AMC@intituleHead` is the title to be printed at the beginning (used for corrected sheet, and empty on subject). `\AMC@note` is printed at the bottom of each page. You can change its value using `\AMCsetFoot{`⟨*foot*⟩`}`.

```
\def\AMCcercle#1#2{%
  {\setlength{\unitlength}{1mm}%
    \begin{picture}(#1,#1)(-#2,-#2)\thinlines\circle*{#1}\end{picture}}}
\def\m@rqueCalage{\AMCcercle{3.6}{1.8}}
\def\m@rque#1{\AMC@tracebox{1}{#1}{\m@rqueCalage}}
\def\he@dtaille#1{% \par cancels the \leavevmode
  % introduced by https://github.com/pietvo/fancyhdr/commit/6b1ad10eeb5bc3d804f3cd2cf193e6440d0229e6
  \par\vbox to 1cm{#1}}
\def\he@dbas#1{\he@dtaille{\vspace*{\fill}#1}}
\def\he@dhaut#1{\he@dtaille{#1\vspace*{\fill}}}
\def\AMC@intituleHead{\AMC@loc@corrected}
\def\AMC@note{}
\def\AMCsetFoot#1{\def\AMC@note{#1}}
\newcommand\AMCStudentNumber{\the\AMCid@etud}
\newcommand\AMCIDBoxesA{\AMC@binaryCode{id=1,ndigits=\AMC@NCBetud}{\the\AMCid@etud}}
\newcommand\AMCIDBoxesB{\AMC@binaryCode{id=2,ndigits=\AMC@NCBpage}{\thepage}}
\newcommand\AMCIDBoxesC{\AMC@binaryCode{id=3,ndigits=\AMC@NCBcheck}{\the\AMCid@check}}
\newcommand\AMCIDBoxesABC{%
  \hbox{\vbox{\noindent\AMCIDBoxesA\\
    \noindent\AMCIDBoxesB\AMCIDBoxesC}}%
}
\def\AMC@pageHook{%
  \AMC@pagepos%
  \ifAMC@pagelayout\global\advance\AMCid@check\m@ne%
  \ifnum\AMCid@check<1\global\AMCid@check=\AMCid@checkmax\fi%
  \ifAMC@watermark\ifAMC@correchead\else\AMCw@terprint{\AMCw@termark}%
  \fi\fi\fi}
\@ifl@t@r\fmtversion{2020/10/01}
    {\AddToHook{shipout/background}{\put(0in,0in){\AMC@pageHook}}}
    {\AtBeginShipout{\AtBeginShipoutUpperLeft{\AMC@pageHook}}}
\fancypagestyle{AMCpageHeadOnly}{%
  \fancyhf{}\fancyhead[C]{\textsc{\AMC@intituleHead}}%
  \renewcommand{\headrulewidth}{0pt}%
  \renewcommand{\footrulewidth}{0pt}%
}
\fancypagestyle{AMCpageFull}{%
  \fancyhf{}%
  \fancyhead[L]{\AMC@LR{\he@dbas{\leavevmode\m@rque{positionHG}}}}%
```

```
  \fancyhead[R]{\AMC@LR{\he@dbas{\leavevmode\m@rque{positionHD}}}}%
  \fancyfoot[L]{\AMC@LR{\leavevmode\m@rque{positionBG}}}%
  \fancyfoot[R]{\AMC@LR{\leavevmode\m@rque{positionBD}}}%
  \fancyhead[C]{\AMC@LR{\he@dhaut{%
        \begin{minipage}[b]{\AMC@CBtaille}\AMCboxColor{black}%
          \ifAMCids@top\vbox to \AMCids@height{\texttt{+\the\AMCid@etud/\thepage/\the\AMCid@check+}}\fi%
          \AMCIDBoxesABC
        \end{minipage}%
        \ifAMCids@side\hbox to \AMCids@width{\hspace*{\fill}%
          \texttt{+\the\AMCid@etud/\thepage/\the\AMCid@check+}}\fi%
      }}}%
  \fancyhfoffset[EOLR]{5mm}%
  \fancyfoot[C]{\AMC@note}%
  \renewcommand{\headrulewidth}{0pt}%
  \renewcommand{\footrulewidth}{0pt}%
}
\newcommand\AMCsubjectPageTag{%
  \fbox{\texttt{\the\AMCid@etud:\thepage}}%
}
\fancypagestyle{AMCpageNoMarks}{%
  \fancyhf{}%
  \fancyhead[R]{\AMCsubjectPageTag}%
  \fancyfoot[C]{\AMC@note}%
  \renewcommand{\headrulewidth}{0pt}%
  \renewcommand{\footrulewidth}{0pt}%
}
\fancypagestyle{AMCpageEmpty}{%
  \fancyhf{}%
  \renewcommand{\headrulewidth}{0pt}%
  \renewcommand{\footrulewidth}{0pt}%
}
\AtBeginDocument{%
  \ifAMC@pagelayout%
    \ifAMC@correchead
      \pagestyle{AMCpageHeadOnly}
    \else
      \pagestyle{AMCpageFull}
    \fi
  \fi
}
```

4.17 Defining a single exam copy content

\onecopy The command `\onecopy[`⟨*n*⟩`]{`⟨*code*⟩`}` generates ⟨*n*⟩ copies of the subject that is described in ⟨*code*⟩. The LaTeXcode ⟨*code*⟩ that generates a single copy can be a little long, so that the environment `examcopy` is often prefered.

```
\newcommand{\onecopy}[2]{%
  \ifx\AMCNombreCopies\undefined\AMCnum@copies=#1%
  \else\AMCnum@copies=\AMCNombreCopies\fi%
  \AMCmessage{TOTAL=\the\AMCnum@copies}%
  \message{^^JAMC:copies:total=\the\AMCnum@copies^^J}%
  \AMCid@etud=\AMCid@etudstart%
  \ifnum\AMCid@etud=0\AMCid@etud=\AMC@premierecopie\fi%
```

```
  \AMCid@etudfin=\AMCnum@copies%
  \advance\AMCid@etudfin\AMCid@etud\relax%
  \ifAMC@correchead\AMCid@etudfin=\AMC@premierecopie%
    \message{^^JAMC:copies:total=1^^J}%
  \fi
  \ifAMC@pdfform\begin{Form}\fi%
  \loop{%
    \global\AMCrep@nnmax=0%
    \ifAMC@calibration\protected@write\AMC@XYFILE{}{%
      \string\rngstate{\the\AMCid@etud}{\the\AMC@SR}%
    }\fi%
    \AMC@zoneformulairefalse\setcounter{page}{1}\setcounter{section}{0}%
    \ifAMC@ensemble\ifAMC@automarks\pagestyle{AMCpageNoMarks}\fi\fi%
    \AMCnumero{1}%
    \ifAMC@calibration\AMCmessage{ETU=\the\AMCid@etud}\fi%
    \AMC@multiclear%
    \global\AMC@keepmemoryfalse%
      #2%
    \ifAMC@keepmemory\else\AMC@mem@clear\fi%
    \clearpage}%
    \message{^^JAMC:copies:add=1^^J}%
  \advance\AMCid@etud\@ne\ifnum\AMCid@etud<\AMCid@etudfin\repeat%
  \global\AMCid@etudstart=\AMCid@etud%
  \ifAMC@pdfform\end{Form}\fi%
  \AMC@multi@report%
}
```

`\AMCaddpagesto` In some situations, one needs all question sheets to have the same number of pages. The command `\AMCaddpagesto{`$\langle n\rangle$`}` adds enough (white) pages to get at least $\langle n\rangle$ pages in the current question sheet.

```
\newcount\AMC@addpages
\newcommand{\AMCaddpagesto}[1]{%
  \AMC@addpages=#1\advance\AMC@addpages\@ne%
  \clearpage%
  \@whilenum\thepage<\AMC@addpages\do{%
    \ifAMC@automarks\pagestyle{AMCpageEmpty}\fi%
    \hbox{}\clearpage%
  }%
}
```

`AMCcleardoublepage` If you want to print the subject all at one time in duplex mode, it is necessary to end each subject with an even number of pages. This can be achieved using `\AMCcleardoublepage` at the end of the copy definition. This command is also useful inserted before the separate answer sheet (if any).

```
\def\AMCcleardoublepage{%
  \clearpage%
  \ifodd\thepage\else%
    \ifAMC@automarks\pagestyle{AMCpageEmpty}\fi%
    \hbox{}\clearpage%
  \fi%
}
```

`\exemplairepair` To make some differences in the copies, checking if the student sheet number is odd, with

\exemplairepair construct, can be useful.

```
\def\exemplairepair{\ifodd\AMCid@etud}
```

\AMClabel
\AMCref
\AMCref

Commands \AMClabel, \AMCref and \AMCpageref replaces LaTeX's \label, \ref and \pageref to be able to use different labels for different sheets.

```
\newcommand\AMCstudentlabel[1]{\the\AMCid@etud-#1}
\def\AMClabel#1{\expandafter\label{\AMCstudentlabel{#1}}}
\def\AMCref#1{\expandafter\ref{\AMCstudentlabel{#1}}}
\def\AMCpageref#1{\expandafter\pageref{\AMCstudentlabel{#1}}}
```

\AMCqlabel

A label can be created for current question with \AMCqlabel{⟨*label*⟩}. This label can be used with \AMCref and \AMCpageref. This command is defined for backward compatibility only, since \AMClabel can also be used.

```
\newcommand{\AMCqlabel}[1]{%
  \AMClabel{#1}%
}
```

4.18 Pre-association

\AMCassociation

Association between sheets and students can be made before the exam with the \AMCassociation[⟨*filename*⟩]{⟨*id*⟩} command. The optional argument ⟨*filename*⟩ will be used when printing student sheets to files.

```
\newcommand{\AMCassociation}[2][]{%
  \ifAMC@calibration%
  \immediate\write\AMC@XYFILE{\string\association{\the\AMCid@etud}{#2}{#1}}%
  \fi%
}
```

MCstudentslistfile

You can also pass AMC the path to the CSV file with students, and the unique key that can be used, with \AMCstudentslistfile{⟨*path*⟩}{⟨*key*⟩}.

```
\newcommand{\AMCstudentslistfile}[2]{%
  \ifAMC@calibration%
  \immediate\write\AMC@XYFILE{\string\with{studentslistfile=#1}}%
  \immediate\write\AMC@XYFILE{\string\with{studentslistkey=#2}}%
  \fi%
}
```

4.19 Package options

See section 3.1 for the options descriptions.

```
\def\AMC@lang@code{}
\DeclareOptionX{noshuffle}{\AMC@ordretrue}
\DeclareOptionX{noshufflegroups}{\AMC@shuffleGfalse}
\DeclareOptionX{fullgroups}{\AMC@fullGroupstrue}
\DeclareOptionX{answers}{\AMC@correcheadtrue\AMC@correctrue}
\DeclareOptionX{indivanswers}{\AMC@correctrue}
\DeclareOptionX{textpos}{\AMC@textPostrue}
\DeclareOptionX{extractonly}{\AMC@extractOnlytrue\AMC@textPostrue\AMCboxStyle{shape=none}\AMCBoxedAnswer
\DeclareOptionX{box}{\AMC@qbloctrue}
\DeclareOptionX{asbox}{\AMC@asqbloctrue}
\DeclareOptionX{separateanswersheet}{\AMC@ensembletrue}
\DeclareOptionX{digits}{\AMC@inside@digittrue}
```

```
\DeclareOptionX{ordre}{\AMC@ordretrue}
\DeclareOptionX{correc}{\AMC@correcheadtrue\AMC@correctrue}
\DeclareOptionX{modele}{\AMC@correcheadtrue\AMC@correcfalse\AMC@ordretrue}
\DeclareOptionX{correcindiv}{\AMC@correctrue}
\DeclareOptionX{init}{\AMC@SR@time}
\DeclareOptionX{bloc}{\AMC@qbloctrue}
\DeclareOptionX{completemulti}{\AMCcomplete@multitrue}
\DeclareOptionX{insidebox}{\AMC@inside@boxtrue}
\DeclareOptionX{ensemble}{\AMC@ensembletrue}
\DeclareOptionX{chiffres}{\AMC@inside@digittrue}
\DeclareOptionX{outsidebox}{\AMC@outside@boxtrue}
\DeclareOptionX{calibration}{\AMC@calibrationtrue}
\DeclareOptionX{nowatermark}{\AMC@watermarkfalse}
\newcommand\AMC@catalogMode{%
  \AMC@catalogtrue%
  \AMC@watermarkfalse\AMC@correcheadtrue%
  \AMC@correctrue\AMC@ordretrue\AMC@shuffleGfalse%
  \AMC@fullGroupstrue%
  \def\AMC@intituleHead{\AMC@loc@catalog}\AMC@affichekeystrue}
\newcommand\AMC@keys@next{\AMC@keyslinefalse}
\newcommand\AMC@keys@line{\AMC@keyslinetrue}
\DeclareOptionX{catalog}{\AMC@catalogMode}
\DeclareOptionX{keys}[next]{\csname AMC@keys@#1\endcsname{}}
\DeclareOptionX{francais}{\def\AMC@lang@code{FR}\AMC@loc@FR}
\DeclareOptionX{lang}{\def\AMC@lang@code{#1}\csname AMC@loc@#1\endcsname}
\DeclareOptionX{versionA}{%
  \def\AMCid@checkmax{31}\def\AMC@NCBetud{9}\def\AMC@NCBpage{4}%
  \def\AMC@NCBcheck{5}\setlength{\AMC@CBtaille}{4cm}%
  \def\AMC@premierecopie{100}}
\DeclareOptionX{plain}{\AMC@plaintrue}
\DeclareOptionX{nopage}{\AMC@pagelayoutfalse}
\DeclareOptionX{postcorrect}{\AMC@postcorrecttrue}
\DeclareOptionX{automarks}{\AMC@automarkstrue}
\newif\ifAMCneeds@storebox\AMCneeds@storeboxfalse
\DeclareOptionX{storebox}{\AMCneeds@storeboxtrue}
\DeclareOptionX{pdfform}{\AMC@pdfformtrue}
\DeclareOptionX{codedigit}{\AMCcodeID@@{#1}}
\newif\ifAMC@survey\AMC@surveyfalse
\DeclareOptionX{survey}{\AMC@surveytrue}
\ProcessOptionsX
\ifAMCneeds@storebox
  \RequirePackage{storebox}\AtBeginDocument{{}}%
\fi
\ifAMC@pdfform
  \AMCmessage{VAR:project:pdfform=1}%
  \AMCboxStyle{shape=form}%
  \RequirePackage[pageanchor=false]{hyperref}%
\else%
   \AMCmessage{VAR:project:pdfform=0}%
\fi
\AtBeginDocument{%
  \ifAMCneeds@storebox%
    \let\AMC@new@savebox=\newstorebox%
```

```
    \let\AMC@save@box=\storebox%
    \let\AMC@use@box=\usestorebox%
  \fi%
  \AMC@new@savebox{\AMC@ovalbox@R}%
  \AMC@new@savebox{\AMC@ovalbox@RF}%
  \AMC@new@savebox{\AMC@ovalbox@}%
  \AMC@new@savebox{\AMC@ovalbox@F}%
  \AMC@shapeprepare%
}
```

4.20 Survey add-on

Some code and *tikz* settings to help handling surveys, see https://survey.codes/pdf/surveyamc_manual.pdf for more details. This survey add-on is originally written by Claudia Saalbach.

estionnaires (*env.*)
auto (*env.*)
question-auto (*env.*)
values (*env.*)
values-auto (*env.*)
variable-auto (*env.*)
\answer

```
\ifAMC@survey
\NewEnviron{Questionnaires}[1]{
\onecopy{#1}{
\BODY
}
}
\NewEnviron{auto}[1]{
\csvreader[head to column names, separator=tab]{#1}{}{
\BODY
}
}
\NewEnviron{question-auto}[3]{
\csvreader[head to column names, separator=tab]{#1}{}{
\ifcsvstrcmp{#2}{#3}{\BODY \\}{}
}
}
\newenvironment{values}{}{}
\NewEnviron{values-auto}[5]{
\csvreader[head to column names, separator=tab]{#1}{}{
\ifcsvstrcmp{#2}{#3}{
\ifcsvstrcmp{#4}{#5}{\BODY \\
}{}
}{}
}
}
\NewEnviron{variable-auto}[3]{
\foreach \x in {#3}{
\csvreader[head to column names, separator=tab]{#1}{}{
\ifcsvstrcmp{#2}{\x}{\BODY}{}
}
}
}
\newcommand{\answer}[5][]{\global\advance\AMCrep@count\@ne\relax%
  \ifAMC@calibration\AMCmessage{REP=\the\AMCrep@count:B}\fi%
  \global\AMCune@bonnetrue%
  \AMCload@@reponse{\une@rep{\ifAMC@correc\AMC@box{#1}{\AMC@checkedbox}%
```

```
  \else\AMC@box{#1}{}\fi}{#2}{#3}{#4}{#5}}{\the\AMCrep@count}\ignorespaces}
\RequirePackage{tikz}
\usetikzlibrary{positioning, shapes, arrows, tikzmark, decorations.pathreplacing}
\tikzset{
    checkbox-sc/.style={
      right=of lab\thecsvrow
    },
    vallab-sc/.style={
      text width=4cm,
      align=left,
    },
    checkbox-mc/.style={
    },
    vallab-mc/.style={
      above=of box\thecsvrow,
      text width=1.4cm,
      align=center,
    },
    varlab-mc/.style={
      text width=4cm,
      align=left,
    },
    node distance= 0mm,
  }
\fi
```

4.21 Package Errors

`\AMC@error@explain` Error to display if `\explain` command is used outside question like environments

```
\def\AMC@error@explain{\PackageError{automultiplechoice}{
    Command \protect\explain\space can only be used inside\MessageBreak question like environments}{Some
}}
```

4.22 Optional features

This package tries to see if optional packages `environ` and `etex` are loadable, and load them if possible. This behaviour can be cancelled by using `plain` option.

```
\ifAMC@plain
\else
  \IfFileExists{environ.sty}{\RequirePackage{environ}}{}
  \ifx\eTeXversion\@undefined
  \else
    \RequirePackage{etex}
  \fi
\fi
```

`examcopy` (*env.*) Then, if `environ` package is loaded and defines command `\NewEnviron`, environment `examcopy` is defined.

Environment `{examcopy}[`$\langle n\rangle$`]` does the same as command `onecopy`: it encloses LaTeX code which makes *one* exam copy. Optional argument $\langle n\rangle$ gives the number of desired copies – this can also be modified redefinig `\AMCNombreCopies`.

```
\@ifpackageloaded{environ}{%
```

```
  \ifx\NewEnviron\undefined\PackageWarning{automultiplechoice}%
  {Package environ loaded but too old version:
    environnement examcopy/copieexamen will NOT be defined.}%
  \else\NewEnviron{examcopy}[1][5]{\onecopy{#1}{\BODY}}\fi}%
{\PackageWarning{automultiplechoice}%
  {Package environ not loaded: environnement
    examcopy/copieexamen will NOT be defined.}}
```

4.23 Use with recent LuaTeX versions

In recent LuaTeX versions, the commands `pdfsavepos`, `pdflastxpos` and `pdflastypos` has been renamed, stripping the `pdf` part. The following code tries to detect this situation and make the bindings between the old and new command names.

```
\ExplSyntaxOn

\cs_if_exist:NTF \pdfsavepos { } {
  \cs_if_exist:NTF \savepos { \cs_new_eq:NN \pdfsavepos \savepos } { }
}
\cs_if_exist:NTF \pdflastxpos { } {
  \cs_if_exist:NTF \lastxpos { \cs_new_eq:NN \pdflastxpos \lastxpos } { }
}
\cs_if_exist:NTF \pdflastypos { } {
  \cs_if_exist:NTF \lastypos { \cs_new_eq:NN \pdflastypos \lastypos } { }
}
```

In some situations, the *page* dimensions are different from the *paper* dimensions. This must be taken into account when computing coordinates.

```

\cs_if_exist:NTF \pdfpagewidth { } {
  \cs_new_eq:NN \pdfpagewidth \paperwidth
}
\cs_if_exist:NTF \pdfpageheight { } {
  \cs_new_eq:NN \pdfpageheight \paperheight
}

\ExplSyntaxOff
```

4.24 External control

`\SujetExterne`
`\ScoringExterne`
`\CorrigeExterne`
`orrigeIndivExterne`
`NoWatermarkExterne`

Some of the package options can be controlled defining *xxx*`Externe` commands. For example, the following command will format the subject document, whatever options are used in the LaTeX file:

```
pdflatex '\nonstopmode\def\SujetExterne{1}\def\NoWatermarkExterne{1}\input{mcq.tex}'
```

```
\ifx\SujetExterne\undefined\else
\message{***SUJET***^^J}
\AMC@calibrationtrue\AMC@correcfalse\AMC@correcheadfalse\AMC@watermarkfalse
\fi
\ifx\ScoringExterne\undefined\else
\message{***SCORING***^^J}
\AMC@calibrationtrue\AMC@correcfalse\AMC@correcheadfalse\AMC@watermarkfalse\AMC@invisibletrue
\fi
```

```
\ifx\CorrigeExterne\undefined\else
\message{***CORRIGE***^^J}
\AMC@calibrationfalse\AMC@correcheadtrue\AMC@correctrue\AMC@watermarkfalse
\fi
\ifx\CorrigeIndivExterne\undefined\else
\message{***CORRIGE***^^J}
\AMC@calibrationtrue\AMC@correcheadfalse\AMC@correctrue\AMC@watermarkfalse
\fi
\ifx\CatalogExterne\undefined\else
\message{***CATALOG***^^J}
\AMC@catalogMode
\fi
\ifx\NoWatermarkExterne\undefined\else
\AMC@watermarkfalse
\fi
\ifx\codeDigitExterne\undefined\else
\AMCcodeID@@{\codeDigitExterne}
\fi
```

4.25 Page layout

The following code sets the correct page layout to have room for signs for scan analysis, and prepares watermark printing:

```
\@ifpackageloaded{geometry}{}{\usepackage{geometry}}
\ifAMC@pagelayout
  \ifAMC@correchead
    \geometry{hmargin=3cm,vmargin={1cm,1cm},includeheadfoot,headheight=1cm,footskip=1cm}
  \else
    \geometry{hmargin=3cm,headheight=2cm,headsep=.3cm,footskip=1cm,top=3.5cm,bottom=2.5cm}
  \fi
  \ifAMC@watermark
    \ifAMC@correchead\else
      \def\AMC@note{\begin{minipage}{0.65\linewidth}
          \AMC@LR{\textcolor{blue}{\AMC@loc@message}}
        \end{minipage}
      }
    \fi
  \fi
\fi
```

4.26 Initialisation

Initialisation of the check counter:

```
\AMCid@check=\AMCid@checkmax
```

Telling outside if separate answer sheet, and boxes labelling, are requested:

```
\ifAMC@ensemble\AMCmessage{VAR:ensemble=1}\fi
\ifAMC@inside@box\AMCmessage{VAR:insidebox=1}\fi
\ifAMC@outside@box\AMCmessage{VAR:outsidebox=1}\fi
\ifAMC@postcorrect\AMCmessage{VAR:postcorrect=1}\fi
```

Preparing writing to `.xy` file :

```
\ifAMC@calibration
```

```
\newwrite\AMC@XYFILE%
\immediate\openout\AMC@XYFILE\jobname.xy%
\immediate\write\AMC@XYFILE{\string\version{\AMC@VERSION}}
\immediate\write\AMC@XYFILE{\string\with{codedigit=\AMCcodeID@mode}}
\immediate\write\AMC@XYFILE{\string\with{version=\AMC@VERSION}}
\immediate\write\AMC@XYFILE{\string\with{ensemble=\ifAMC@ensemble yes\else no\fi}}
\immediate\write\AMC@XYFILE{\string\with{insidebox=\ifAMC@inside@box yes\else no\fi}}
\immediate\write\AMC@XYFILE{\string\with{outsidebox=\ifAMC@outside@box yes\else no\fi}}
\immediate\write\AMC@XYFILE{\string\with{postcorrect=\ifAMC@postcorrect yes\else no\fi}}
\immediate\write\AMC@XYFILE{\string\with{extractonly=\ifAMC@extractOnly yes\else no\fi}}
\immediate\write\AMC@XYFILE{\string\with{lang=\AMC@lang@code}}
\ifx\AMCNombreCopies\undefined%
\immediate\write\AMC@XYFILE{\string\with{ncopies=default}}%
\else%
\immediate\write\AMC@XYFILE{\string\with{ncopies=\AMCNombreCopies}}%
\fi%
\fi
```

4.27 French command names

For backward compatibility, a lot of commands have their french counterpart:

```
\let\reponses=\choices\let\endreponses=\endchoices
\let\reponseshoriz=\choiceshoriz\let\endreponseshoriz=\endchoiceshoriz
\let\reponsesperso=\choicescustom\let\endreponsesperso=\endchoicescustom
\let\bonne=\correctchoice
\let\mauvaise=\wrongchoice
\let\bareme=\scoring
\let\baremeDefautM=\scoringDefaultM
\let\baremeDefautS=\scoringDefaultS
\def\exemplaire{\AMC@loc@FR\onecopy}
\@ifpackageloaded{environ}{%
  \let\copieexamen=\examcopy\let\endcopieexamen=\endexamcopy}{}
\let\melangegroupe=\shufflegroup
\let\restituegroupe=\insertgroup
\let\alafin=\lastchoices
\let\formulaire=\AMCform
\let\AMCdebutFormulaire=\AMCformBegin
\let\champnom=\namefield
\let\choixIntervalles=\AMCIntervals
```

5 Outputs

In the `.xy` file, `0/`$\langle n\rangle$ means student sheet number 0 (there is only one "student sheet" numbered 0 for this document as we did not use `\onecopy`) and page number $\langle n\rangle$ inside this student sheet. Then, each instance of the `\tracepos` command shows x and y positions as arguments #2 and #3 (unit is `sp`, such that $65536 \times 72.27\,\mathtt{sp}$ is one inch). One has to take min and max of the x-values to determine the left and right position of the box, and min and max values of y-values to determine top and bottom position of the box.

5.1 `namefield` command

Lines in the `.xy` file from a `\namefield` command:

```
\tracepos{0/35:__zone:id:__n}{0sp}{39236125sp}{square}
\tracepos{0/35:__zone:id:__n}{5873801sp}{0sp}{square}
\tracepos{0/35:__zone:id:__n}{15861297sp}{0sp}{square}
\tracepos{0/35:__zone:id:__n}{0sp}{36250947sp}{square}
```

5.2 AMCboxedchar command

Lines in the .xy file from a \AMCboxedchar command:

```
\tracepos{0/35:test}{22855914sp}{23116900sp}{square}
\tracepos{0/35:test}{23561334sp}{22411480sp}{square}
```

5.3 AMCcode command

Lines in the .xy file from a \AMCcode command. Here, code[⟨n⟩]:⟨q⟩,⟨v⟩ relates to digit number ⟨n⟩ from the right (⟨n⟩=1 for units, ⟨n⟩=2 for tens, ⟨n⟩=3 for hundreds and so on), question number ⟨q⟩ (\AMCcode uses a fake question; this number can be ignored), and value ⟨v⟩-1 (box number ⟨v⟩ for the digit).

```
\tracepos{0/60:case:code[5]:16,1}{24875504sp}{27248508sp}{square}
\tracepos{0/60:case:code[5]:16,1}{25580924sp}{26543088sp}{square}
\boxchar{0/60:case:code[5]:16,1}{A}
\tracepos{0/60:case:code[5]:16,2}{24875504sp}{26134396sp}{square}
\tracepos{0/60:case:code[5]:16,2}{25580924sp}{25428976sp}{square}
\boxchar{0/60:case:code[5]:16,2}{B}
\tracepos{0/60:case:code[5]:16,3}{24875504sp}{25020284sp}{square}
\tracepos{0/60:case:code[5]:16,3}{25580924sp}{24314864sp}{square}
\boxchar{0/60:case:code[5]:16,3}{C}
\tracepos{0/60:case:code[5]:16,4}{24875504sp}{23906172sp}{square}
\tracepos{0/60:case:code[5]:16,4}{25580924sp}{23200752sp}{square}
\boxchar{0/60:case:code[5]:16,4}{D}
\tracepos{0/60:case:code[4]:17,1}{26540303sp}{29476732sp}{square}
\tracepos{0/60:case:code[4]:17,1}{27245723sp}{28771312sp}{square}
\boxchar{0/60:case:code[4]:17,1}{0}
\tracepos{0/60:case:code[4]:17,2}{26540303sp}{28362620sp}{square}
\tracepos{0/60:case:code[4]:17,2}{27245723sp}{27657200sp}{square}
\boxchar{0/60:case:code[4]:17,2}{1}
\tracepos{0/60:case:code[4]:17,3}{26540303sp}{27248508sp}{square}
\tracepos{0/60:case:code[4]:17,3}{27245723sp}{26543088sp}{square}
\boxchar{0/60:case:code[4]:17,3}{2}
\tracepos{0/60:case:code[4]:17,4}{26540303sp}{26134396sp}{square}
\tracepos{0/60:case:code[4]:17,4}{27245723sp}{25428976sp}{square}
\boxchar{0/60:case:code[4]:17,4}{3}
\tracepos{0/60:case:code[4]:17,5}{26540303sp}{25020284sp}{square}
\tracepos{0/60:case:code[4]:17,5}{27245723sp}{24314864sp}{square}
\boxchar{0/60:case:code[4]:17,5}{4}
\tracepos{0/60:case:code[4]:17,6}{26540303sp}{23906172sp}{square}
\tracepos{0/60:case:code[4]:17,6}{27245723sp}{23200752sp}{square}
\boxchar{0/60:case:code[4]:17,6}{5}
\tracepos{0/60:case:code[3]:18,1}{28032160sp}{29476732sp}{square}
```

```
\tracepos{0/60:case:code[3]:18,1}{28737580sp}{28771312sp}{square}
\boxchar{0/60:case:code[3]:18,1}{0}
\tracepos{0/60:case:code[3]:18,2}{28032160sp}{28362620sp}{square}
\tracepos{0/60:case:code[3]:18,2}{28737580sp}{27657200sp}{square}
\boxchar{0/60:case:code[3]:18,2}{1}
\tracepos{0/60:case:code[3]:18,3}{28032160sp}{27248508sp}{square}
\tracepos{0/60:case:code[3]:18,3}{28737580sp}{26543088sp}{square}
\boxchar{0/60:case:code[3]:18,3}{2}
\tracepos{0/60:case:code[3]:18,4}{28032160sp}{26134396sp}{square}
\tracepos{0/60:case:code[3]:18,4}{28737580sp}{25428976sp}{square}
\boxchar{0/60:case:code[3]:18,4}{3}
\tracepos{0/60:case:code[3]:18,5}{28032160sp}{25020284sp}{square}
\tracepos{0/60:case:code[3]:18,5}{28737580sp}{24314864sp}{square}
\boxchar{0/60:case:code[3]:18,5}{4}
\tracepos{0/60:case:code[3]:18,6}{28032160sp}{23906172sp}{square}
\tracepos{0/60:case:code[3]:18,6}{28737580sp}{23200752sp}{square}
\boxchar{0/60:case:code[3]:18,6}{5}
\tracepos{0/60:case:code[2]:19,1}{29524017sp}{29476732sp}{square}
\tracepos{0/60:case:code[2]:19,1}{30229437sp}{28771312sp}{square}
\boxchar{0/60:case:code[2]:19,1}{0}
\tracepos{0/60:case:code[2]:19,2}{29524017sp}{28362620sp}{square}
\tracepos{0/60:case:code[2]:19,2}{30229437sp}{27657200sp}{square}
\boxchar{0/60:case:code[2]:19,2}{1}
\tracepos{0/60:case:code[2]:19,3}{29524017sp}{27248508sp}{square}
\tracepos{0/60:case:code[2]:19,3}{30229437sp}{26543088sp}{square}
\boxchar{0/60:case:code[2]:19,3}{2}
\tracepos{0/60:case:code[2]:19,4}{29524017sp}{26134396sp}{square}
\tracepos{0/60:case:code[2]:19,4}{30229437sp}{25428976sp}{square}
\boxchar{0/60:case:code[2]:19,4}{3}
\tracepos{0/60:case:code[2]:19,5}{29524017sp}{25020284sp}{square}
\tracepos{0/60:case:code[2]:19,5}{30229437sp}{24314864sp}{square}
\boxchar{0/60:case:code[2]:19,5}{4}
\tracepos{0/60:case:code[2]:19,6}{29524017sp}{23906172sp}{square}
\tracepos{0/60:case:code[2]:19,6}{30229437sp}{23200752sp}{square}
\boxchar{0/60:case:code[2]:19,6}{5}
\tracepos{0/60:case:code[1]:20,1}{31015874sp}{29476732sp}{square}
\tracepos{0/60:case:code[1]:20,1}{31721294sp}{28771312sp}{square}
\boxchar{0/60:case:code[1]:20,1}{0}
\tracepos{0/60:case:code[1]:20,2}{31015874sp}{28362620sp}{square}
\tracepos{0/60:case:code[1]:20,2}{31721294sp}{27657200sp}{square}
\boxchar{0/60:case:code[1]:20,2}{1}
\tracepos{0/60:case:code[1]:20,3}{31015874sp}{27248508sp}{square}
\tracepos{0/60:case:code[1]:20,3}{31721294sp}{26543088sp}{square}
\boxchar{0/60:case:code[1]:20,3}{2}
\tracepos{0/60:case:code[1]:20,4}{31015874sp}{26134396sp}{square}
\tracepos{0/60:case:code[1]:20,4}{31721294sp}{25428976sp}{square}
\boxchar{0/60:case:code[1]:20,4}{3}
\tracepos{0/60:case:code[1]:20,5}{31015874sp}{25020284sp}{square}
\tracepos{0/60:case:code[1]:20,5}{31721294sp}{24314864sp}{square}
```

```
\boxchar{0/60:case:code[1]:20,5}{4}
\tracepos{0/60:case:code[1]:20,6}{31015874sp}{23906172sp}{square}
\tracepos{0/60:case:code[1]:20,6}{31721294sp}{23200752sp}{square}
\boxchar{0/60:case:code[1]:20,6}{5}
```

Contents

1 Introduction **1**

2 Samples **1**
2.1 Standard layout . . . 4
2.2 Separate answer sheet . . . 5
2.3 Without markers . . . 6

3 Usage **10**
3.1 Package options . . . 10
3.2 Questions and answers . . . 11
3.3 Scoring . . . 13
3.4 Groups of questions . . . 14
3.5 Students identification . . . 15
3.6 Separate answer sheet . . . 16
3.7 Random computation questions . . . 17
3.8 French command names . . . 20
3.9 Customisation . . . 21
3.9.1 Boxes . . . 21
3.9.2 Codes . . . 22
3.9.3 Answers . . . 22

4 Implementation **23**
4.1 Variables . . . 23
4.2 Dimensions . . . 26
4.3 Human readable sheet ID position . . . 26
4.4 Localisation . . . 27
4.4.1 English . . . 27
4.4.2 Catalan . . . 28
4.4.3 Dutch . . . 28
4.4.4 French . . . 28
4.4.5 German . . . 29
4.4.6 Italian . . . 29
4.4.7 Norwegian . . . 29
4.4.8 Portuguese . . . 30
4.4.9 Spanish . . . 30
4.4.10 Japanese . . . 30
4.4.11 Other languages . . . 31
4.5 Interaction with other packages . . . 31
4.5.1 cleveref . . . 31
4.6 Random . . . 31
4.6.1 Random pseudo-generator . . . 31
4.6.2 Uniform random deviates . . . 31
4.6.3 Tokens shuffling . . . 32
4.7 Keys numbering . . . 32
4.8 Boxes . . . 33
4.8.1 Character logging . . . 33
4.8.2 Position logging . . . 33
4.8.3 Boxes to be checked by students . . . 35

4.8.4 Scoring zones . . . 40
4.8.5 Binary boxes . . . 41
4.9 Checking Environment . . . 42
4.10 Handling groups of questions . . . 43
4.11 Questions . . . 47
4.11.1 Managing answers . . . 47
4.11.2 Separate answer sheet . . . 48
4.11.3 Formatting answers . . . 53
4.11.4 Score zones . . . 55
4.11.5 Formatting questions . . . 57
4.11.6 Explanations . . . 59
4.12 Scoring . . . 60
4.13 Numerical data . . . 60
4.13.1 Codes . . . 60
4.13.2 Numerical questions . . . 64
4.13.3 Intervals . . . 78
4.14 Open questions . . . 78
4.15 Boxes with letters only . . . 81
4.16 Page formatting . . . 82
4.16.1 Watermark . . . 82
4.16.2 Signs for scan analysis . . . 83
4.17 Defining a single exam copy content . . . 84
4.18 Pre-association . . . 86
4.19 Package options . . . 86
4.20 Survey add-on . . . 88
4.21 Package Errors . . . 89
4.22 Optional features . . . 89
4.23 Use with recent LuaTeX versions . . . 90
4.24 External control . . . 90
4.25 Page layout . . . 91
4.26 Initialisation . . . 91
4.27 French command names . . . 92

5 Outputs **92**
5.1 `namefield` command . . . 92
5.2 `AMCboxedchar` command . . . 93
5.3 `AMCcode` command . . . 93

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

\" 150, 156
\@aucune 843, 847, 848
\@firstoftwo 661
\@ifl@t@r 3, 2361
\@ifstar 1080, 1081
\@secondoftwo 663
\@skiphyperreffalse . . . 1262
\@skiphyperreftrue 1258, 1262
\@tempboxa . 2328, 2332, 2333
\@tempdimb 2317, 2319
\@tempdimc 2318, 2319
\␣ 1298
\~ 191, 192, 197, 198

A

\aa 177
\AddToHook 2362
\alafin *21*, 2732
\amc 870, 881, 883, 885, 886, 888, 891, 892, 894, 897, 898, 901, 902, 903, 904, 906, 908, 909, 911, 913, 914, 916, 918, 919, 922, 923, 925, 926, 928, 929, 931, 932, 934, 935, 937, 940, 941, 943, 945, 949, 950, 951, 952, 1363, 1364, 1365, 1366, 1367, 1368, 1369, 1371, 1378, 1381, 1386, 1390, 1398, 1401, 1402, 1403, 1406, 1411, 1413, 1416, 1417, 1418, 1419, 1421, 1423, 1425, 1426, 1428, 1429, 1433, 1434, 1435, 1437, 1438, 1439, 1441, 1446, 1448, 1450, 1458, 1463, 1464, 1465, 1467, 1469, 1472, 1473, 1474, 1475, 1477, 1479, 1482, 1483, 1485, 1486, 1488, 1489, 1490, 1491, 1511, 1512, 1517, 1518, 1519, 1520, 1521, 1522, 1527, 1529, 1530, 1531, 1532, 1533, 1535, 1537, 1538, 1539, 1540, 1543, 1550, 1551, 1552, 1554, 1556, 1557, 1559, 1561, 1563, 1566, 1567, 1569, 1571, 1572, 1573, 1575, 1581, 1586, 1598, 1632, 1634, 1636, 1638, 1640, 1643, 1646, 1649, 1652, 1655, 1658, 1661, 1664, 1667, 1669, 1671, 1674, 1683, 1685, 1693, 1697, 1698, 1699, 1705, 1707, 1719, 1720, 1721, 1722, 1723, 1724, 1725, 1726, 1729, 1732, 1736, 1739, 1742, 1746, 1749, 1753, 1756, 1761, 1763, 1765, 1772, 1774, 1777, 1786, 1787, 1788, 1789, 1791, 1793, 1796, 1801, 1802, 1804, 1808, 1810, 1813, 1814, 1815, 1816, 1818, 1819, 1820, 1823, 1826, 1831, 1833, 1836, 1840, 1841, 1842, 1844, 1847, 1852, 1858, 1860, 1862, 1865, 1873, 1874, 1881, 1882, 1884, 1885, 1886, 1888, 1891, 1892, 1895, 1896, 1898, 1904, 1908, 1909, 1911, 1914, 1915, 1922, 1932, 1933, 1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941, 1943, 1948, 1949, 1952, 1958, 1960, 1962, 1963, 1966, 1967, 1971, 1972, 1973, 1975, 1976, 1978, 1979, 1980, 1981, 1982, 1984, 1985, 1988, 1990, 1995, 1996, 1997, 1998, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2009, 2010, 2011, 2012, 2016, 2018, 2020, 2023, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2036, 2037, 2038, 2039, 2043, 2044, 2046, 2052, 2059, 2064, 2066, 2067, 2073, 2082, 2084, 2085, 2086, 2087, 2091, 2092, 2093, 2097, 2098, 2099, 2100, 2102, 2103, 2104, 2117, 2119, 2125, 2134, 2137, 2138, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2152, 2153, 2156
\AMC@addpages 2445, 2447, 2449
\AMC@affecte 281, 1273
\AMC@amclog 19
\AMC@answerBox 480
\AMC@answerBox@ . . . 361, 488, 535, 537, 553, 555, 621, 623, 631, 633
\AMC@binaryBoxes 654
\AMC@binaryCode . . . 604, 655, 2348, 2349, 2350

\AMC@box 522, 1142, 1143, 1147, 2589, 2590
\AMC@boxcolor 379, 505, 2239
\AMC@boxcolor@ 379, 380, 381, 383, 401, 410, 422, 426, 443
\AMC@boxeddown 393, 493, 504
\AMC@boxedheight 406, 420, 421, 423, 424, 427, 428, 472, 495, 499, 500, 512, 515
\AMC@boxedrule 398, 420, 421, 422, 491, 502
\AMC@boxedwidth 409, 420, 421, 423, 424, 427, 428, 472, 494, 499, 501, 512, 513, 1043, 1809
\AMC@catalogMode 2505, 2513, 2672
\AMC@CBtaille 602, 2376, 2519
\AMC@checkedbox 361, 382, 385, 435, 437, 447, 453, 463, 621, 623, 1142, 1710, 1730, 1742, 1746, 2589
\AMC@chiffres 1351
\AMC@coli 1008, 1012, 1013, 1014
\AMC@crosschar 404, 507
\AMC@crossrule 426, 492, 508
\AMC@definitnumero . 282, 286
\AMC@draw@crossfalse . . 384
\AMC@draw@crosstrue . . . 386
\AMC@error@explain 1337, 1339, 2615
\AMC@fillcolor@ 382, 400, 401, 417, 422
\AMC@fin@rep . . 830, 1092, 1096, 1100, 1103, 1106
\AMC@formBox 522
\AMC@formBox@ 522
\AMC@formcolspecs 1009, 1063
\AMC@formspecs . . 1000, 1015
\AMC@fullGroupsfalse . . . 36
\AMC@fullGroupstrue 2483, 2509
\AMC@if@separate@question 881, 956, 961, 969, 975, 1496, 2129, 2275, 2311
\AMC@imax . 715, 727, 728, 729
\AMC@intituleHead 2344, 2365, 2510
\AMC@keepmemoryfalse . . 2435
\AMC@keepmemorytrue . . . 997
\AMC@keyBox@ 551, 1389, 1712, 1715
\AMC@keys@line 2512
\AMC@keys@next 2511
\AMC@lang@code 2480, 2515, 2516, 2712
\AMC@loc@CA 107
\AMC@loc@catalog 101, 114, 128, 141, 155, 169, 182, 194, 208, 220, 2510
\AMC@loc@corrected 100, 113, 127, 140, 154, 168, 181, 193, 207, 219, 2344
\AMC@loc@DE 148
\AMC@loc@draft . . 95, 108, 122, 135, 149, 163, 176, 189, 202, 215, 2315
\AMC@loc@ES 201
\AMC@loc@explain 102, 115, 142, 156, 195, 221, 1337
\AMC@loc@FR . . 134, 2515, 2727
\AMC@loc@IT 162
\AMC@loc@JA 214
\AMC@loc@message 96, 109, 123, 136, 150, 164, 177, 190, 203, 216, 2690
\AMC@loc@namesurname . . . 106, 119, 132, 146, 160, 173, 186, 199, 212
\AMC@loc@NL 121
\AMC@loc@NO 175
\AMC@loc@none . . 103, 116, 129, 143, 157, 170, 183, 196, 209, 222, 843
\AMC@loc@PT 188
\AMC@loc@q 99, 112, 126, 139, 153, 167, 180, 192, 206, 218, 1265
\AMC@loc@qf 98, 111, 125, 138, 152, 166, 179, 191, 205, 217, 852
\AMC@loc@question 104, 117, 130, 144, 158, 171, 184, 197, 210, 223, 229
\AMC@loc@questions 105, 118, 131, 145, 159, 172, 185, 198, 211, 224, 229
\AMC@logchar 289, 390
\AMC@logfile . 19, 20, 21, 290
\AMC@LR 23, 392, 2320, 2371, 2372, 2373, 2374, 2375, 2690
\AMC@makeovalbox 415, 434, 435, 436, 437
\AMC@mem@add 911, 957, 964, 971, 978, 1497
\AMC@mem@add@ifneeded 868, 1305
\AMC@mem@addsingle@ifneeded 960, 1073, 1075, 1077, 1079
\AMC@mem@addvar 916
\AMC@mem@aid 935, 970
\AMC@mem@answer 967, 1107, 1114, 1115, 1116, 1121, 1127
\AMC@mem@category . 929, 963, 2129, 2275, 2311
\AMC@mem@clear . . . 886, 2437
\AMC@mem@next . 906, 962, 976
\AMC@mem@openQuestion 974, 1304
\AMC@mem@qidaffname 923, 977
\AMC@mem@show . 950, 987, 996
\AMC@mem@show@filter 951, 991
\AMC@mn@leftmargin 1153, 1161, 1164, 1170
\AMC@mn@rightmargin 1154, 1162, 1165, 1168
\AMC@mn@sep . 1152, 1168, 1170
\AMC@mn@test 1151, 1158, 1167
\AMC@multi@report 1416, 2443
\AMC@multiclear . 1421, 2434
\AMC@NCBcheck 598, 2350, 2519
\AMC@NCBetud . 598, 2348, 2518
\AMC@NCBpage . 598, 2349, 2518
\AMC@new@savebox 362, 2544, 2548, 2549, 2550, 2551
\AMC@note 2345, 2346, 2384, 2394, 2689
\AMC@numeric@alsocorrect 1689, 2055, 2059, 2069, 2073
\AMC@numeric@keepas 1687, 2048, 2049
\AMC@numeric@scoreapprox 1679, 2014, 2024
\AMC@numeric@scoreexact 1677, 2013, 2019

\AMC@numeric@scorewrong 1681, 2021, 2024
\AMC@numerotation 281, 285, 286
\AMC@outside@sep .. 497, 503
\AMC@oval@radius 422, 496, 513, 515
\AMC@ovalbox@ 436, 450, 2550
\AMC@ovalbox@F 437, 448, 2551
\AMC@ovalbox@R 433, 434, 456, 2548
\AMC@ovalbox@RF 435, 454, 2549
\AMC@pageHook 2355, 2362, 2363
\AMC@pagepos 295, 2356
\AMC@premierecopie 603, 2418, 2421, 2520
\AMC@prepare .. 284, 287, 288
\AMC@prepare@element ... 678, 686, 801
\AMC@printformoutside@false 526
\AMC@printformoutside@true 528, 529
\AMC@printkeyoutside@false 543
\AMC@printkeyoutside@true 545, 547
\AMC@qaff 853, 1255, 1298, 1299, 1333
\AMC@save@box 363, 418, 2545
\AMC@setcolors@ 378, 399, 416, 441
\AMC@shape@form 475
\AMC@shape@form@base 462, 476
\AMC@shape@form@ticked 464, 466, 469
\AMC@shape@none 479
\AMC@shape@oval 440
\AMC@shape@square 397
\AMC@shapename 295, 303, 312, 321, 498
\AMC@shapename@ 295, 393, 509
\AMC@shapeprepare 509, 517, 2552
\AMC@shapeprepare@form 461
\AMC@shapeprepare@none 478
\AMC@shapeprepare@oval 432
\AMC@shapeprepare@square 396
\AMC@shuffletoks 262, 748, 751, 823
\AMC@smashbox .. 366, 369, 370, 371, 372, 373, 376
\AMC@smashboxheight 367, 370, 371, 372, 375
\AMC@smashcentered 368, 410, 411, 443, 444
\AMC@SR 231, 233, 236, 237, 238, 239, 241, 243, 244, 2428
\AMC@SR@count 236, 237, 239, 247, 251, 254, 257, 258, 259, 260
\AMC@SR@time 248, 2496
\AMC@SRadvance 234, 241, 242
\AMC@SRbit 241
\AMC@SRconst .. 232, 236, 239
\AMC@SRmax 247, 275
\AMC@SRnextByte 247
\AMC@SRnum 249, 250, 252, 253, 257, 260, 275, 276
\AMC@SRset 233, 245, 246, 248
\AMC@SRtest 242, 253
\AMC@SRvalue 244
\AMC@stepQuestion 1255, 1290, 1330
\AMC@sti 262, 270, 274, 277, 278
\AMC@stil 263, 271, 272, 273, 275, 279
\AMC@surveyfalse 2529
\AMC@surveytrue 2530
\AMC@sz@box 1174, 1176, 1178, 1181, 1224
\AMC@sz@callin 1186, 1191, 1199, 1201, 1266, 1300, 1301
\AMC@sz@callin@question 1176
\AMC@sz@callout 1185, 1190, 1194, 1196, 1293, 1294
\AMC@sz@callout@margin 1178
\AMC@sz@callout@margins 1181
\AMC@sz@depth 1174, 1184, 1189
\AMC@sz@height 1174, 1183, 1188
\AMC@sz@init@margins .. 1180
\AMC@sz@width 1174, 1182, 1187
\AMC@sza@box 1209, 1217, 1220
\AMC@sza@callin 862, 863, 1231, 1236, 1244, 1246
\AMC@sza@callin@margin 1221
\AMC@sza@callin@margins 1225
\AMC@sza@callin@none .. 1213
\AMC@sza@callin@question 1217
\AMC@sza@callout 858, 859, 1230, 1235, 1239, 1241
\AMC@sza@callout@margin 1220
\AMC@sza@callout@margins 1224
\AMC@sza@callout@none . 1212
\AMC@sza@callout@question 1216
\AMC@sza@depth 1209, 1229, 1234
\AMC@sza@height 1209, 1228, 1233
\AMC@sza@init@margin .. 1219
\AMC@sza@init@margins . 1223
\AMC@sza@init@none 1211
\AMC@sza@init@question 1215
\AMC@sza@width 1209, 1227, 1232
\AMC@tempenv 659, 660
\AMC@tf@colW 1042, 1043, 1046, 1052
\AMC@tf@idtext 1023, 1028, 1045, 1048
\AMC@tf@idtitle 1001, 1024, 1029
\AMC@tf@nanswers . 1026, 1039, 1040, 1052, 1063
\AMC@tf@ncols 1019, 1027, 1051, 1059, 1063, 1067
\AMC@tfaw 1022, 1055, 1057, 1058
\AMC@tfbox . 1025, 1048, 1049
\AMC@tfcolw .. 1021, 1052, 1053, 1054, 1057, 1058
\AMC@tracebox 295, 349, 442, 468, 586, 588, 591, 1123, 2338
\AMC@tracechar 337, 539, 557
\AMC@tracepos 296, 407, 413, 1109, 1112, 1296, 1305
\AMC@traceposx 305, 325
\AMC@traceposy 314, 324, 326
\AMC@unnumero 281
\AMC@use@box 364, 448, 450, 454, 456, 2546
\AMC@VERSION 2704, 2706
\AMC@watertext .. 2315, 2326
\AMC@XYFILE 298, 307, 316, 328, 333, 334, 335, 339, 345, 347, 1276, 1299, 1413, 2427, 2471, 2476, 2477,

2702, 2703, 2704, 2705, 2706, 2707, 2708, 2709, 2710, 2711, 2712, 2714, 2716
\AMCaddpagesto 2445
\AMCanswer 1127, 1128, 2228, 2290
\AMCassociation 2469
\AMCbeforeQuestion 1264, 1292, 1373
\AMCbeginAnswer 1099, 1128, 2227, 2289
\AMCbeginQuestion 1264, 1298, 1313, 1319, 1333, 1372
\AMCbin@begin 653
\AMCbin@did 608, 616, 619, 621, 626, 629, 631, 637
\AMCbin@digit 606, 639, 642, 648, 651
\AMCbin@hsep 610, 613, 617, 627
\AMCbin@id 607, 612, 620, 621, 630, 631
\AMCbin@ndigits 609, 611, 648, 651
\AMCbin@number 605, 638, 641, 643, 645
\AMCbin@one 615, 643
\AMCbin@sequence . . 604, 638, 643, 644, 649, 650
\AMCbin@style 614, 638
\AMCbin@zero . . 625, 644, 649
\AMCbloc 1271
\AMCbo@help . 2279, 2280, 2293
\AMCboHide 2306, 2312
\AMCboOpts 2283
\AMCboShow 2284, 2312
\AMCbotextGoto . . 2278, 2307
\AMCboxColor 520, 2376
\AMCboxDimensions . 521, 638
\AMCBoxedAnswers 1082, 2487
\AMCBoxOnly 2278
\AMCboxOutsideLetter 522, 2229, 2291
\AMCboxStyle *21*, 491, 2487, 2537
\AMCcercle 2334, 2337
\AMCchoiceLabel 480, 561, 1005, 1068
\AMCchoiceLabelFormat 394, 480
\AMCcleardoublepage *16*, 2454
\AMCcode 1490
\AMCcodeGrid *15*, 1351
\AMCcodeGridInt . . . *15*, 1351
\AMCcodeH 1491
\AMCcodeHspace 1352, 1393, 1452
\AMCcodeID 1357, 1432
\AMCcodeID@@ 1356, 1360, 2528, 2678
\AMCcodeID@dot 1355
\AMCcodeID@mode . 1358, 2705
\AMCcodeID@squarebrackets 1354
\AMCcodeVspace 1353, 1391, 1451, 1454
\AMCcompleteMulti . . *12*, 65
\AMCcurrentenv 657, 660, 1289
\AMCdebutFormulaire *21*, 2734
\AMCdecimalPoint 1620, 1805, 1837
\AMCdefault@groupmode 675, 689, 690
\AMCdontAnnotate . 333, 2244
\AMCdontScan 333, 2242
\AMCdump@reponses . 823, 837
\AMCemptybox 1149, 1174, 1209
\AMCendAnswer 1100, 1128, 2227, 2289
\AMCexponent 1622, 2089, 2101
\AMCexternalQuestion . . 1280
\AMCform . *16*, 868, 1069, 2733
\AMCformAfterQuestion 851, 1305
\AMCformAnswer 850, 1005, 1046
\AMCformAnswerA . . . 867, 971
\AMCformatChoices 1494, 2130, 2276, 2312
\AMCformBeforeQuestion 850, 856, 1044
\AMCformBegin . *16*, 868, 2734
\AMCformFilter 989
\AMCformHSpace . . 68, 866, 1001, 1042, 1045, 2290
\AMCformQuestion . 850, 1045
\AMCformQuestionA . 854, 978
\AMCformQuestionN . 853, 861
\AMCformS 868
\AMCformTH 1001, 1020, 1030, 1045, 1047, 1049, 1053
\AMCformVSpace 68, 850, 1044, 1064
\AMCgroup@pre 744, 767
\AMCgrouploop@next 769, 782, 800
\AMCgrouploop@prep 755, 780, 798
\AMCgrouppre@cyclic . . . 738
\AMCgrouppre@fixed 700
\AMCgrouppre@withoutreplacement 716
\AMCgrouppre@withreplacement 707
\AMChorizAnswerSep 1117, 1118, 1126
\AMChorizBoxSep 1119, 1120, 1123, 1125
\AMCid@check 29, 330, 2350, 2357, 2358, 2377, 2381, 2696
\AMCid@checkmax 598, 2358, 2518, 2696
\AMCid@etud . 30, 292, 300, 309, 318, 330, 333, 334, 335, 340, 355, 476, 1299, 2347, 2348, 2377, 2381, 2389, 2417, 2418, 2420, 2428, 2433, 2440, 2441, 2461, 2462, 2471
\AMCid@etudfin 32, 2419, 2420, 2421, 2440
\AMCid@etudstart 31, 2417, 2441
\AMCid@name . 67, 567, 570, 572, 576, 578, 920, 977, 1109, 1112, 1123, 1273, 1276, 1305, 1389, 1712, 1715, 2171
\AMCid@quest 28, 333, 334, 335, 567, 570, 572, 576, 578, 586, 588, 591, 919, 977, 1109, 1112, 1123, 1273, 1275, 1276, 1296, 1299, 1305, 1389, 1713, 1716, 1945, 2210, 2286
\AMCIDBoxesA 2348, 2352
\AMCIDBoxesABC . . 2351, 2378
\AMCIDBoxesB 2349, 2353
\AMCIDBoxesC 2350, 2353
\AMCids@height . 79, 90, 2377
\AMCids@sidefalse . . . 82, 84
\AMCids@sidetrue 86
\AMCids@topfalse 82, 86

\AMCids@toptrue 84
\AMCids@width . . 78, 89, 2380
\AMCidsPosition 76
\AMCidsVar 80
\AMCidsVarN 80, 81
\AMCif@env . . 658, 1337, 1339
\AMCifcategory 952
\AMCinterBquest . . . 73, 1305
\AMCinterBrep 68, 1113
\AMCinterIquest . . . 72, 1305
\AMCinterIrep 68, 1090
\AMCIntervalFormat 2141, 2152
\AMCIntervals *17*, 2141, 2736
\AMClabel 2462, 2467
\AMCload@@reponse . 834, 836, 1142, 1147, 2589
\AMCload@counter 27, 812, 813, 814, 815, 818, 819, 823, 826, 827, 828, 829, 832
\AMCload@reponse . . 817, 836
\AMClocalized 94
\AMCloop@k 721, 727, 731, 733, 754, 762, 771
\AMCmarginNote 1155, 1181, 1224
\AMCmem@elt@cat 926, 938, 942
\AMCmessage 19, 282, 594, 996, 1140, 1146, 1275, 1283, 1284, 1285, 1305, 1309, 1314, 1320, 1347, 1348, 1349, 1350, 1388, 1502, 1709, 1741, 1743, 1748, 1750, 1755, 1757, 1766, 1779, 1795, 1887, 1890, 1923, 1925, 1945, 2010, 2017, 2034, 2036, 2039, 2041, 2044, 2046, 2049, 2053, 2056, 2058, 2065, 2070, 2072, 2112, 2210, 2267, 2286, 2303, 2415, 2433, 2536, 2540, 2587, 2697, 2698, 2699, 2700
\AMCncol@Background 1630, 2081
\AMCncol@Border . 1626, 2081
\AMCncol@BorderWidth 1628, 2080
\AMCneeds@storeboxfalse 2525
\AMCneeds@storeboxtrue 2526
\AMCnobloc 1270, 1377
\AMCnoCompleteMulti *12*, 66, 2226
\AMCNombreCopies 2413, 2414, 2713, 2716
\AMCnoScoreZone . 1254, 1374
\AMCntextGoto . . . 1611, 2121
\AMCntextSign 1618, 1856, 1857
\AMCntextVHead 1612, 1809, 1955
\AMCnum@copies . 33, 2413, 2414, 2415, 2416, 2419
\AMCnum@questions . 824, 826
\AMCnumeric@Hspace 1613, 1625, 1767, 1769, 1805, 1824, 2101
\AMCnumeric@Vspace 1614, 1624, 1740, 1747, 1754, 1781, 1797, 1811, 1845, 1857, 2088, 2090
\AMCnumericChoices . *18*, 1494
\AMCnumericChoicesPlain 2128, 2135
\AMCnumericHide . 2125, 2130
\AMCnumericOpts 1705
\AMCnumericShow . 2117, 2130
\AMCnumero 1256, 2432
\AMCocol@Background 2161, 2168, 2233
\AMCocol@BoxFrameRule 2162, 2186, 2233
\AMCocol@Foreground 2164, 2169, 2238, 2239, 2249, 2252
\AMCocol@FrameRule 2163, 2187, 2260
\AMCOpen 2159
\AMCopen@answer 2165, 2172, 2201, 2217
\AMCopen@boxframerule 2184, 2185, 2232
\AMCopen@boxmargin 2182, 2183, 2231
\AMCopen@contentcommand 2173, 2261
\AMCopen@framerule 2188, 2189, 2259
\AMCopen@Hspace . . 2174, 2175, 2230, 2235, 2247
\AMCopen@LineHeight 2178, 2179, 2200
\AMCopen@Lines 2180, 2181, 2205
\AMCopen@lines 2198
\AMCopen@lineuptext 2167, 2190, 2215, 2218, 2221
\AMCopen@question 2166, 2171, 2234, 2235
\AMCopen@Width 2176, 2177, 2199
\AMCopenHide 2270, 2276
\AMCopenOpts 2197
\AMCopenShow 2208, 2276
\AMCotextGoto . . . 2159, 2271
\AMCotextReserved 2160, 2170, 2246, 2250, 2252
\AMCoutsideLabelFormat 522
\AMCpageref 2465
\AMCpostNquest 74, 2110
\AMCpostOquest 75, 2263
\AMCqlabel 2466
\AMCquestionaff 1255
\AMCquestionNumberfalse 1375
\AMCquestionNumbertrue . 47
\AMCrandomseed 245
\AMCref 2462, 2462
\AMCrep@@count 838, 840, 842
\AMCrep@bloc 1089, 1107
\AMCrep@count 567, 570, 572, 576, 578, 809, 820, 840, 842, 846, 932, 970, 1084, 1109, 1112, 1123, 1132, 1140, 1143, 1146, 1147, 1383, 1387, 1388, 1389, 1495, 1497, 1708, 1709, 1713, 1716, 2586, 2587, 2590
\AMCrep@fini 834, 837, 841, 849
\AMCrep@init . . 830, 1091, 1094, 1098, 1102, 1105
\AMCrep@itemize . 1088, 1107
\AMCrep@ligne . . . 1094, 1107
\AMCrep@nn 810, 1085, 1133, 1134, 1135
\AMCrep@nnmax 811, 1040, 1134, 1135, 2426
\AMCrep@o 831, 833
\AMCrep@perso . . . 1098, 1107
\AMCrep@r 835
\AMCrep@tikz 1102, 1107

\AMCrep@tikzmat . 1105, 1107
\AMCreTick 333, 2243
\AMCrien@deux 817, 834
\AMCscoreZone 582, 1149
\AMCscoreZoneAnswerSheet 1149
\AMCsection 1072
\AMCsectionNumbered 1072, 1080
\AMCsectionStar . 1076, 1080
\AMCsetFoot 2346
\AMCsetScoreZone 1207, 1208, 1254
\AMCsetScoreZoneAnswerSheet 1252, 1253, 1254
\AMCshowSignificantDigits 1559
\AMCsignificantDigits . 1554
\AMCstartWithQuestion 1272, 1281, 1291
\AMCstudentlabel 2462, 2463, 2464, 2465
\AMCStudentNumber 2347
\AMCstudentslistfile . . 2474
\AMCsubjectPageTag 2388, 2393
\AMCsubsection 1072
\AMCsubsectionNumbered 1074, 1081
\AMCsubsectionStar 1078, 1081
\AMCsw@p 262
\AMCsw@p@ 264, 266, 268
\AMCsz@loggedfalse 582
\AMCsz@loggedtrue 595
\AMCtableForm 999
\AMCtext 93
\AMCtok@ik 721, 722, 724, 729, 730, 731, 732, 733, 753, 761, 774, 783, 802
\AMCtok@k 666, 681, 682, 683, 687, 802
\AMCtok@max 667
\AMCtok@size 668, 756, 757, 758, 759, 767, 775, 784, 803
\AMCw@termark . . . 2314, 2359
\AMCw@terprint . . 2314, 2359
amcxyfile (env.) 344
\AMCzone 349, 350
\answer 291, 1131, 2554
answers (option) *10*
asbox (option) *10*
\association 2471
\AtBeginShipout 5, 2363
\AtBeginShipoutUpperLeft 2363
auto (env.) 2554
automarks (option) *11*

B

\bareme *21*, 2724
\baremeDefautM *21*, 2725
\baremeDefautS *21*, 2726
\bf 1005, 1621
bloc (option) *21*
\bonne *21*, 2722
\bool 944, 1366, 1367, 1368, 1390, 1401, 1402, 1425, 1428, 1432, 1439, 1441, 1450, 1698, 1763, 1774, 1789, 1808, 1813, 1854, 1858, 1873, 1886, 1892, 1896, 1911, 1921, 1922, 1962, 1966, 1975, 2016, 2052, 2064, 2082
box (option) *10*
\boxchar 340
\boxput 8, 403, 442

C

\c 949, 1668
calibration (option) . . *5*, *10*
catalog (option) *10*
\CatalogExterne 2670
\champnom *21*, 2735
\char 1514
\CheckBox 469
chiffres (option) *21*
\choices 2719
choices (env.) *12*, 1082
\choicescustom 2721
choicescustom (env.) *12*, 1082
\choiceshoriz 2720
choiceshoriz (env.) . *12*, 1082
\choixIntervalles . *21*, 2736
\cleargroup *14*, 790
\clist 1363, 1369, 1378, 1413, 1426, 1430, 1464, 1473, 1474, 1568, 1571, 1582, 1583, 1667, 1668, 1802, 1803, 1833, 1834, 1934, 1935, 1952, 1960
\codeDigitExterne 2677, 2678
\columnsep 1031, 1054
\columnseprule . . 1010, 1032
completemulti (option) . . *10*
\coordinate 1157
\copieexamen 2729
copieexamen (env.) *21*
\copygroup *14*, 790
\copygroupfrom *14*, 790
correc (option) *21*
correcindiv (option) *21*
\correctchoice *13*, 848, 1131, 2151, 2722
\CorrigeExterne 2654
\CorrigeIndivExterne . . 2654
\crefalias 228
\crefname 229
\cs 881, 885, 886, 888, 891, 894, 897, 901, 906, 908, 911, 913, 916, 918, 922, 923, 925, 928, 929, 931, 934, 935, 937, 940, 949, 950, 951, 952, 1371, 1381, 1386, 1411, 1416, 1418, 1421, 1423, 1463, 1467, 1472, 1477, 1482, 1485, 1488, 1489, 1490, 1491, 1509, 1517, 1527, 1530, 1535, 1539, 1551, 1554, 1556, 1559, 1561, 1567, 1581, 1586, 1598, 1693, 1697, 1705, 1707, 1721, 1736, 1761, 1772, 1787, 1801, 1831, 1852, 1881, 1943, 2117, 2119, 2125, 2134, 2137, 2138, 2146, 2156, 2636, 2637, 2639, 2640, 2642, 2643, 2646, 2647, 2649, 2650
\csvreader 2561, 2566, 2572, 2581

D

\define@boolkey 506, 1033, 1034, 2191, 2192, 2193, 2194, 2195, 2281
\define@choicekey 80, 498, 1192, 1237
digits (option) *10*

\ding 470
\dontannotate 334
\dontscan 333
\dotfill 358, 2203, 2219
\draw . . . 422, 426, 1168, 1170

E

\element *14*, 666
\endchoices 2719
\endchoicescustom 2721
\endchoiceshoriz 2720
\endcopieexamen 2729
\endexamcopy 2729
\endreponses 2719
\endreponseshoriz 2720
\endreponsesperso 2721
ensemble (option) *21*
environments:
amcxyfile 344
auto 2554
choices *12*, 1082
choicescustom . . *12*, 1082
choiceshoriz . . . *12*, 1082
copieexamen *21*
examcopy 2626
question *11*, 1269
question-auto 2554
questionmult . . . *11*, 1269
Questionnaires 2554
questionouverte . . . 1269
reponses *21*
reponseshoriz *21*
reponsesperso *21*
tikz-multi 1082
tikz-single 1082
values 2554
values-auto 2554
variable-auto 2554
variable-multi 1269
variable-single . . . 1269
\evensidemargin . 1162, 1164
\examcopy 2729
examcopy (env.) 2626
\exemplaire *21*, 2727
\exemplairepair 2461
\explain 1335, 2616
\explaincontext 1342
\ExplSyntaxOff 954, 1493, 1610, 2127, 2140, 2158, 2653
\ExplSyntaxOn 868, 1361, 1507, 1615, 2132, 2142, 2634

F

\fancypagestyle 2364, 2369, 2391, 2398
\fbox 2389
\fmtversion 3, 2361
\footrulewidth 2367, 2386, 2396, 2401
\foreach 2580
\formulaire *21*, 2733
\fp 1521, 1529, 1531, 1533, 1537, 1542, 1544, 1546, 1550, 1557, 1562, 1569, 1587, 1590, 1932, 1933, 1971, 1972, 1981, 2009, 2010, 2036, 2144, 2145, 2147, 2148, 2149, 2150, 2152, 2153
francais (option) *10*
fullgroups (option) *11*

G

\group 1513, 1525

H

\he@dbas . . . 2342, 2371, 2372
\he@dhaut 2343, 2375
\he@dtaille . 2339, 2342, 2343
\hfuzz 372
\ht 370, 1150

I

\ifAMC@affichekeys 38, 1297, 1298
\ifAMC@asqbloc . 44, 851, 857
\ifAMC@automarks 62, 983, 2431, 2450, 2457
\ifAMC@calibration 48, 297, 306, 315, 328, 333, 334, 335, 338, 1140, 1146, 1274, 1282, 1299, 1308, 1347, 1348, 1349, 1350, 1388, 1412, 2427, 2433, 2470, 2475, 2587, 2701
\ifAMC@catalog 49, 389
\ifAMC@correc 40, 381, 1142, 1336, 1343, 1711, 2201, 2217, 2589
\ifAMC@correchead 37, 2359, 2405, 2421, 2682, 2688
\ifAMC@draw@cross 365, 404, 425
\ifAMC@ensemble 56, 527, 544, 565, 584, 871, 983, 986, 990, 995, 1298, 1500, 1944, 2110, 2111, 2209, 2214, 2228, 2234, 2266, 2285, 2290, 2293, 2302, 2431, 2697, 2707
\ifAMC@extractOnly 42, 2078, 2711
\ifAMC@fullGroups . . 36, 757
\ifAMC@inside@box 54, 547, 575, 2229, 2291, 2698, 2708
\ifAMC@inside@digit . 57, 481
\ifAMC@invisible . . . 63, 295
\ifAMC@keepmemory 993, 2437
\ifAMC@keysline 39, 1297, 1298
\ifAMC@ordre 34, 831
\ifAMC@outside@box 55, 527, 545, 2699, 2709
\ifAMC@pagelayout 60, 2357, 2404, 2681
\ifAMC@pdfform 64, 353, 2424, 2442, 2535
\ifAMC@plain 50, 2618
\ifAMC@postcorrect 61, 848, 2700, 2710
\ifAMC@printformoutside 525, 534
\ifAMC@printformoutside@ 524, 531
\ifAMC@printkeyoutside 542, 552
\ifAMC@printkeyoutside@ 541, 549
\ifAMC@qbloc . . . 43, 1264, 1296, 1305, 1332, 1334, 2122, 2272, 2308
\ifAMC@rbloc . 45, 1089, 1092
\ifAMC@shuffleG 35, 748, 751
\ifAMC@survey . . . 2529, 2554
\ifAMC@textPos . 41, 1109, 1112, 1122, 1296, 1305
\ifAMC@watermark 53, 2359, 2687
\ifAMC@zoneformulaire 59, 529, 566, 872
\ifAMCcomplete@multi 46, 845

\ifAMCformulaire@dedans 58, 528, 569, 585, 1944, 2111, 2209, 2266, 2285, 2302
\ifAMCids@side 77, 2380
\ifAMCids@top 76, 2377
\ifAMCneeds@storebox 2525, 2532, 2543
\ifAMCquestionNumber 47, 1262
\ifAMCsz@logged . . . 582, 593
\ifAMCtype@multi . . . 52, 845
\ifAMCune@bonne 51, 847
\ifcase 81
\ifcsname . . . 1193, 1198, 1203, 1238, 1243, 1248
\ifcsvstrcmp 2567, 2573, 2574, 2582
\ifdim 1167
\ifKV@AMCBoxOnly@ordered 2294
\ifKV@AMCdim@cross 385, 400, 417
\ifKV@AMCOpen@annotate 2244
\ifKV@AMCOpen@dots 2202
\ifKV@AMCOpen@lineup 2213, 2256
\ifKV@AMCOpen@retick . . 2243
\ifKV@AMCOpen@scan 2242
\ifKV@AMCtf@headers . . . 1062
\ifKV@AMCtf@inside 1068
\ifSurveySingleAnswer 1311, 1314, 1320
indivanswers (option) . *6*, *10*
init (option) *10*
\InputIfFileExists 17
\insertgroup . . *14*, 747, 2731
\insertgroupfrom . . . *14*, 747
insidebox (option) *10*
\int 883, 885, 889, 895, 902, 941, 1365, 1378, 1417, 1419, 1428, 1434, 1435, 1458, 1522, 1538, 1566, 1569, 1571, 1572, 1573, 1575, 1588, 1591, 1593, 1599, 1600, 1602, 1603, 1605, 1699, 1719, 1722, 1724, 1725, 1726, 1728, 1737, 1762, 1773, 1780, 1786, 1788, 1790, 1796, 1802, 1804, 1809, 1815, 1819, 1823, 1826, 1833, 1835, 1841, 1844, 1847, 1859, 1884, 1885, 1888, 1891, 1895, 1898, 1904, 1908, 1909, 1914, 1915, 1936, 1937, 1938, 1941, 1948, 1949, 1956, 1957, 1963, 1966, 1967, 1973, 1978, 1980, 1984, 1997, 1998, 2003, 2011, 2012, 2018, 2020, 2023, 2031, 2038, 2039, 2043, 2044, 2046, 2059, 2066, 2067, 2073, 2087, 2100

K

\keys 1397, 1424, 1617, 1694, 1947, 2120
keys (option) *10*

L

\l 1956, 1960
\lastchoices 830, 2732
\lastxpos 2640
\lastypos 2643
\linebreak 2224

M

\m@rque 2338, 2371, 2372, 2373, 2374
\m@rqueCalage . . . 2337, 2338
\marginpar 1178, 1220
\mauvaise *21*, 2723
\melangegroupe *21*, 2730
\MessageBreak 2616
\multicolsep 1064
\multiSymbole 1264, 1307, 1326

N

\namefield . . . *15*, 350, 2735
\namefielddots 351
\newbox 366, 1025
\newline 1297
\newsavebox 362
\newstorebox 2544
\next@Answer 1131, 1139, 1145
\nobreak 523, 2257
\node 1114, 1115, 1322
nopage (option) *6*, *11*
noshuffle (option) *10*
noshufflegroups (option) . *10*
\nouveaugroupe 666, 791
nowatermark (option) . . *4*, *10*
\NoWatermarkExterne . . . 2654

O

\oddsidemargin . . 1161, 1165
\onecopy 2412, 2556, 2630, 2727
options:
- answers *10*
- asbox *10*
- automarks *11*
- bloc *21*
- box *10*
- calibration *5*, *10*
- catalog *10*
- chiffres *21*
- completemulti *10*
- correc *21*
- correcindiv *21*
- digits *10*
- ensemble *21*
- francais *10*
- fullgroups *11*
- indivanswers *6*, *10*
- init *10*
- insidebox *10*
- keys *10*
- nopage *6*, *11*
- noshuffle *10*
- noshufflegroups *10*
- nowatermark *4*, *10*
- ordre *21*
- outsidebox *10*
- pdfform *11*
- plain *11*
- postcorrect *11*
- separateanswersheet *5*, *10*, *16*
- storebox *11*

\or 83, 85
ordre (option) *21*
outsidebox (option) *10*
\ouverte@vs 1269

P

\PackageError 651, 693, 764, 2615
\pageref 2465
pdfform (option) *11*
\pdfpageheight 332, 2649, 2650
\pdfpagewidth 332, 2646, 2647
\pgfextractx 1158

\pgfpointanchor 1159
\pgfpointdiff 1158
\pgfpointorigin 1158
plain (option) *11*
postcorrect (option) *11*
\prg 870, 873, 875, 878, 1474, 1570, 1583

Q

\question 1269, 1276
question (env.) *11*, 1269
question-auto (env.) . . . 2554
\QuestionIndicative . *13*, 1314, 1320, 1347, 1382
questionmult (env.) . *11*, 1269
Questionnaires (env.) . . 2554
\questionnum 1299
questionouverte (env.) . 1269

R

\raggedcolumns 1066
\raisebox 1621
\refstepcounter 1262
\reponses 2719
reponses (env.) *21*
\reponseshoriz 2720
reponseshoriz (env.) *21*
\reponsesperso 2721
reponsesperso (env.) *21*
\restituegroupe . . . *21*, 2731
\retick 335
\rngstate 2428

S

\savebox 363
\savepos 2637
\sbox 1150
\scoring *13*, 1314, 1320, 1347, 2724
\scoringDefaultM *13*, 1347, 2725
\scoringDefaultS *13*, 1347, 2726
\ScoringExterne 2654
\section 1073, 1077
separateanswersheet (option) *5*, *10*, *16*
\seq 1364, 1438, 1446, 1512, 1520, 1521, 1522
\setdefaultgroupmode . . 689
\setgroupmode 675, 689
\shuffle@it 823, 837
\shufflegroup *14*, 713, 725, 747, 2730
\shufflegroupslice 730, 733, 750
\smash 2201, 2217
\space 2616
\start@Answers . . 1083, 1278
\storebox 2545
storebox (option) *11*
\str 938
\strut 1168, 1170
\SujetExterne 2654
\SurveySingleAnswerfalse 1311

T

\tex 1515
\textasciicircum 1623
\TextField 355
\textit 102, 115, 142, 156, 195, 221
\textsc 2365
\textsf 507
\thecsvrow 1114, 1115, 2595, 2604
tikz-multi (env.) 1082
tikz-single (env.) 1082
\tikzset 2593
\times 1623
\tl 903, 904, 909, 914, 943, 945, 1509, 1511, 1518, 1519, 1720, 1723, 1729, 1732, 1883, 1897, 1899, 1902, 1903, 1907, 1912, 1913, 1915, 1918, 1919, 1927, 1939, 1940
\tracepos 299, 308, 317

U

\une@rep 1088, 1089, 1094, 1098, 1102, 1105, 1142, 1147, 2589
\useasboundingbox 420
\usebox 364
\usestorebox 2546
\usetikzlibrary 2592

V

\value 1006
values (env.) 2554
values-auto (env.) 2554
variable-auto (env.) . . . 2554
variable-multi (env.) . . 1269
variable-single (env.) . 1269
\version 2704
\vfuzz 372
\vrule 1869
\vtop 1441

W

\wd 372, 373, 1049, 1150
\with . . . 1413, 2476, 2477, 2705, 2706, 2707, 2708, 2709, 2710, 2711, 2712, 2714, 2716
\wrongchoice *13*, 847, 848, 1131, 2151, 2723

X

\x 2580, 2582
\xyclose 347
\xyopen 345