

# The exesheet class and package

Antoine Missier  
antoine.missier@ac-toulouse.fr

August 30, 2021, v1.1

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Titles</b>	<b>2</b>
2.1	The <code>\exercice</code> command . . . . .	2
2.2	The <code>\subpart</code> command . . . . .	3
2.3	The <code>\annex</code> command . . . . .	4
2.4	Exercise titles in table of contents (notoc package option) . . . . .	4
2.5	Short exercises: the <code>\exe</code> command . . . . .	4
2.6	The <code>\points</code> command . . . . .	5
<b>3</b>	<b>Enumerations and lists</b>	<b>5</b>
3.1	List settings (nosetlist package option) . . . . .	5
3.2	List of exercises : the <code>exenumerate</code> environment . . . . .	5
3.3	Items aligned by lines . . . . .	6
3.4	Items aligned by columns: <code>colsenum</code> . . . . .	7
<b>4</b>	<b>Questions and solutions</b>	<b>8</b>
4.1	The <code>questions</code> and <code>answers</code> environments . . . . .	8
4.2	More about <code>answers</code> environment . . . . .	9
4.3	The <code>\question</code> command . . . . .	9
<b>5</b>	<b>Marginal notes for marking scheme</b>	<b>9</b>
5.1	The <code>\pts</code> command . . . . .	10
5.2	The <code>\totalexe</code> , <code>\note</code> and <code>\totalpoints</code> commands . . . . .	10
<b>6</b>	<b>Implementation</b>	<b>12</b>
6.1	Class and package basic instructions . . . . .	12
6.2	Internationalization . . . . .	13
6.3	Colors . . . . .	14
6.4	Titles . . . . .	15
6.5	Enumerations and lists . . . . .	16
6.6	Questions and answers . . . . .	19
6.7	Marginal notes . . . . .	21

# 1 Introduction

The `exesheet` package is used for typesetting exercise or exam sheets. In addition, the `exesheet` class loads the `schooldocs` package. This one makes adjustments for margins and title and defines various layout styles with particular header and footer, appropriate for exercise sheets (among others). See the `schooldocs` documentation for more details. The `exesheet` class is based on `article` and has the same options.

Many other packages are dedicated to exercise sheets. Most propose to encapsulate each exercise in an environment while `exesheet` begins each exercise with the `\exercise` command, which works like a subsection (with same features) and is suitable for documents consisting exclusively of exercises. The package provides also alternative formatting, more relevant for short exercises.

Another specificity of the `exesheet` package is specific settings for enumeration lists, different from L<sup>A</sup>T<sub>E</sub>X standard settings, usefull for the numbering of questions or answers inside an exercise.

Other packages provide often more or less elaborate mechanisms to manage the placement of answers. `exesheet` has no such ambitions: for all exercises of the sheet, we can display subject only, answer only or both, but always at the place they are inserted in the source file. On the other hand this choice may be very flexible: we can do a correct version for all exercises together, or a correct version per exercise, per part (subpart of exercise), per question, per sub-question.

Finally the original functionality of this package is the display of a detailed marking scheme, with optional explanations, in the margin of exercise answers.

There are two possible options: `notoc` (see 2.4) and `nosetlist` (see 3.1).

In the current document, a frame is used to highlight examples beside the rest of the documentation.

## 2 Titles

### 2.1 The `\exercise` command

`\exercise` Each exercise begins with the `\exercise[⟨opt⟩]` command. This command typesets **Exercise**, as a document subsection, followed by automatic numbering, unique for the whole document. The optional parameter `⟨opt⟩` is used to put additional text on the same title line, for example to precise a subject or a marking scheme. `\exercise[ (to begin)]` yields:

#### Exercise 1 (to begin)

Try to use this first command now, it's easy.

`\hrulefill` can be put in the optional argument to produce an horizontal rule.

To bring additional text closer to the exercise number, we can use the `\unskip` command which eliminates preceding space, see the following example, obtained with `\exercise[\unskip : calculation\ \hrulefill]`:

## Exercise 2: calculation

Calculate  $1 + 1$ .

`\exercisename` The word *Exercise* has automatic translation in a few languages<sup>1</sup> when the `babel` package is loaded. It can be redefined, with `\renewcommand`, which should be placed after `\begin{document}`, or else we can use macros from the `babel` package (allowing dynamic language change), e.g.

`\addto\captionssweedish{\def\exercisename{"Övning}}` (in the preamble).

`\labelexercise` This command calls `\exercisename` following by the exercise number. It can be redefined. For example, to add a period after the exercise number:

`\renewcommand{\labelexercise}{\exercisename~\theexercise.}`

`\theexercise` To change only the numbering type, redefine the `\theexercise` command, based on the `exercise` counter.

`\labelexercisestyle` This macro (which is empty by default) allows to define a particular style for exercise titles. In the present document, we defined in the preamble:

`\renewcommand{\labelexercisestyle}{\rmfamily\color{black}}`<sup>2</sup>.

`\exercise*` The starred version `\exercise*[\langle opt \rangle]{\langle label \rangle}` allows to choose another `\langle label \rangle` for a particular exercise and removes the numbering. For instance: `\exercise*[(Fermat theorem)]{Problem}` yields:

### Problem (Fermat theorem)

Proove that there are no positive integers  $x, y, z$  such that  $x^n + y^n = z^n$  for any integer  $n$  greater than 2.

## 2.2 The `\subpart` command

`\subpart` An exercise may contain several parts that we obtain with the `\subpart[\langle opt \rangle]` command, typeset like a subsubsection.

### Exercise 3

#### Part A (preliminary)

First of all, prepare your cup of tea.

#### Part B

Now you are ready to make the current exercise.

`\thesubpart` By default, the subpart numbering uses letters : A, B, C, etc. This numbering type can be redefined with the `\thesubpart` command based on the `subpart` counter, for instance `\renewcommand\thesubpart{\arabic{subpart}}`.

`\subpartname` As for `\exercise`, the `\subpart` command uses `\subpartname` (with automatic translation in a few languages if `babel` is loaded), `\labelsubpart` and

`\labelsubpartstyle` `\labelsubpartstyle`, which are editable.

`\subpart*` Like `\exercise*`, the starred version `\subpart*[\opt]{\label}` allows to freely typeset the subpart `{\label}`, for instance `\subpart*{First part}`.

## 2.3 The `\annex` command

`\annex` The `\annex[\opt]` command typesets the title **ANNEX**, in uppercase letters, centred and in the subsection style, with an optional parameter, added on the same line.

### ANNEX (to return)

`\annexname` The word *Annex* has automatic translation in a few languages if `babel` package is loaded. It can be extended to other languages or modified by redefining `\annexname`.

`\annexstyle` The annex title style is set by the `\annexstyle` macro, defined as follows: `\newcommand\annexstyle{\MakeUppercase}`. This command may be redefined as one wants.

## 2.4 Exercise titles in table of contents (notoc package option)

`notoc` By default, the titles *Exercise*, *Part* or *Annex*, appear in the table of contents (or in the pdf file summary when `hyperref` package is used). To avoid this, you can set the boolean `notoc` to `true` or call the `exesheet` package (or class) with `notoc` option. But notice that title optional arguments will always be ignored in the table of contents.

## 2.5 Short exercises: the `\exe` command

`\exe` The `\exe` command starts an exercise by the abbreviation **Ex.** followed by the exercise number, without using sectioning commands, and the exercise body begins on the same line. An exercise starts a new paragraph without indentation.

**Ex. 4** — This is a short exercise who can contain several paragraphs or questions however.

Here for example starts a new paragraph.

**Ex. 5** — This is another short exercise.

`\exname` The abbreviation *Ex* may be changed by redefining `\exname`. The `\exlabel` macro calls `\exname` following by a period then the exercise number, and `\exsepmark` `\exsepmark` typesets a long dash. We may change these features by redefining these commands.

`\exe*` The starred version prints no separator as shown below:

**Ex. 6** Another short exercise without separator.

<sup>1</sup>Translation is currently integrated into the package for the following languages: French, German, Spanish, Italian, Portuguese.

<sup>2</sup>In the present document, to highlight real sections and subsections titles, their color and font have been modified with the `\allsectionsfont` macro from the `sectsty` package.

## 2.6 The `\points` command

`\points` The `\points{<pts>}` command displays the number of points awarded to an exercise. It is intended to be entered in the optional argument of the `\exercise` command<sup>3</sup>. `\exercise[\points{5}]` yields:

### Exercise 7

5 points

Try to read this whole document without drinking tea and you get five points.

`\pointsname` Like before, the words *points* and in the singular *point* (if *<pts>* is inferior to 2) get automatic translation (with `babel`) in a few languages and are modifiables.

`\pointname` The style used by the `\points` command may be redefined with `\pointsstyle`.

`\pointsstyle` The color is set by `pointscolor` with the `\definecolor` command (from the `xcolor` package by Uwe Kern, loaded by `exesheet`), for instance:

`pointscolor` `\definecolor{pointscolor}{named}{blue}`.

## 3 Enumerations and lists

### 3.1 List settings (nosetlist package option)

Enumeration lists are intended to represent questions and sub-questions inside exercises. For a good highlight, labels are typeset in bold. Moreover, they are left aligned, at the start of the line, without indentation, and the vertical space between items is increased compared with L<sup>A</sup>T<sub>E</sub>X standard lists. These settings are done by the `\setlist` command<sup>4</sup> from the `enumitem` package by Javier Bezos. Lists with `itemize` environment are kept in their default configuration<sup>5</sup>.

### Exercise 8

1. First question
  - (a) First sub-question
  - (b) Second sub-question
2. Second question

`nosetlist` We can avoid enumeration list modifications and restore the L<sup>A</sup>T<sub>E</sub>X default settings by passing the option `nosetlist` to the `exesheet` class or package.

### 3.2 List of exercises : the `exenumerate` environment

`exenumerate` When an exercise sheet is made of short independant exercises, it would be ill-advised to display the complete title *Exercise* for each. In addition to the `\exe` command, previously presented, we provide an even lighter solution with the

<sup>3</sup>This command which has to be used in the optional argument of `\exercise`, as several others presented below, is incompatible with the `memoir` class which redefines section commands.

<sup>4</sup>Labels may also be changed by the `\labelenumi` and `\labelenumii` commands.

<sup>5</sup>The `french` option of the `babel` package alters `itemize` lists behavior and uses long dash as labels for each list level. This behavior is problematic when mathematics follows the dash symbol because the latter may be confused with the minus sign. Modifications of the `french` option (which supersedes `frenchb` of `babel`) are canceled to restore L<sup>A</sup>T<sub>E</sub>X default settings.

`exenumerate` environment, which is only an enumeration list in which spaces between items are further increased compared with those of `enumerate`. Here is an example (the main list is an `exenumerate` environment but sublists are produced with regular `enumerate` environments):

1. Translate the following sentences in english:
  - (a) Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi.
  - (b) Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus.
2. Translate the following sentence in german:
 

Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi.
3. Translate the following sentences in french:
  - (a) Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.
  - (b) Nulla ullamcorper vestibulum turpis.

The environment takes an optional parameter, like `enumerate`, which enables, among others, to typeset alternative list labels, e.g. `\begin{exenumerate}[A.]`. There are many other options (see the `enumitem` package documentation).

### 3.3 Items aligned by lines

`tablenum1` These three environments are used to typeset short questions (`tablenum1`), sub-questions (`tablenuma`) or `itemize` lists (`tablitem`) on the same line. They have the same syntax: `\begin{tablenum1}[\langle opt \rangle](\langle cols \rangle)`. The `\langle cols \rangle` parameter is the number of columns used by the environment. It must be *in parentheses*. This parameter can be omitted, then its value is 2. As for classic lists, each item begins with the `\item` command.

These three environments are defined by the `\NewTasksEnvironment` macro, from the `tasks` package by Clemens Niederberger. They take an optional argument `\langle opt \rangle` explained in the documentation of this package, e.g. `label=1` gives a roman numbering following by a closing parenthesis (like in `enumitem` or `enumerate` packages). There are also many possibilities to place items in an original way, for instance, the `\item*` command allows to specify the number of columns the item is supposed to span. In the following example, the five `\item` are placed in order between `\begin{tablenum1}(3)` and `\end{tablenum1}`. Notice that numbering is made line by line.

#### Exercise 9

Give the derivative of the following functions:

1.  $f(x) = \frac{1-x^2}{e^x + e^{-x}},$
1.  $g(x) = \ln\left(\frac{1-x}{1+x^2}\right),$
1.  $h(x) = \int_0^1 e^{xy} dy,$
1.  $k(x) = \sum_{i=1}^{\infty} \frac{1}{x^i},$
1.  $l(x) = \int_{\frac{1}{x}}^x \frac{1}{\ln t} dt.$

For `tablenuma`, labels are letters (a, b, c, ...) surrounded by parentheses. This cannot be modified globally (except by redefining the environment with `\RenewTasksEnvironment`). If the `exesheet` package is called with option `nosetlist`, labels of `tablenum1` and `tablenuma` environments are displayed in normal font with indentation, as for `enumerate`.

### 3.4 Items aligned by columns: `colsenum`

`colsenum` To get numbering of items by columns, we have the `colsenum` environment: `\begin{colsenum}[\langle opt \rangle]{\langle cols \rangle}`. The mandatory parameter is the number of columns and the optional one will be passed to `enumerate`, allowing, for example, to change the type of numbering (a, A, etc.). To use this environment, we have to load the `multicol` package in the preamble.

#### Exercise 10

Give the derivative of the following functions:

1.  $f(x) = \frac{1-x^2}{e^x + e^{-x}},$
3.  $h(x) = \int_0^1 e^{xy} dy,$
5.  $l(x) = \int_{\frac{1}{x}}^x \frac{1}{\ln t} dt.$
2.  $g(x) = \ln \left( \frac{1-x}{1+x^2} \right),$
4.  $k(x) = \sum_{i=1}^{\infty} \frac{1}{x^i},$

`colsenum*` We will notice that, on each line, items are not necessarily well aligned, what can produce inelegant effects. On the other hand, the `colsenum` environment doesn't try to align columns from the bottom by adjusting space between items. If we want to get this (which is the default option in `multicol`) we have the `colsenum*` environment (same syntax than `colsenum`). Here what we get in that case, with the same exercise.

#### Exercise 11

Give the derivative of the following functions:

1.  $f(x) = \frac{1-x^2}{e^x + e^{-x}},$
3.  $h(x) = \int_0^1 e^{xy} dy,$
5.  $l(x) = \int_{\frac{1}{x}}^x \frac{1}{\ln t} dt.$
2.  $g(x) = \ln \left( \frac{1-x}{1+x^2} \right),$
4.  $k(x) = \sum_{i=1}^{\infty} \frac{1}{x^i},$

We can see that these alignments are less good than those obtained by numbering of items by line. The numbering of items by columns may nevertheless be preferable when there are many items with variable heights, and a number of items which can be different from one column to the other. Moreover, an advantage of `colsenum` is that the choice of labels is automatic depending on the list level (and the language), unlike `tablenum1` or `tablenuma`.

`colsitem` For `itemize` lists, the environment `colsitem` produces items aligned by columns rather than by lines as for `tablitem`: `\begin{colsitem}[\langle opt \rangle]{\langle cols \rangle}`.  
`colsitem*` The optional parameter, which is passed to the underlying `itemize` environment,

allows to change the item label (bullet by default). And, as for `colsenum*`, the `colsite*` environment produces an alignment of columns from the bottom.

## 4 Questions and solutions

### 4.1 The questions and answers environments

<code>questions</code>	The package provides the two environments <code>questions</code> and <code>answers</code> to make
<code>answers</code>	appear or disappear questions and answers of exercises. Display is controlled by
	two booleans: <code>questions</code> and <code>answers</code> . Their default value is <code>true</code> .
<code>\questiononly</code>	The <code>\questiononly</code> command allows to display the questions without the
<code>\answeronly</code>	answers and <code>\answeronly</code> displays the solutions without the questions <sup>6</sup> .
<code>\correctionstyle</code>	In the case (by default) where questions and answers are displayed both,
<code>correctioncolor</code>	answers are then typeset in the style <code>\correctionstyle</code> , which uses the color
<code>\correctionname</code>	<code>correctioncolor</code> . This color may be chosen with the <code>\definecolor</code> macro (by default <code>\definecolor{correctioncolor}{rgb}{0,0.2,0.6}</code> = kind of dark blue). Moreover, in that case, the title <b>Correction</b> is displayed at the beginning of the <code>answers</code> environment. It is defined by the <code>\correctionname</code> macro (with translation in a few languages and that we can change: <sup>7</sup> for example we may prefer <i>Solution</i> than <i>Correction</i> ). The style defined by <code>\correctionstyle</code> will apply to the title but also to the whole environment. Here an example.

#### Exercise 12

3 points

1. Is the exesheet package useful ?
2. Isn't there any other packages that deal with exercises ?

#### Correction

1. Yes, the exesheet package is very useful.
2. There are many other packages that deal with exercises, and give the ability to produce separately questions and solutions, for example `exercise` by Paul Pichaureau, `exercises` by Roger Jud, `exsheets` (superseded by `xsim`) by Clemens Niederberger, `exframe` by Niklas Beisert, `exam` by Philip Hirschhorn, `answers` by Mike Piff and Joseph Wright, `probsoln` by Nicola Talbot, etc.

`\points` When only answers are displayed, the text color remains black, the word *Correction* is not displayed and the `\points` macro is patched to suppress the display of points. An extended solution for a variable display of points will be discussed in section 5.2 with the `\totalpoints` macro.

<sup>6</sup>Of course we can also assign the booleans `questions` and `answers` directly with the  $\text{\LaTeX}$  command `\setboolean`.

<sup>7</sup>Except for English, if you want to change package keywords you have to do it *after* `\begin{document}` because `babel` will modify these definitions at the end of the preamble.



## 4.2 More about answers environment

Internally, we have used the `\comment` and `\endcomment` macros from the `versions` package by Uwe Lück. Other excellent packages allow to manage selectively piece of code. Let us mention `verbatim` by Rainer Schöpf, `comment` by Victor Eijkhout, `version` by Donald Arseneau and Stephen Bellantoni, `optional` by Donald Arseneau and `codesection` by Matthias Pospiech.

The `versions` package used provides furthermore the `\includeversion{⟨env⟩}` and `\excludeversion{⟨env⟩}` macros who allow to make appear or disappear any environment `⟨env⟩` and these optional environments may be nested<sup>8</sup>.

However the `questions` and `answers` environments perform another task, not only making appear or disappear piece of text. Indeed, there is a problem with the title *Correction*. In which format to typeset it and at which level to put it in the table of contents (or in the summary of the pdf file)? In fact it depends on which level the environment has been nested. We can make a single `answers` environment for all the exercises of the sheet or an `answers` environment for each exercise, for each exercise part, for each question or sub-question. In fact, the typeset of the title *Correction* and his level in the table of contents will be adjusted automatically.

`[⟨level⟩]` Nevertheless, we can imagine twisted situations in which the title level will not be correctly calculated. It is then possible to force the level of the title *Correction* with `\begin{answers}[⟨level⟩]`. The optional `⟨level⟩` parameter is defined as follows: 1 for section level titles, 2 for subsections (like *Exercise*), 3 for subsubsections (like *Part*), another number for lower levels (then they will not appear in the table of contents).

`answers*` The starred version `answers*` allows to make the title **Correction** completely disappear.

## 4.3 The \question command

`\question` Instead of the `questions` and `answers` environments, we can also use the simple `\question{⟨ques⟩}{⟨ans⟩}` macro in which display of `⟨ques⟩` and `⟨ans⟩` arguments is controlled by the same previous commands `\questionsonly` and `\answeronly`. It can be more readable (in the source code) when questions and answers are short. In the case of displaying both questions and answers, the *Correction* title will appear on a new line, in bold, as for levels lower than subsubsection in the `answers` environment (and without entries in the table of contents). But this command doesn't work with `verbatim` inside.

## 5 Marginal notes for marking scheme

The `exesheet` package allows the display of a marking scheme with comments and explanations for answers.

---

<sup>8</sup>The `codesection` package also allows such nesting, including in the preamble, as well as the optional package, but the latter only manages short optional code.

## 5.1 The `\pts` command

`\pts` When exercises are typeset with the `\exe` macro, or as a list with the `exenumerate` environment, the marking scheme is displayed in the margin, on the line where we put the `\pts{<num>}` command (in general the first line of the exercise). The `<num>` parameter is the number of points assigned to the exercise.

(3 pts)

**Ex. 13** — The first exercise with marking scheme.

(1.5 pt)

**Ex. 14** — The second one.

`\ptsname` The abbreviation `pts` (or `pt` when the number of points is inferior to 2) is automatically added thanks to the macro `\ptsname` or `\ptname` (translated in a few languages if `babel` is loaded). The display color of the points

`\ptname`

`ptscolor` is defined by `ptscolor` which can be changed with `\definecolor`: by default `\definecolor{ptscolor}{named}{red}`. The display style is defined by

`\ptsstyle` `\ptsstyle`.

`\displaypts` The display of the marking scheme, as above, is internally controlled by the `marginpts` boolean. The `\displaypts` command, which has to be placed in the preamble, defines `marginpts` to `true`, places marginal notes to the left (and not to the right which is the default behavior of `LATEX`) and shifts the text body right to increase the left margin compared with the right one, with a ratio 3:2, as it is in the present document. This ratio is defined by the `geometry` macro from the `geometry` package (by Hideo Umeki) and is valid for the whole document.

## 5.2 The `\totalex`, `\note` and `\totalpoints` commands

`\totalex` To give a more detailed marking scheme, the `exesheet` package provides some extended commands: `\totalex{<num>}` to display the total number of points of an exercise and especially `\note[<pts>]{<comment>}`, to indicate some marking detail in answers. The optional parameter `<pts>` is the number of points of the question and the mandatory parameter `<comment>` is used to give details about the marking scheme. Below the title has been obtained with `\exercise[\totalex{4}]` and the first comment with

`\note[1]{0,5 for the anti-derivative\0,5 for simplification of $\ln$}.`

4 pts

### Exercise 15

For each following question, say if the assertion is true or false. Justify the answer carefully.

1.  $\int_0^{\sqrt{3}} \frac{1}{x + \sqrt{3}} dx = \ln 2,$

1.  $\int_2^e \frac{1}{x \ln x} dx = -\ln 2,$

1. The function  $F$  defined on  $\mathbf{R}$  by  $F(x) = \int_0^x \frac{1}{t^2 + t + 1} dt$  is increasing on  $\mathbf{R}$ .

## Correction

1

0,5 for the anti-derivative  
0,5 for simplification of  $\ln$

1. We calculate:

$$\int_0^{\sqrt{3}} \frac{1}{x + \sqrt{3}} dx = \left[ \ln(x + \sqrt{3}) \right]_0^{\sqrt{3}} = \ln(2\sqrt{3}) - \ln \sqrt{3} = \ln \left( \frac{2\sqrt{3}}{\sqrt{3}} \right) = \ln 2.$$

TRUE.

1.5

1 for the anti-derivative  
0.5 for the integral value

2. We have  $\frac{1}{x \ln x} = \frac{\frac{1}{x}}{\ln x} = \frac{u'(x)}{u(x)}$  with  $u(x) = \ln x$ , which is positive on  $[2, e]$ .

Hence

$$\int_2^e \frac{1}{x \ln x} dx = \left[ \ln(\ln x) \right]_2^e = \ln(\ln e) - \ln(\ln 2) = \ln 1 - \ln(\ln 2) = -\ln(\ln 2).$$

FALSE.

we could also see that  $\frac{1}{x \ln x} > 0$   
on  $[2, e]$  while  $-\ln 2 < 0$

1.5

3. The function  $F$  defined on  $\mathbf{R}$  by

$$F(x) = \int_0^x \frac{1}{t^2 + t + 1} dt$$

0.5 for  $F'$   
1 for the sign of  $F'$  and  
conclusion

is derivable on  $\mathbf{R}$  and its derivative is such that  $F'(x) = \frac{1}{x^2 + x + 1}$ . The denominator is a quadratic polynomial, always positive because its discriminant is  $\Delta = -3 < 0$ . Thus  $F$  is increasing on  $\mathbf{R}$ .

TRUE.

In the comment of answer 2, a wider vertical space is produced at line break with the optional argument `\\[2ex]`<sup>9</sup>. The last comment, not placed beside the number of points of answer 3, has been produced by placing on the first line after the formula: `\note{0.5 for $F'$\\1 for the sign of $F'$ and conclusion}`.

`\note*`

If we don't want to add comment, we have the `\note*{<pts>}` command (better than `\note[<pts>]{}`), used here in answer 3: `\note*{1.5}`.

`markingcolor`  
`\markingstyle`  
`\ptsboxlength`

The display color of the points, in `\totalexe` and `\note`, is defined by `markingcolor` and the style by `\markingstyle`, which are modifiables. The oval box produced by `\totalexe` is obtained with the `\ovalbox` command of the `fancybox` package by Timothy Van Zandt, with corner arcs set by `\cornersize{1}`. The box length is not adjusted to content but depends of the value of `\ptsboxlength` in order to keep a uniform appearance from one exercise to the other.

`notecolor`  
`\notestyle`  
`\noteragged`

Comment notes are typeset by default in a dark green color defined by `\definecolor{notecolor}{rgb}{0.0,0.4,0.0}`. The style of the comment is set by the `\notestyle` macro and itself uses the `\noteragged` macro. The latter is set by default as `\raggedleft` to obtain a *right* alignment (what is customary for notes in the left margin).

`\displaypoints`

Display of points and comments is optional. It is internally controlled by the booleans `marginpoints` and `marginfullnotes`. The `\displaypoints` macro, to place in the preamble, sets `marginpoints` to `true`, chooses the left margin for

<sup>9</sup>When a marginal note at the end of a page, is followed by another one at the top of the next page, the latter is shifted down. We can raise this note, by adding at the line break a negative vertical space : `\note*{\\[-5ex] 2}`.

display marginal notes and sets the ratio between left and right margins to 3:2<sup>10</sup>. Points will then be displayed without the comments.

`\displaynotes` The `\displaynotes[⟨align⟩]` macro, to be placed also in the preamble, defines both booleans to `true` and allows then the complete display of points and comments. The ratio between left/right margins is then set to 5/1<sup>11</sup>. The optional parameter `⟨align⟩` is used to define alignment: `\raggedleft` by default, but we could use `\centering` or `\raggedright` (not recommended in left margin).

`\displaynotesright` The `\displaynotesright[⟨align⟩]` macro displays a detailed marking scheme with comments in the right margin, with the same type of settings as `\displaynotes` but an optional parameter `⟨align⟩` set by default to `\raggedleft`.

`\totalpoints` The `\totalpoints{⟨num⟩}` macro is intended to replace `\points` when using a detailed marking scheme. When the scale is not displayed (`marginpoints` boolean to `false`), it uses `\points` and when the scale is displayed, it uses `\totalexe`. For example, in the exercise 15, we should use `\totalpoints` rather than `\totalexe`, because, when the detailed marking scheme is not displayed, the total points will be typeset as in the exercise 12 rather than in the margin.

## 6 Implementation

### 6.1 Class and package basic instructions

The `exesheet` class is based on the `article` class and gives it all its options, except explicit options `notoc` and `nosetlist`.

```
1 \*class
2 \DeclareOption{notoc}{\PassOptionsToPackage{notoc}{exesheet}}
3 \DeclareOption{nosetlist}{\PassOptionsToPackage{nosetlist}{exesheet}}
4 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
5 \ProcessOptions \relax
6 \LoadClass{article}
7 \RequirePackage{exesheet}
8 \RequirePackage{schooldocs}
9 \end{class}
```

Then we define initialization instructions of the `exesheet` package.

The `shortlabel` option in the `enumitem` package allows to use labels like in the `enumerate` package e.g. 1., a), A. etc.

```
10 \*package
11 \RequirePackage{ifthen}
12 \newboolean{notoc}
13 \newboolean{nosetlist}
14 \DeclareOption{notoc}{\setboolean{notoc}{true}}
15 \DeclareOption{nosetlist}{\setboolean{nosetlist}{true}}
16 \ProcessOptions \relax
17 \RequirePackage{xcolor}
18 \RequirePackage[shortlabels]{enumitem}
```

<sup>10</sup>So that the effect on the margin ratio is correct, the `\displaypoints` macro must be placed after other commands that could alter the page geometry, for example the `\pagestyle` of the `schooldocs` package.

<sup>11</sup>The present document kept a ratio of 3/2 because the `ltxdoc` class, used here, leaves large enough margins and previous examples are not concerned with display of comments.

```

19 \RequirePackage{tasks}
20 \RequirePackage{versions}
21 %\@ifpackageloaded{doc}{\RequirePackage{verbatim}}
22 \RequirePackage{geometry}
23 \RequirePackage{fancybox}
24

```

## 6.2 Internationalization

Here we define the keywords and their translation in French, German, Spanish Italian, Portuguese, thanks to the `\addto\captions<language>` macro of the `babel` package.

Translations are placed in a macro `\exetranslate` because, if `exesheet` is called before `babel` (this is necessarily the case for the `exesheet` class), then these translations should only happen after having loaded the `babel` package, therefore the `\AtBeginDocument`. On the other hand, if the package is loaded *after* `babel`, `\AtBeginDocument{\exetranslate}` doesn't work, curiously. But calling immediately `\exetranslate` works in that case.

Accented characters cannot be used here because they are not recognized if `inputenc` is loaded after `exesheet`, despite the `\AtBeginDocument`. So we have used the basic L<sup>A</sup>T<sub>E</sub>X commands to produce them.

**TODO:** give the choice to use `polyglossia` instead of `babel`; see the translations or `multilang` packages.

```

25 \def\exercisename{Exercise}
26 \def\subpartname{Part}
27 \def\annexname{Annex}
28 \def\exname{Ex}
29 \def\pointsname{points}
30 \def\pointname{point}
31 \def\correctionname{Correction}
32 \def\ptsname{pts}
33 \def\ptname{pt}
34
35 \newcommand\exetranslate{%
36   \@ifpackageloaded{babel}{%
37     \addto\captionsfrench{%
38       \def\exercisename{Exercice}
39       \def\subpartname{Partie}
40       \def\annexname{Annexe}
41       \def\exname{Ex}
42       \def\pointsname{points}
43       \def\pointname{point}
44       \def\correctionname{Correction}
45       \def\ptsname{pts}
46       \def\ptname{pt}
47     }
48     \addto\captionsgerman{%
49       \def\exercisename{"\Ubung}
50       \def\subpartname{Teil}
51       \def\annexname{Anhang}
52       \def\exname{"\Ub}
53       \def\pointsname{Punkte}

```

```

54         \def\pointname{Punkt}
55         \def\correctionname{Verbesserung}
56         \def\ptsname{Pkte}
57         \def\ptname{Pkt}
58     }
59     \addto\captionsspanish{%
60         \def\exercisename{Ejercicio}
61         \def\subpartname{Parte}
62         \def\annexname{Anexo}
63         \def\exname{Ej}
64         \def\pointsname{puntos}
65         \def\pointname{punto}
66         \def\correctionname{Correcci'on}
67         \def\ptsname{ptos}
68         \def\ptname{pot}
69     }
70     \addto\captionsitalian{%
71         \def\exercisename{Esercizio}
72         \def\subpartname{Parte}
73         \def\annexname{Ansesso}
74         \def\exname{Es}
75         \def\pointsname{punti}
76         \def\pointname{punto}
77         \def\correctionname{Correzione}
78         \def\ptsname{pti}
79         \def\ptname{pt}
80     }
81     \addto\captionoportuges{%
82         \def\exercisename{Exerc'icio}
83         \def\subpartname{Parte}
84         \def\annexname{Anexo}
85         \def\exname{Ex}
86         \def\pointsname{pontos}
87         \def\pointname{ponto}
88         \def\correctionname{Corre'c c\~ao}
89         \def\ptsname{pts}
90         \def\ptname{pt}
91     }
92     }{}
93 }
94
95 \AtBeginDocument{\exetranslate} % if loaded before babel
96 \exetranslate % necessary when loaded after babel
97

```

## 6.3 Colors

The color `pointscolor` is used by the macro `\points` and `markingcolor` by the macros `\totalexe`, `\note*` and also for the optional argument of `\note`; `notecolor` is used by the mandatory argument of `\note`.

```

98 \definecolor{pointscolor}{named}{red}
99 \definecolor{ptscolor}{named}{red}
100 \definecolor{markingcolor}{named}{red}

```

```

101 \definecolor{notecolor}{rgb}{0.0, 0.4, 0.0} % kind of dark green
102 \definecolor{correctioncolor}{rgb}{0,0.2,0.6} % kind of dark blue
103

```

## 6.4 Titles

The `\exercise` counter numbers exercises for the whole document regardless of any section. To reset the counter at some point, just write `\setcounter{exercise}{0}` and if we want an automatic reset at each section, add in the preamble `\makeatletter \addtoresetautocounter{exercise}{section} \makeatother`.

The parts counter depends on `exercise` and is reset at each new exercise.

The `\labelexercisestyle` and `\labelsubpartstyle` commands are empty, but allow to customize the style, for instance:

```
\renewcommand\labelexercisestyle{\sffamily}.
```

By default, the table of contents displays the titles of exercises and parts. The `\notoc` boolean is used to modify this behavior. If we want to display only exercise titles but not parts, we can place in the preamble `\setcounter{tocdepth}{2}`.

`\exercise`

```

104 \newcounter{exercise}
105
106 \newcommand{\labelexercise}{\exercisename~\theexercise}
107 \newcommand{\labelexercisestyle}{}
108 \newcommand*{\@exercise}[1] [] {%
109     \refstepcounter{exercise}
110     \subsection*{\labelexercisestyle\labelexercise\ #1}
111     \ifthenelse{\boolean{notoc}}{}{
112         \addcontentsline{toc}{subsection}{\labelexercise}
113     }
114 }
115 \newcommand*{\@@exercise}[2] [] {%
116     \subsection*{\labelexercisestyle #2 #1}
117     \setcounter{subpart}{0} % resets the parts counter
118     \ifthenelse{\boolean{notoc}}{}{
119         \addcontentsline{toc}{subsection}{#2}
120     }
121 }
122 \newcommand{\exercise}{\@ifstar{\@@exercise}{\@exercise}}
123

```

`\subpart`

```

124 \newcounter{subpart}[exercise] %
125 \renewcommand{\thesubpart}{\Alph{subpart}}
126
127 \newcommand{\labelsubpart}{\subpartname~\thesubpart}
128 \newcommand{\labelsubpartstyle}{}
129 \newcommand*{\@subpart}[1] [] {%
130     \refstepcounter{subpart}%
131     \subsubsection*{\labelsubpartstyle\labelsubpart\ #1}
132     \ifthenelse{\boolean{notoc}}{}{
133         \addcontentsline{toc}{subsubsection}{\labelsubpart}
134     }

```

```

135     }
136 \newcommand*{\@@subpart}[2][]{%
137     \subsubsection*{\labelsubpartstyle #2 #1}
138     \ifthenelse{\boolean{notoc}}{}{
139         \addcontentsline{toc}{subsubsection}{#2}
140     }
141 }
142 \newcommand{\subpart}{\ifstar{\@@subpart}{\@subpart}}
143

```

`\annex`

```

144 \newcommand{\annexstyle}{\MakeUppercase}
145 \newcommand*{\annex}[1][]{%
146     \subsection*{\mbox{} \hfill \annexstyle{\annexname} #1 \hfill \mbox{}}
147     \ifthenelse{\boolean{notoc}}{}{
148         \addcontentsline{toc}{subsection}{\annexname}
149     }
150 }
151

```

`\exe`

```

152 \newcommand{\exlabel}{\exname.\theexercise}
153 \newcommand{\exsepmark}{---}
154 \newcommand{\@exe}{\textbf{\exlabel~\exsepmark}~}
155 \newcommand{\@@exe}{\textbf{\exlabel}~}
156 \newcommand{\exe}{\bigskip\par\noindent\refstepcounter{exercise}
157     \ifstar{\@@exe}{\@exe}
158 }
159

```

`\points (ini)`

```

160 \newcommand{\pointstyle}{%
161     \small\mdseries\sffamily\color{pointcolor}\fbox}
162 \newcommand*{\points}[1]{\hfill
163     \pointstyle{#1}%
164     \ifthenelse{\lengthtest{#1 cm < 2cm}}{\pointname}{\pointsname}%
165 }
166 }
167

```

Percent symbols are necessary to avoid spaces between the `\fbox` and its inner text. Note that, without `\lengthtest`, the test `#1 < 2` doesn't work with decimal numbers.

## 6.5 Enumerations and lists

When using `babel` with the `french` option<sup>12</sup>, `itemize` lists are modified with the same dash label for each list level. These modifications are cancelled here to restore default L<sup>A</sup>T<sub>E</sub>X `itemize` lists (labels and spaces). As for `\exetranslate`, we have create the `\standardfrenchlists` command who must be called into `AtBeginDocument` or not, depending on whether `exesheet` is loaded before `babel` or after.

---

<sup>12</sup>The `french` option of `babel` supersedes the `frenchb` option.



The `\setlist` command comes from the `enumitem` package (`\setenumerate` is deprecated). By default `itemsep=1ex` for lists of first level, and `leftmargin=1.5em` allows to align labels on the start of lines.

```

168 \newcommand\standardfrenchlists{%
169   \ifpackagewith{babel}{frenchb}{
170     \frenchbsetup{StandardLists=true}}{}
171   \ifpackagewith{babel}{french}{
172     \ifundefined{frenchsetup}{
173       \frenchbsetup{StandardLists=true}}{}
174     \frenchsetup{StandardLists=True}}
175   }{}
176 }
177 \ifthenelse{\boolean{nosetlist}}{}{
178   \AtBeginDocument{% if loaded before babel package
179     \standardfrenchlists}
180   \standardfrenchlists % necessary when loaded after babel
181   \setlist[enumerate]{font=\bfseries}
182   \setlist[enumerate,1]{topsep=1.5ex plus 1ex minus 1ex,leftmargin=1.5em}
183 }
184

```

`\exenumerate`

```

185 \newenvironment{exenumerate}[1][{}]{%
186   \setlist[enumerate]{font=\bfseries}
187   \setlist[enumerate,1]{leftmargin=1.5em,
188     itemsep=3ex plus 1ex minus 1ex,topsep=3ex plus 1ex minus 1ex}
189   \setlist[enumerate,3]{noitemsep,nolistsep}
190   \setlist[itemize]{noitemsep,nolistsep}
191   \begin{enumerate} [#1]
192   }\end{enumerate}}
193

```

`tablenum1`    The `\NewTasks` command comes from the `tasks` package. It allows to define the  
`tablenuma`    environments `tablenum1`, `tablenuma` and `tablitem`. The horizontal spaces are  
              adjusted to get a good alignment with items of other `enumerate` (or `itemize`)  
              environments.

```

194 \ifthenelse{\boolean{nosetlist}}{
195   \NewTasksEnvironment[label=1.,
196     column-sep=1em,
197     after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablenum1}{\item}(2)
198   \NewTasksEnvironment[label=(a),
199     column-sep=1em,label-align=right,
200     item-indent=2.15em,label-width=1.6em,label-offset=0.5em,
201     after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablenuma}{\item}(2)
202 }{% by default
203   \NewTasksEnvironment[label=1.,label-format=\bfseries,
204     column-sep=1em,label-align=right,
205     item-indent=1.5em,label-width=1em,label-offset=0.5em,
206     after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablenum1}{\item}(2)
207   \NewTasksEnvironment[label=(a),label-format=\bfseries,
208     column-sep=1em,label-align=right,
209     item-indent=2.15em,label-width=1.6em,label-offset=0.5em,
210     after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablenuma}{\item}(2)

```

```

211     }
212 \PackageWarning{exesheet}{Environment tablenum is deprecated and replaced by tablenum1}
213
tablitem
214 \NewTasksEnvironment[label=\labelitemi,
215     label-align=right,
216     item-indent=2.3333em,label-offset=0.5em,
217     after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablitem}[\item](2)
218
colsenum
colsenum* 219 \newenvironment{colsenum}[2][{}]{%
220     \setlength{\multicolsep}{2ex}
221     \raggedcolumns % default is \flushcolumns
222     \begin{multicols}{#2} % #2 = number of columns
223     \begin{enumerate}[#1] % #1 = options of enumerate
224     }{
225     \end{enumerate}
226     \end{multicols}
227 }
228
229 \newenvironment{colsenum*}[2][{}]{%
230     \setlength{\multicolsep}{2ex}
231     \begin{multicols}{#2} % #2 = number of columns
232     \begin{enumerate}[#1] % #1 = options of enumerate
233     }{
234     \end{enumerate}
235     \end{multicols}
236 }
237
colsitem
colsitem* 238 \newenvironment{colsitem}[2][{}]{%
239     \setlength{\multicolsep}{2ex}
240     \raggedcolumns
241     \begin{multicols}{#2}
242     \begin{itemize}[#1]
243     }{
244     \end{itemize}
245     \end{multicols}
246 }
247
248 \newenvironment{colsitem*}[2][{}]{%
249     \setlength{\multicolsep}{2ex}
250     \begin{multicols}{#2}
251     \begin{itemize}[#1]
252     }{
253     \end{itemize}
254     \end{multicols}
255 }
256

```

## 6.6 Questions and answers

`\questiononly`    The `questions` and `answers` booleans control the display of corresponding environments. When created, a boolean has the `false` value by default, but we define them to `true`. The `\questiononly` and `\answeronly` macros are used as user interface to display only one of the two environments.

```
257 \newboolean{questions}
258 \newboolean{answers}
259 \setboolean{questions}{true}
260 \setboolean{answers}{true}
261 \newcommand{\questiononly}{
262     \setboolean{questions}{true}\setboolean{answers}{false}}
263 \newcommand{\answeronly}{
264     \setboolean{questions}{false}\setboolean{answers}{true}}
265
```

`questions`    It is the `\comment` and `\endcomment` macros, provided by the `versions` package, that allow the magic of conditional displays (we can also find them in the `verbatim` or `version` packages). The noteworthy `codesection` package, allows to encapsulate optional code between the macros `\BeginCodeSection{<skip>}` and `\EndCodeSection{<skip>}`, both in the text body and in the preamble, but these macros cannot be used inside an environment as we did here for `\comment` and `\endcomment`.

```
266 \newcounter{exe@ini}
267 \newcounter{subpart@ini}
268
269 \newenvironment{questions}{
270     \ifthenelse{\boolean{questions}}{
271         \setcounter{exe@ini}{\value{exercise}}
272         \setcounter{subpart@ini}{\value{subpart}}
273     }{\comment}%
274     {\ifthenelse{\boolean{questions}}{\endcomment}}
275
```

`answers`    The internal macro `\set@toclevel` calculates the title level of the word *Correction* to display at the start of an `answers` environment (when `questions` and `answers` are displayed together). The principle is to compare the state of the counters `exercise` and `subpart` with those saved at the time of the call of `questions`. The `\@enumdepth` counter indicates the `enumerate` list level in which we are (0 = out of lists). The optional parameter of the `answers` environment allows to force this title level.

```
276 \newcounter{@toclevel}
277 \newcommand{\set@toclevel}[1][1]{
278     \ifthenelse{\equal{#1}{}}{
279         \ifthenelse{\value{exercise} > \value{exe@ini}}{
280             \setcounter{@toclevel}{1}
281         }{\ifthenelse{\equal{the \@enumdepth}{0}}{
282             % we're not in an enumerate environment
283             \ifthenelse{(\value{subpart} > \value{subpart@ini})\or
284                 (\value{subpart} = 0)}{
285                 \setcounter{@toclevel}{2}
286             }{\setcounter{@toclevel}{3}}
287         }{\setcounter{@toclevel}{4}}
288     }
```

```

288     }\setcounter{@toclevel}{#1}}
289
290 \newcommand{\correctionstyle}{\color{correctioncolor}}
291
292 \newenvironment{answers}[1][\% #1 is the optional level
293     \ifthenelse{\boolean{answers}}{\%
294         \ifthenelse{\boolean{questions}}{\%
295             \set@toclevel[#1]
296             \ifthenelse{\value{@toclevel} = 1}{
297                 \section*{\correctionstyle\correctionname}
298                 \ifthenelse{\boolean{notoc}}{}{
299                     \addcontentsline{toc}{section}{\correctionname}}
300                 \setcounter{exercise}{0}
301             }{\ifthenelse{\value{@toclevel} = 2}{\%
302                 \subsection*{\correctionstyle\correctionname}
303                 \ifthenelse{\boolean{notoc}}{}{
304                     \addcontentsline{toc}{subsection}{\correctionname}}
305                 \setcounter{subpart}{0}
306             }{\ifthenelse{\value{@toclevel} = 3}{\%
307                 \subsubsection*{\correctionstyle\correctionname}
308                 \ifthenelse{\boolean{notoc}}{}{
309                     \addcontentsline{toc}{subsubsection}{
310                         \correctionname}}
311                 }\par\textbf{\correctionstyle\correctionname}\par
312             }%
313         }%
314     }%
315     \correctionstyle%
316     }\}%
317 }\comment}
318 }\ifthenelse{\boolean{answers}}{}{\endcomment}}
319
320 \newenvironment{answers*}{\ifthenelse{\boolean{answers}}{}{\comment}}%
321     {\ifthenelse{\boolean{answers}}{}{\endcomment}}
322

```

In the `answers` environment, if we place `\correctionstyle` before `\subsubsection`, the preceding vertical space may be too wide.

`\question`

```

323 \newcommand{\question}[2]{\%
324     \ifthenelse{\boolean{questions}}{#1}{
325         \bgroup
326         \ifthenelse{\boolean{answers}}{
327             \ifthenelse{\boolean{questions}}{
328                 \ifx#2\empty\else
329                     \par\correctionstyle\textbf{\correctionname}\par
330                 \fi
331             }{}
332         }#2}{
333         \egroup
334     }
335

```

`\points (patched)`

```

336 \let\@oldpoints\points
337 \renewcommand*{\points}[1]{%
338     \ifthenelse{\boolean{questions}}{\@oldpoints{#1}}{}
339 }

```

## 6.7 Marginal notes

The commands `\displaypts`, `\displaypoints` and `\displaynotes` change the ratio between left and right margins<sup>13</sup>.

```

\pts
\displaypts 340 \newboolean{marginpts}
341 \newcommand*{\pointmark}[1]{%
342     \ifthenelse{\lengthtest{#1 cm < 2cm}}{#1 \ptname}{#1 \ptsname}}
343 \newcommand{\ptsstyle}[1]{%
344     \footnotesize\centering\sffamily\color{ptscolor} (#1)}
345 \newcommand*{\pts}[1]{%
346     \ifthenelse{\boolean{marginpts}}{%
347         \mbox{%
348             \marginpar{\hspace{0pt}%
349                 \ptsstyle{\pointmark{#1}}}%
350             }%
351     \ignorespaces
352 }
353 \newcommand{\displaypts}{%
354     \reversemarginpar
355     \geometry{hmarginratio=3:2}
356     \setboolean{marginpts}{true}
357 }
358

```

`\totalex` In the following macros using `\marginpar`, percent symbols and `\ignorespaces` are necessary to avoid too much space in the text (or the margin) where these macros are inserted.

```

359 \newlength{\ptsboxlength}
360 \setlength{\ptsboxlength}{3.1em}
361 \cornersize{1}
362 \newcommand*{\totalex}[1]{%
363     \ifthenelse{\boolean{marginpoints}}{%
364         \mbox{%
365             \marginpar{\markingstyle{\ovalbox{%
366                 \makebox[\ptsboxlength]{\pointmark{#1}}}%
367             }}%
368         }%
369     \ignorespaces
370 }
371

```

`\note` The booleans `marginpoints` and `marginfullnotes` control the display of marginal notes. If `marginpoints` is false, `marginfullnotes` will be ignored. The `\noteragged` command is initialized outside `\displaynotes` such that we can use the `\note` command without `\displaynotes` (see further).

---

<sup>13</sup>So that the effect on the margin ratio is correct, these macros must be called, in the preamble, after other commands that also could alter the page geometry.

```

372 \newboolean{marginpoints}
373 \newboolean{marginfullnotes}
374
375 \newcommand{\markingstyle}[1]{\hspace{0pt}\footnotesize\sffamily%
376   \centering\color{markingcolor}\textbf{#1}}
377 \newcommand{\noteragged}{\raggedleft}
378 \newcommand{\notestyle}[1]{\hspace{0pt}\footnotesize\sffamily%
379   \noteragged\noindent\color{notecolor} #1}
380 \newcommand{\@note}[2][]{%
381   \ifthenelse{\boolean{marginpoints}}{%
382     \mbox{}%
383     \marginpar{%
384       \ifthenelse{\equal{#1}{}}{\}\{\markingstyle{#1}\}\%
385       \ifthenelse{\boolean{marginfullnotes}}{\notestyle #2}{}%
386     }%
387   }\}%
388   \ignorespaces
389 }
390 \newcommand{\@@note}[1]{%
391   \ifthenelse{\boolean{marginpoints}}{%
392     \mbox{}%
393     \marginpar{\markingstyle{#1}}%
394   }\}%
395   \ignorespaces
396 }
397 \newcommand{\note}{\@ifstar{\@@note}{\@note}}
398

```

`\displaypoints`      In `\displaynotes`, the additional length 1 in matches the default free space  
`\displaynotes`      to the left of `\oddsidemargin`.  
`\displaynotesright`

```

399 \newcommand{\displaypoints}{%
400   \reversemarginpar
401   \geometry{hmarginratio=3:2}
402   \setboolean{marginpoints}{true}
403 }
404
405 \newcommand*{\displaynotes}[1][\raggedleft]{%
406   \reversemarginpar
407   \renewcommand{\noteragged}{#1}
408   \geometry{hmarginratio=5:1}
409   \setlength{\marginparwidth}{\oddsidemargin}
410   \addtolength{\marginparwidth}{1in}
411   \addtolength{\marginparwidth}{-\marginparsep}
412   \setlength{\marginparwidth}{0.8\marginparwidth}
413   \setboolean{marginpoints}{true}
414   \setboolean{marginfullnotes}{true}
415 }
416
417 \newcommand*{\displaynotesright}[1][\raggedright]{%
418   \normalmarginpar
419   \renewcommand{\noteragged}{#1}
420   \geometry{hmarginratio=1:5}
421   \setlength{\marginparwidth}{\paperwidth}
422   \addtolength{\marginparwidth}{-\textwidth}

```

```

423 \addtolength{\marginparwidth}{-\oddsidemargin}
424 \addtolength{\marginparwidth}{-\marginparsep}
425 \addtolength{\marginparwidth}{-1in}
426 \setlength{\marginparwidth}{0.8\marginparwidth}
427 \setboolean{marginpoints}{true}
428 \setboolean{marginfullnotes}{true}
429 }

```

The formatting of marginal notes can possibly be made more regular with the `ragged2e` package, not loaded by `exesheet`. We will then have to enter, as an optional parameter of `\displaypoints`: `\RaggedLeft`, `\Centering`, `\RaggedRight` or justifying.

`\totalpoints`

```

430 \newcommand{\totalpoints}{%
431 \ifthenelse{\boolean{marginpoints}}{\totalexe}{\points}}
432 \end{package}

```