The automultiplechoice package*

Alexis Bienvenüe
paamc@passoire.fr

March 21, 2018

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}
```

*This document corresponds to version revision: r:4007644 from AMC 1.3.0+git2018-03-21

```
      \wrongchoice{Addis Abeba}
      \wrongchoice{Ankara}
      \wrongchoice{Apia}
    \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 ??.

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 **??**. 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 **??**.

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 **??** for an example). Commands `\AMCformBegin` and `\AMCform` must be used to describe the separate answer sheet (see section **??**).

`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` calcels the "DRAFT" indications above pages.

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

`francais` asks for french localisation.

`lang=XX` asks for localisation in `XX` language. At present, only `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 **??**).

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

```
\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}
```

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

`\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.

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

Question 1 ♣ Which contries are in the Americas?

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

`choices` `choiceshoriz` `choicescustom`

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

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

```
\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}
```

Question 1 ♣ 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.

- environment `choiceshoriz` is chosen for short answers:

```
\begin{question}{insect}
  From those animals, which
  is an insect?
  \begin{choiceshoriz}
    \correctchoice{Ant}
    \wrongchoice{Horse}
    \wrongchoice{Turtle}
  \end{choiceshoriz}
\end{question}
```

Question 1 From those animals, which is an insect?

☐ Horse ☐ Ant ☐ Turtle

- environment `choicescustom` is provided to customize answers formatting. See **??** 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. 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, consider the following code:

```
\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:\insertgroup[2]{serie}.
```

which produces:

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

\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 grou ⟨*to*⟩ – if optional parameter ⟨*n*⟩ is given, only the ⟨*n*⟩ first 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, consider the following code:

```
\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}.
```

which produces:

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.

You can find an example involving questions in section **??**.

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:

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

Name and surname:

. . . . . .

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.

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

12

☐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

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

\AMCcodeGridInt[h]{student}{3}

☐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

3.6 Separate answer sheet

\AMCformBegin
\AMCform
\AMCcleardoublepage

To produce separate answer sheets as seen in section **??**,

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 **??** 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):

```
\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}
```

Question 1 How much are 2 plus 8?

☐ 9 ■ 10 ☐ 16 ☐ -6

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.

```
\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{reponses}[o]
      \AMCIntervals{\VQr}{0}{1}{0.1}
    \end{reponses}
  \end{multicols}
\end{question}
```

Question 1
Let X and Y be two independent random variables, following exponential laws with respective parameters 5 and 8. In which interval lies the probability $\mathrm{P}[X < Y]$?

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

\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:

```
\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}
```

Question 2
Compute $\sqrt{11}$ and round it up to 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

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`.

`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).

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 **??** for details.

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.

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

`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).

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

Question 1 $2+2=$

Ⓐ 1 Ⓑ 4 Ⓒ 10

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...

```
\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}
```

Question 1 2+2= (☐ 4 ☐ 3 ☐ 2)

4 Implementation

This package uses the following other packages:

```
\RequirePackage{xcolor} % \fcolorbox to fill (or not) a box
\RequirePackage{fancyhdr} % \pagestyle{empty}
\RequirePackage{bophook} % \AtBeginPage
\RequirePackage{xkeyval} % \setkeys
\RequirePackage{rotating} % \rotatebox
\RequirePackage{fancybox} % \boxput
\RequirePackage{expl3}
```

`\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{AUTOQCM[#1]^^J}}
```

`\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 **??**).

`\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@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).

`\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@correc\AMC@correcfalse
\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`).

`\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\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 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}
}
```

4.4.3 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 :}
}
```

4.4.4 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}
}
```

4.4.5 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}
}
```

4.4.6 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}
}
```

4.4.7 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-choice
  \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}
}
```

4.4.8 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}
}
```

4.4.9 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{◇}
}
```

4.4.10 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 $\{0,\dots,n-1\}$ 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{\AMC@amclog{AUTOQCM[NUM=#1=#2]^^J}%
  \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{`$\langle char \rangle$`}{`$\langle key \rangle$`}` logs the character written in the box referenced as $\langle key \rangle$ in the `.cs` 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@CSFILE{}{%
    \string\answer%
    {\the\AMCid@etud/\thepage:#2}%
    {#1}}%
}
```

4.8.2 Position logging

\AMC@tracebox
\AMC@pagepos

Command `\AMC@tracebox{`$\langle trace \rangle$`}{`$\langle key \rangle$`}{`$\langle content \rangle$`}` makes a LaTeX box around $\langle content \rangle$, and, if $\langle trace \rangle$ 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 $\langle key \rangle$.

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}}\fi}
```

`\AMCdontScan` `\AMCdontAnnotate`

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

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

`\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`

The following lines defines an environment to use a particular file for positions outputs. This is used mainly for documentation or testing.

```
\newwrite\AMC@XYspecial
\newwrite\AMC@tmpXY
\newenvironment{amcxyfile}[1]{%
  \openout\AMC@XYspecial#1%
```

```
  \let\AMC@tmpXY=\AMC@XYFILE%
  \let\AMC@XYFILE=\AMC@XYspecial%
}{\let\AMC@XYFILE=\AMC@tmpXY\closeout\AMC@XYspecial}
```

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

```
\newcommand{\namefield}[1]{\AMC@tracebox{1}{nom}{#1}}
```

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 **??**.

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 **??**.
- `\AMC@answerBox@{K}{\AMC@checkedbox}{}{}` produces ■
- `\AMC@answerBox@{}{8}{}{}` produces 8
- `\AMC@answerBox@{K}{8}{1}{testb}` produces Ⓚ 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@radius]
      (-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{%
  \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}%
```

```
}
\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` 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.

`\AMCchoiceLabel` 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.

`\AMCchoiceLabelFormat` 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\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%
  \csname AMC@shapeprepare@\AMC@shapename@ \endcsname%
}
```

```
\AMCboxStyle{shape=square,size=2.5ex,down=.4ex,rule=.5pt,outsidesep=.1em,color=black,cross,crosschar,crossru
\newcommand\AMCboxColor[1]{\AMCboxStyle{color=#1}}
\let\AMCboxDimensions=\AMCboxStyle
```

\AMCboxOutsideLetter
\AMC@box
\AMC@formBox@
\AMC@formBox
outsideLabelFormat

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%
    \AMC@amclog{AUTOQCM[VAR:scorezones=1]^^J}%
    \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@binaryBoxes

Command `\AMC@binaryBoxes[`$\langle ndigits \rangle$`]{`$\langle n \rangle$`}` prints $\langle ndigits \rangle$ boxes to represent number $\langle n \rangle$ in its binary form. `\AMCbin@one` and `\AMCbin@zero` print individual digit-boxes.

For example, `\AMC@binaryBoxes[12]{367}` shows $367 = 000101101111_2$ using 12 boxes:

```
\newtoks\AMCbin@sequence
\newcount\AMCbin@number
\newcount\AMCbin@ndigits
\newcount\AMCbin@id
\newcount\AMCbin@digit
\def\AMCbin@one{\advance\AMCbin@digit\@ne%
  \AMC@answerBox@{}{\AMC@checkedbox}{1}{chiffre:\the\AMCbin@id,\the\AMCbin@digit}}
\def\AMCbin@zero{\advance\AMCbin@digit\@ne%
  \AMC@answerBox@{}{}{1}{chiffre:\the\AMCbin@id,\the\AMCbin@digit}}
\def\AMCbin@begin#1{\AMCbin@id=#1\AMCbin@digit=\z@}
\newcommand{\AMC@binaryBoxes}[2][1]{%
{\AMCboxDimensions{shape=square,size=.32cm,down=0pt,rule=.2pt,cross=false}\AMCbin@sequence={}\AMCbin@number=
\AMCbin@ndigits=\z@%
\loop%
\ifnum\AMCbin@number>\z@%
\advance\AMCbin@ndigits\@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@ndigits<#1\advance\AMCbin@ndigits\@ne%
\AMCbin@sequence=\expandafter{\expandafter\AMCbin@zero\the\AMCbin@sequence}\repeat%
\the\AMCbin@sequence%
\ifnum\AMCbin@ndigits>#1\PackageError{automultiplechoice}{Too low AMC@NCB value (got #1 but needs \the\AMCbi
}}
```

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` 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.

`\element`

```
\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}%
  \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=\z@\fi%
  \ifnum\AMCtok@size<\@ne%
    \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][0]{%
  \AMCgrouploop@prep{#1}{#2}{#3}%
  {\loop%
    \AMCgrouploop@next{#2}%
    {\the\csname #2@\romannumeral\AMCtok@ik\endcsname}%
  \ifnum\AMCtok@size>\z@\repeat}%
}
\newcommand{\insertgroup}[2][0]{%
  \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 **??** 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][0]{%
  \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}%
}
\newcommand{\copygroup}[3][0]{%
  \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
\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]{%
  \advance\AMCload@counter\@ne\relax%
  \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}}
```

\AMC@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%
  \AMC@amclog{AUTOQCM[BR=0]^^J}\AMC@mem@show%
  \AMC@keepmemorytrue%
  \fi}
```

\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
choiceshoriz
choicescustom
\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}
\newenvironment{choices}[1][r]{%
  \AMCrep@count=\z@\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]{%
  \AMCrep@count=\z@\def\une@rep{\AMCrep@ligne}\AMCrep@init{#1}%
  \par\begin{center}}%
  {\AMC@fin@rep\end{center}}
\newenvironment{choicescustom}[1][r]{%
  \AMCrep@count=\z@\def\une@rep{\AMCrep@perso}\AMCrep@init{#1}%
```

```
  \AMCbeginAnswer\ignorespaces}%
  {\AMC@fin@rep\AMCendAnswer}
```

\AMCrep@bloc
\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\noindent\begin{minipage}{\linewidth}%
    \begin{itemize}\item[#1] #2\end{itemize}\end{minipage}%
  \vspace{\AMCinterBrep}}
\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}%
  \mbox{#1\hspace*{\AMChorizBoxSep}#2}\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{}
```

\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{\correctchoice}[2][]{\global\advance\AMCrep@count\@ne\relax%
  \ifAMC@calibration\AMC@amclog{AUTOQCM[REP=\the\AMCrep@count:B]^^J}\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][]{\global\advance\AMCrep@count\@ne\relax%
  \ifAMC@calibration\AMC@amclog{AUTOQCM[REP=\the\AMCrep@count:M]^^J}\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,align=le
    \else%
      \draw (current page.west |- here)+(0cm,0pt) node[anchor=text,align=right,text width=\AMC@mn@leftmargin
    \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 margin
\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 margin
\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}\@skiphype
\newcommand\AMC@qaff{\arabic{AMCquestionaff}}
```

`\AMCbeforeQuestion` `\AMCbeginQuestion` `\multiSymbole`

The command `\AMCbeforeQuestion` opens a new question. The command `\AMCbeginQuestion{`$\langle n \rangle$`}{`$\langle sign \rangle$`}` will format the question header, where $\langle n \rangle$ is the question number and $\langle sign \rangle$ 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` `questionmult` `questionouverte` `\ouverte@vs`

Environment `{question}{`$\langle key \rangle$`}` encloses a simple question (with one and only one correct choice) with associated unique key $\langle key \rangle$ and the proposed answers.

Environment `{questionmult}{`$\langle key \rangle$`}` 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).

```
\ifx\question\undefined\else\let\question\undefined\fi
\def\AMCnobloc{\AMC@qblocfalse}
\def\AMCbloc{\AMC@qbloctrue}
\newenvironment{question}[2][]{%
  \def\AMCcurrentenv{question}%
  \AMC@stepQuestion%
  \global\def\AMCid@name{#2}\AMC@affecte{#2}{\AMCid@quest}%
  \ifAMC@calibration\AMCmessage{Q=\the\AMCid@quest}\fi%
  \AMCbeforeQuestion%
  \ifx\@empty\AMC@sz@callout\@empty\else%
    \csname\AMC@sz@callout\endcsname%
  \fi%
  \AMCtype@multifalse\ifAMC@qbloc\noindent\begin{minipage}{\linewidth}\fi%
  \ifAMC@affichekeys\index{\texttt{#2}}\fi%
  \AMCbeginQuestion{\ifAMC@affichekeys\ifAMC@ensemble\AMC@qaff\ \fi[\texttt{#2}]\else\AMC@qaff\fi}{#1}%
  \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}\vspace{\AMCinterBquest}\else\vspace{\AMCinterIquest}\fi\AMCmessage{FQ}\AMC@mem@a
\newenvironment{questionmult}[1]{%
  \AMCune@bonnefalse\begin{question}[{\multiSymbole}]{#1}%
  \AMCtype@multitrue\ifAMC@calibration%
  \AMC@amclog{AUTOQCM[MULT]^^J}\fi}%
{\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` is used inside `question`-like environments to give the explanation for the answers of a question.

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

```
  \AMCif@env{question}{}{\AMC@error@explain}%
\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\AMC@amclog{AUTOQCM[B=#1]^^J}\fi}
\def\scoringDefaultS#1{\ifAMC@calibration\AMC@amclog{AUTOQCM[BDS=#1]^^J}\fi}
\def\scoringDefaultM#1{\ifAMC@calibration\AMC@amclog{AUTOQCM[BDM=#1]^^J}\fi}
\def\QuestionIndicative{\ifAMC@calibration\AMC@amclog{AUTOQCM[INDIC]^^J}\fi}
```

4.13 Numerical data

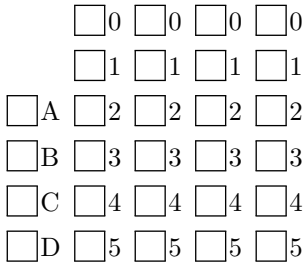
4.13.1 Codes

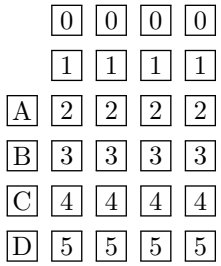
\AMCcodeGrid
\AMCcodeGridInt

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)*⟩`]`.

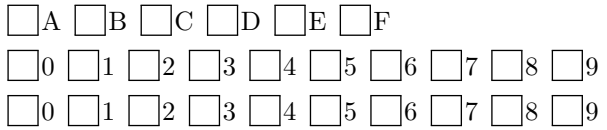
As an example,
`\AMCcodeGrid{code}{ABCD,012345,012345,012345,012345}`
produces the opposite boxes (two results are show here: without or with `separateanswersheet` option), and trace positions of all the boxes in the `.xy` file with the `code` identifier: the first digit is represented by question with 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 **??** for the first set of boxes (without `separateanswersheet`, with digits outside boxes).


0 0 0 0
1 1 1 1
A 2 2 2 2
B 3 3 3 3
C 4 4 4 4
D 5 5 5 5



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.

```
\newcount\AMC@chiffres
\newdimen\AMCcodeHspace\AMCcodeHspace=.5em
\newdimen\AMCcodeVspace\AMCcodeVspace=.5em
\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

\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\AMC@amclog{AUTOQCM[ REP = \the\AMCrep@count : M ]^^J}\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 }
}

\cs_new:Npn \amc_code_generate:nNn #1#2#3 {
  { \keys_set:nn { amccode } { #3 }
    \amc_code_init:N #2
    \clist_map_inline:Nn #2 { % iterates over 'digits'
      \begin{question}{#1[ \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 3 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}%
    \AMC@amclog{AUTOQCM[QPART]^^J}%
  \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) * 10^(#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}
  }
}

\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 },
  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 },
  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 }
}

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

\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
  \AMC@amclog{AUTOQCM[REP= \the\AMCrep@count :
    \ifx#2\AMC@checkedbox B\else M\fi ]^^J}
  \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.

```
\cs_new:Npn \amc_num_digit_box:nn #1 #2 {
  \int_compare:nNnTF { #1 } = { #2 } {
    \amc_num_char:nn{ #1 }{\AMC@checkedbox}
  } {
```

```
    \amc_num_char:nn{ #1 }{}
  }
}
```

The command `\amc_num_sign_boxes:Nn{`⟨*negative*⟩`}{`⟨*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:N #1 #2 {
  \bool_if:nTF { #1 } {
    \hbox{\amc_num_char:nn{$+$}{}}
    \vspace{\AMCnumeric@Vspace}
    \AMC@amclog{AUTOQCM[B=set. sign #2 =1]^^J}
    \hbox{\amc_num_char:nn{$-$}{\AMC@checkedbox}}
    \AMC@amclog{AUTOQCM[B=set. sign #2 =-1]^^J}
  } {
    \hbox{\amc_num_char:nn{$+$}{\AMC@checkedbox}}
    \vspace{\AMCnumeric@Vspace}
    \AMC@amclog{AUTOQCM[B=set. sign #2 =1]^^J}
    \hbox{\amc_num_char:nn{$-$}{}}
    \AMC@amclog{AUTOQCM[B=set. sign #2 =-1]^^J}
  }
}
```

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 }
    \AMC@amclog{AUTOQCM[B= set. #1 = ##1 ]^^J}
    \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 }
      }}
    \AMC@amclog{AUTOQCM[B= set. #1 = ##1 ]^^J}
    \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 }
      }}
    \AMC@amclog{AUTOQCM[B= set. #1 = ##1 ]^^J}
    \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{\AMCntextVHead{ \int_eval:n
                { \amc_num_digit_int - 1 } }}}
          \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*⟩`}{`⟨*positive*⟩`}` draws boxes for integer entry, including the sign if ⟨*sign bool*⟩ is true.

```
\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
        \amc_num_sign_boxes:N { #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:Nnnn{`⟨*tl var*⟩`}{`⟨*sign bool*⟩`}{`⟨*prefix*⟩`}{`⟨*n*⟩`}` builds a scoring to compute an integer from a serie of ⟨*n*⟩-digits boxes, with name prefix ⟨*prefix*⟩, using a sign variable if ⟨*sign bool*⟩ is true.

```
\cs_new:Npn \amc_num_build_integer_scoring:Nnnn #1 #2 #3 #4 {
  \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 {
      \AMC@amclog{AUTOQCM[B=requires. #3
        \int_to_Alph:n \amc_num_digit_int = 1]^^J}
    } {
      \AMC@amclog{AUTOQCM[B=default. #3
        \int_to_Alph:n \amc_num_digit_int = 0]^^J}
    }
    \int_compare:nNnTF \amc_num_digit_int = #4 { } {
      \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
  }
  \tl_put_left:Nn #1 { ( }
  \tl_put_right:Nn #1 { ) }
  \bool_if:NTF { #2 } {
    \bool_if:NTF \amc_num_strict_bool {
      \AMC@amclog{AUTOQCM[B=requires. sign #3 =1]^^J}
    } {
      \AMC@amclog{AUTOQCM[B=default. sign #3 =1]^^J}
    }
    \tl_put_right:Nx #1 { * ( sign #3 ) }
  } { }
}

```

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

```

\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
\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
```

```
    \AMC@amclog{AUTOQCM[Q=\the\AMCid@quest]^^J}
  \fi\fi
```

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

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

```
    \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 * 10^( \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
      }
    }
```

Now extracts the required digits:

```
    \amc_fp_to_digits:Nnnn \amc_num_digits_clist \amc_num_correct_fp
      \amc_num_ndigits_int \amc_num_base_int
    \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
    } {}
```

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.

```
    \bool_if:NTF \amc_num_scoring_bool {
      \AMC@amclog{AUTOQCM[B=haut=,mz=,
        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)
      }
      )]^^J}
  } {}
  \amc_num_build_integer_scoring:Nnnn
    \amc_num_compute_tl \amc_num_sign_bool { digit } \amc_num_ndigits_int
  \int_compare:nNnTF \amc_num_expo_int > 0 {
    \amc_num_build_integer_scoring:Nnnn
      \amc_num_expo_tl \amc_num_exposign_bool { expo } \amc_num_expo_int
    \AMC@amclog{AUTOQCM[B= set. intE = \amc_num_expo_tl ]^^J}
    \tl_put_right:Nx \amc_num_compute_tl
      { * \int_use:N\amc_num_base_int **( intE - (\int_use:N\amc_num_correct_expo_int) ) }
  } {}
  \AMC@amclog{AUTOQCM[B= set.intV = \fp_to_int:N\amc_num_correct_fp ,
    set.intX = \amc_num_compute_tl ]^^J}
  \bool_if:NTF \amc_num_significant_bool {
    \AMC@amclog{AUTOQCM[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)) )"]^^J}
  } {
    \AMC@amclog{AUTOQCM[B=set.Vdifference=abs((intV)-(intX))]^^J}
  }
```

Begin now with the frame around all the boxes:

```
\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
        { \fp_compare_p:nNn \amc_num_correct_fp < 0}}
        \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
          { \int_compare_p:nNn \amc_num_correct_expo_int < 0 }}
        } {}
      }}
    } {
      \amc_num_integer_boxes:NnnNn
      \amc_num_digits_clist { digit } \amc_num_decd_int \amc_num_sign_bool
      { \fp_compare_p:nNn \amc_num_correct_fp < 0}
      \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
        { \int_compare_p:nNn \amc_num_correct_expo_int < 0 }
```

```
      } {}
    }
  }
}
```

And tell AMC that we finished with this question:

```
  \ifAMC@ensemble\else\vspace{1.5ex}\par\fi
  \ifAMC@ensemble\ifAMCformulaire@dedans
    \AMC@amclog{AUTOQCM[FQ]^^J}
  \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 4 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
```

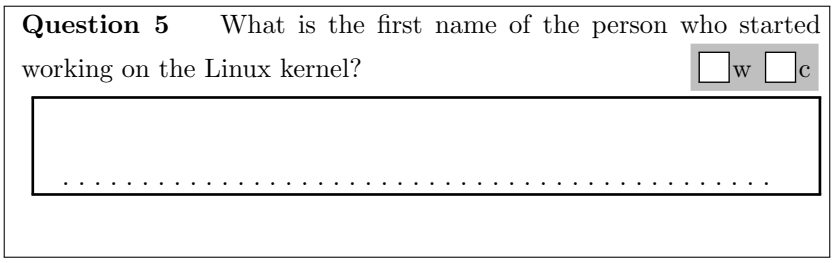
4.14 Open questions

`\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}{annotate}[false]{}
\define@boolkey{AMCOpen}{lineup}[false]{}
\setkeys{AMCOpen}{dots,scan,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%
    \AMC@amclog{AUTOQCM[Q=\the\AMCid@quest]^^J}%
  \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@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%
  \AMC@amclog{AUTOQCM[FQ]^^J}%
  \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%
    \AMC@amclog{AUTOQCM[Q=\the\AMCid@quest]^^J}%
  \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%
  \AMC@amclog{AUTOQCM[FQ]^^J}%
  \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{\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{\AMCbin@begin{1}\AMC@binaryBoxes[\AMC@NCBetud]{\the\AMCid@etud}}
\newcommand\AMCIDBoxesB{\AMCbin@begin{2}\AMC@binaryBoxes[\AMC@NCBpage]{\thepage}}
\newcommand\AMCIDBoxesC{\AMCbin@begin{3}\AMC@binaryBoxes[\AMC@NCBcheck]{\the\AMCid@check}}
\newcommand\AMCIDBoxesABC{%
  \hbox{\vbox{\noindent\AMCIDBoxesA\\
    \noindent\AMCIDBoxesB\AMCIDBoxesC}}%
}
\AtBeginPage{\ifAMC@pagelayout\global\advance\AMCid@check\m@ne%
  \ifnum\AMCid@check<1\global\AMCid@check=\AMCid@checkmax\fi%
  \AMC@pagepos%
  \ifAMC@watermark\ifAMC@correchead\else\AMCw@terprint{\AMCw@termark}%
  \fi\fi\fi}
\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%
  \AMC@amclog{AUTOQCM[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\fi
  \ifAMC@pdfform\begin{Form}\fi%
  \loop{%
    \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\AMC@amclog{AUTOQCM[ETU=\the\AMCid@etud]^^J}\fi%
    \AMC@keepmemoryfalse%
      #2%
    \ifAMC@keepmemory\else\AMC@mem@clear\fi%
    \clearpage}%
  \advance\AMCid@etud\@ne\ifnum\AMCid@etud<\AMCid@etudfin\repeat%
  \ifAMC@pdfform\end{Form}\fi%
  \global\AMCid@etudstart=\AMCid@etud%
}
```

\AMCaddpagesto In some situations, one needs all question sheets to have the same number of pages. The command `\AMCaddpagesto{`⟨*n*⟩`}` adds enough (white) pages to get at least ⟨*n*⟩ 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{⟨*id*⟩} command.

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

4.19 Package options

See section **??** 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{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}
\DeclareOptionX{catalog}{\AMC@catalogMode}
\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}
\ProcessOptionsX

\ifAMCneeds@storebox
  \RequirePackage{storebox}\AtBeginDocument{{}}%
\fi
\ifAMC@pdfform
  \AMC@amclog{AUTOQCM[VAR:project:pdfform=1]^^J}%
  \AMCboxStyle{shape=form}%
  \RequirePackage[pageanchor=false]{hyperref}%
\else%
   \AMC@amclog{AUTOQCM[VAR:project:pdfform=0]^^J}%
\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}
}
```

4.20 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}{Somethin
}}
```

4.21 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 Then, if environ package is loaded and defines command \NewEnviron, environment examcopy is defined.

Environment {examcopy}[⟨n⟩] does the same as command onecopy: it encloses LaTeX code which makes *one* exam copy. Optional argument ⟨n⟩ 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.}}
```

4.22 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 } { }
}

\ExplSyntaxOff
```

4.23 External control

\SujetExterne
\ScoringExterne
\CorrigeExterne
\rrigeIndivExterne
\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@calibrationfalse\AMC@correcheadfalse\AMC@correctrue\AMC@watermarkfalse
\fi
\ifx\CatalogExterne\undefined\else
\message{***CATALOG***^^J}
\AMC@catalogMode
\fi
\ifx\NoWatermarkExterne\undefined\else
\AMC@watermarkfalse
\fi
```

4.24 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
```

4.25 Initialisation

Initialisation of the check counter:

```
\AMCid@check=\AMCid@checkmax\advance\AMCid@check\@ne
```

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

```
\ifAMC@ensemble\AMC@amclog{AUTOQCM[VAR:ensemble=1]^^J}\fi
\ifAMC@inside@box\AMC@amclog{AUTOQCM[VAR:insidebox=1]^^J}\fi
\ifAMC@outside@box\AMC@amclog{AUTOQCM[VAR:outsidebox=1]^^J}\fi
\ifAMC@postcorrect\AMC@amclog{AUTOQCM[VAR:postcorrect=1]^^J}\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=squarebrackets}}
\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{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
```

Preparing writing to `.cs` file :

```
\ifAMC@catalog%
\newwrite\AMC@CSFILE%
\immediate\openout\AMC@CSFILE\jobname.cs%
\fi%
```

4.26 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, `1/`$\langle n\rangle$ means student sheet number 1 (there is only one "student sheet" 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 65536×72.27 `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{0/30:nom}{0sp}{19505360sp}{square}
\tracepos{0/30:nom}{6038827sp}{0sp}{square}
\tracepos{0/30:nom}{16026323sp}{0sp}{square}
\tracepos{0/30:nom}{0sp}{16520182sp}{square}
```

5.2 `AMCboxedchar` command

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

```
\tracepos{0/31:test}{22597209sp}{38766033sp}{square}
\tracepos{0/31:test}{23302629sp}{38060613sp}{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{0/53:case:code[5]:16,1}{25579605sp}{27964975sp}{square}
\tracepos{0/53:case:code[5]:16,1}{26285025sp}{27259555sp}{square}
\boxchar{0/53:case:code[5]:16,1}{A}
\tracepos{0/53:case:code[5]:16,2}{25579605sp}{26850863sp}{square}
\tracepos{0/53:case:code[5]:16,2}{26285025sp}{26145443sp}{square}
\boxchar{0/53:case:code[5]:16,2}{B}
\tracepos{0/53:case:code[5]:16,3}{25579605sp}{25736751sp}{square}
\tracepos{0/53:case:code[5]:16,3}{26285025sp}{25031331sp}{square}
\boxchar{0/53:case:code[5]:16,3}{C}
\tracepos{0/53:case:code[5]:16,4}{25579605sp}{24622639sp}{square}
\tracepos{0/53:case:code[5]:16,4}{26285025sp}{23917219sp}{square}
\boxchar{0/53:case:code[5]:16,4}{D}
\tracepos{0/53:case:code[4]:17,1}{27244404sp}{30193199sp}{square}
\tracepos{0/53:case:code[4]:17,1}{27949824sp}{29487779sp}{square}
\boxchar{0/53:case:code[4]:17,1}{0}
\tracepos{0/53:case:code[4]:17,2}{27244404sp}{29079087sp}{square}
\tracepos{0/53:case:code[4]:17,2}{27949824sp}{28373667sp}{square}
\boxchar{0/53:case:code[4]:17,2}{1}
\tracepos{0/53:case:code[4]:17,3}{27244404sp}{27964975sp}{square}
\tracepos{0/53:case:code[4]:17,3}{27949824sp}{27259555sp}{square}
\boxchar{0/53:case:code[4]:17,3}{2}
\tracepos{0/53:case:code[4]:17,4}{27244404sp}{26850863sp}{square}
\tracepos{0/53:case:code[4]:17,4}{27949824sp}{26145443sp}{square}
\boxchar{0/53:case:code[4]:17,4}{3}
\tracepos{0/53:case:code[4]:17,5}{27244404sp}{25736751sp}{square}
\tracepos{0/53:case:code[4]:17,5}{27949824sp}{25031331sp}{square}
\boxchar{0/53:case:code[4]:17,5}{4}
\tracepos{0/53:case:code[4]:17,6}{27244404sp}{24622639sp}{square}
\tracepos{0/53:case:code[4]:17,6}{27949824sp}{23917219sp}{square}
\boxchar{0/53:case:code[4]:17,6}{5}
\tracepos{0/53:case:code[3]:18,1}{28736261sp}{30193199sp}{square}
\tracepos{0/53:case:code[3]:18,1}{29441681sp}{29487779sp}{square}
\boxchar{0/53:case:code[3]:18,1}{0}
\tracepos{0/53:case:code[3]:18,2}{28736261sp}{29079087sp}{square}
\tracepos{0/53:case:code[3]:18,2}{29441681sp}{28373667sp}{square}
\boxchar{0/53:case:code[3]:18,2}{1}
\tracepos{0/53:case:code[3]:18,3}{28736261sp}{27964975sp}{square}
\tracepos{0/53:case:code[3]:18,3}{29441681sp}{27259555sp}{square}
\boxchar{0/53:case:code[3]:18,3}{2}
\tracepos{0/53:case:code[3]:18,4}{28736261sp}{26850863sp}{square}
```

```
\tracepos{0/53:case:code[3]:18,4}{29441681sp}{26145443sp}{square}
\boxchar{0/53:case:code[3]:18,4}{3}
\tracepos{0/53:case:code[3]:18,5}{28736261sp}{25736751sp}{square}
\tracepos{0/53:case:code[3]:18,5}{29441681sp}{25031331sp}{square}
\boxchar{0/53:case:code[3]:18,5}{4}
\tracepos{0/53:case:code[3]:18,6}{28736261sp}{24622639sp}{square}
\tracepos{0/53:case:code[3]:18,6}{29441681sp}{23917219sp}{square}
\boxchar{0/53:case:code[3]:18,6}{5}
\tracepos{0/53:case:code[2]:19,1}{30228118sp}{30193199sp}{square}
\tracepos{0/53:case:code[2]:19,1}{30933538sp}{29487779sp}{square}
\boxchar{0/53:case:code[2]:19,1}{0}
\tracepos{0/53:case:code[2]:19,2}{30228118sp}{29079087sp}{square}
\tracepos{0/53:case:code[2]:19,2}{30933538sp}{28373667sp}{square}
\boxchar{0/53:case:code[2]:19,2}{1}
\tracepos{0/53:case:code[2]:19,3}{30228118sp}{27964975sp}{square}
\tracepos{0/53:case:code[2]:19,3}{30933538sp}{27259555sp}{square}
\boxchar{0/53:case:code[2]:19,3}{2}
\tracepos{0/53:case:code[2]:19,4}{30228118sp}{26850863sp}{square}
\tracepos{0/53:case:code[2]:19,4}{30933538sp}{26145443sp}{square}
\boxchar{0/53:case:code[2]:19,4}{3}
\tracepos{0/53:case:code[2]:19,5}{30228118sp}{25736751sp}{square}
\tracepos{0/53:case:code[2]:19,5}{30933538sp}{25031331sp}{square}
\boxchar{0/53:case:code[2]:19,5}{4}
\tracepos{0/53:case:code[2]:19,6}{30228118sp}{24622639sp}{square}
\tracepos{0/53:case:code[2]:19,6}{30933538sp}{23917219sp}{square}
\boxchar{0/53:case:code[2]:19,6}{5}
\tracepos{0/53:case:code[1]:20,1}{31719975sp}{30193199sp}{square}
\tracepos{0/53:case:code[1]:20,1}{32425395sp}{29487779sp}{square}
\boxchar{0/53:case:code[1]:20,1}{0}
\tracepos{0/53:case:code[1]:20,2}{31719975sp}{29079087sp}{square}
\tracepos{0/53:case:code[1]:20,2}{32425395sp}{28373667sp}{square}
\boxchar{0/53:case:code[1]:20,2}{1}
\tracepos{0/53:case:code[1]:20,3}{31719975sp}{27964975sp}{square}
\tracepos{0/53:case:code[1]:20,3}{32425395sp}{27259555sp}{square}
\boxchar{0/53:case:code[1]:20,3}{2}
\tracepos{0/53:case:code[1]:20,4}{31719975sp}{26850863sp}{square}
\tracepos{0/53:case:code[1]:20,4}{32425395sp}{26145443sp}{square}
\boxchar{0/53:case:code[1]:20,4}{3}
\tracepos{0/53:case:code[1]:20,5}{31719975sp}{25736751sp}{square}
\tracepos{0/53:case:code[1]:20,5}{32425395sp}{25031331sp}{square}
\boxchar{0/53:case:code[1]:20,5}{4}
\tracepos{0/53:case:code[1]:20,6}{31719975sp}{24622639sp}{square}
\tracepos{0/53:case:code[1]:20,6}{32425395sp}{23917219sp}{square}
\boxchar{0/53:case:code[1]:20,6}{5}
```

Contents