The **exframe** Package

Niklas Beisert

Institut für Theoretische Physik
Eidgenössische Technische Hochschule Zürich
Wolfgang-Pauli-Strasse 27, 8093 Zürich, Switzerland

nbeisert@itp.phys.ethz.ch

2020/01/11, v3.31

Abstract

exframe is a LaTeX 2_ε package which provides a general purpose framework to describe and typeset exercises and exam questions along with their solutions. The package features mechanisms to hide or postpone solutions, to assign and handle points, to collect problems on exercise sheets, to store and use metadata and to implement a consistent numbering. It also provides a very flexible interface for configuring and customising the formatting, layout and representation of the exercise content.

Contents

1   Introduction

This package provides a framework to describe and typeset exercises (homework problems, classroom exercises, quizzes, exam questions, exercise questions in books and lecture notes, . . .) and their solutions or answers. The aim of this package is to set up a few LaTeX environments into which questions and corresponding answers can be filled conveniently. The main task of the package is to manage the text and data that are provided in the source document, perform some common operations on them, and then output the content appropriately. The package has the following goals, tasks and features:

- The package is designed with generality in mind. It is meant to be usable in many different situations. The primary target is science and education, but it may well be useful in other areas.

- The package defines a basic functional layout for the output and provides many options to reshape the layout and formatting according to the author's needs and wishes.

- The package can handle two layers of exercises: main problems and subproblems. The use of subproblems is optional.

- The display of solutions can be configured: Solutions can be hidden for a hand-out version of exercise sheets. When displayed, they may appear immediately, collectively after the problem, at the end of each sheet or at some manually defined location.

- The package can handle exercise sheets which combine several exercise problems: A LaTeX document can consist of an individual sheet or of a collection of sheets (e.g. spanning a lecture course). In the latter case, the document files can be set up such that single sheets as well as a collection of all sheets can be compiled; the package childdoc may be of assistance.

- The package can handle points to be credited: Points will be displayed according to the layout. Overall points for a problem or a sheet can be added automatically. Points can also be stored and used elsewhere.

- The package provides an interface to specify exercise metadata (author, source, . . .): Some basic types of metadata are predefined and more specific metadata categories can be added.

- The package can use alternative counters for equations within solutions (and problems). This is to ensure a consistent numbering independently of whether solutions are output or not.

2 Usage

To use the package exframe add the command

$$\usepackage\{exframe\}$$

to the preamble of the LaTeX document.

2.1 Exercise Environments

The package provides four environments to describe the main entities of exercise problems. Additional information on the exercises can be provided in the optional arguments to these environments which will be discussed in the following sections. Furthermore, a limited set of commands is provided for control and extra features, see the sections below for details.

problem   The problem environment describes an exercise problem:

$$\begin\{problem\}[\textit{opts}]$$
$$\textit{problem text and subproblems}$$
$$\end\{problem\}$$

As one of the many available options *opts*, one can provide a title for the exercise by specifying title={*title*}. If no title is given, the problem number will be displayed instead. See section 2.3 and section 2.4 for a description of the available options.

subproblem   The subproblem environment describes a subproblem, part or an individual question of an exercise problem:

$$\begin\{subproblem\}[\textit{opts}]$$
$$\textit{subproblem text}$$
$$\end\{subproblem\}$$

A subproblem environment must be contained within a problem environment (however, a problem block need not contain subproblem blocks).

solution   The solution environment describes the solution to a problem or a subproblem:

$$\begin\{solution\}[\textit{opts}]$$
$$\textit{solution text}$$
$$\end\{solution\}$$

A solution environment should be at the end of a subproblem or problem environment (it is not mandatory to provide a solution). It can be contained within the corresponding environment or it can follow it. Depending on the choice of solution display, see section 2.2, the output may have a slightly different layout. In terms of logic, it is preferred to define a solution *within* the corresponding environment; this may also have some technical advantages and produce a slightly better result in terms of layout.

sheet   The sheet environment describes an exercise sheet:

$$\begin\{sheet\}[\textit{opts}]$$
$$\textit{sheet text and problems}$$
$$\end\{sheet\}$$

A sheet typically contains one or several problems (it is not mandatory to group problems into a sheet). There may or may not be additional auxiliary text introducing the problems. A header will be added to the sheet according to the specified layout.

2.2 Solution and Problem Display

There are several options to control the output of solutions and of problems.

solutions   Most importantly, the display of solutions can be disabled or enabled altogether:

$$\text{\textbackslash exercisesetup\{solutions[=true|false]\}}$$

Solutions are hidden by default, and their display needs to be activated explicitly (it suffices to specify the option `solutions` without the value `true`). It is also possible to control the display by an analogous package option `solutions`, see section 2.8 for further information.

\ifsolutions   The display of solutions is reflected by the conditional `\ifsolutions`. As the hiding of
onlysolutions   solutions is performed automatically, the conditional would typically be used to change some details, e.g. for adjusting titles:

$$\text{\textbackslash ifsolutions Solutions\textbackslash else Exercises\textbackslash fi}$$

Alternatively, content to be processed only in solutions mode can be enclosed in an `onlysolutions` block:

```
\begin{onlysolutions}
...
\end{onlysolutions}
```

This structure can be useful to hide auxiliary text or material if all solution content is to be stripped from a source file, e.g. by an automated `sed` filter rule (doubling of backslashes required for `sed` as well as for shell script strings):

```
sed "/\\\\begin{solution}/,/\\\\end{solution}/d;"\
"/\\\\begin{onlysolutions}/,/\\\\end{onlysolutions}/d"
```

solutionequation   As solutions can contain numbered equations while the display of solutions can be switched on and off, it is important to assign a different counter for equations within solutions in order for the equation numbers to be stable. A separate counter for equations within solutions is enabled by default. It can be disabled by:

$$\text{\textbackslash exercisestyle\{solutionequation=false\}}$$

This option prepends the letter 'S' to equation numbers within solutions which are counted separately; the display can be configured differently, see section 2.6.

solutionbelow   The package allows to collect solutions and defer their display to particular locations:
\insertsolutions

$$\text{\textbackslash exercisestyle\{solutionbelow=}pos\}$$

The available choices for *pos* are to display solutions where they are defined (`here`), defer them to the end of the current subproblem (`subproblem`), problem (`problem`) or sheet (`sheet`) or display them at a manually chosen location (`manual`). Note that typically solutions are defined at the end of a (sub)problem and therefore the choice `here` is similar to (`sub`)`problem`. The latter form, however, makes sure that a solution does not inherit the margin of the parent environment. The alternate modes `problem*` and `subproblem*` positions the solution *after* the (sub)problem environment such that it does not inherit any layout, but also no definitions made in the parent environment. In `manual` mode, all solutions are collected (with appropriate headers) until they are output by the directive `\insertsolutions`. If no solutions are stored in the buffer (or if the mode is not `manual`), `\insertsolutions` has no effect.

4

\writesolutions   Another option to handle solutions is to write them to a file for later use. Writing to a file is initiated by:

$$\text{\texttt{\textbackslash writesolutions}}[\textit{filename}]$$

The optional argument describes the filename as *filename*`.sol`; no argument defaults to the main tex filename as `\jobname.sol`; the extension `.sol` can be customised by the configuration `extsolutions`. This mode overrides the `solutionbelow` behaviour described above; all subsequent solutions are written to the file. The file is closed by `\closesolutions` and the display of solutions returns to manual mode. It is not necessary to close a file as it will be closed automatically by reading from a file, writing to another file or by the end of the document.

\readsolutions   Solutions are read from a file by:

$$\text{\texttt{\textbackslash readsolutions}}[\textit{filename}]$$

This command outputs a sectional title and reads the file via `\input{`*filename*`.sol}`.

solutionbuf
problembuf
The package offers similar functionality to control the display of problems. In order to have any control over the content of `problem` environments, the latter needs to be read into an internal buffer. Reading of solutions and problems to internal buffers is activated or deactivated by:

$$\text{\texttt{\textbackslash exercisesetup\{solutionbuf[=true|false]\}}}$$
$$\text{\texttt{\textbackslash exercisesetup\{problembuf[=true|false]\}}}$$

By default, `solution` environments are read to an internal buffer, while the content of `problem` environments is processed directly by the TEX engine. Therefore, the following options to control the display of `problem` environments require the statement `\exercisesetup{problembuf}`.

problemmanual
\insertproblems
The immediate display of `problem` environments is controlled by:

$$\text{\texttt{\textbackslash exercisestyle\{problemmanual[=true|false]\}}}$$

In the default automatic mode, problems are displayed directly where they are declared. In manual mode, problems are collected to an internal buffer, and only displayed by issuing `\insertproblems`.

Note that `solution` environments should be declared within the corresponding `problem` environment in order to preserve their appropriate association. The `solution` environment is then processed at the place where the `problem` environment is displayed, and it may (or may not) be deferred further.

\writeproblems
\readproblems
Problems can be written out to an external file for later usage. The functionality is analogous to solutions and uses the macros:

$$\text{\texttt{\textbackslash writeproblems}}[\textit{filename}]$$
$$\text{\texttt{\textbackslash readproblems}}[\textit{filename}]$$

The optional argument describes the filename as *filename*`.prb`; no argument defaults to the main tex filename as `\jobname.prb`; the extension `.prb` can be customised by the configuration `extproblems`.

disable
insertproblemselect
The display of a particular `problem` can be suppressed altogether by an optional argument:

$$\text{\texttt{\textbackslash begin\{problem\}[disable]}}$$

This option can be exploited to automatically suppress certain classes of problems as follows: A hook function `insertproblemselect` declared by:

$$\verb|\exerciseconfig{insertproblemselect}[1]{|\mathit{code}\verb|}|$$

can call `\setproblemdata{disable}` whenever a problem is to be suppressed. In order to decide, the optional argument of the `problem` environment is passed on to the hook function as the single argument. Note that the argument needs to be processed manually.

2.3   Metadata

In a collection of exercise problems it makes sense to keep track of metadata for the overall collection as well as for individual problems and potentially display some of them. The framework defines a standard set of metadata fields and offers functionality to add more specialised metadata fields.

`\exercisedata`   Global metadata is specified by the command:

$$\verb|\exercisedata{|\mathit{data}\verb|}|$$

The argument *data* is a comma-separated list of metadata specifications in the form *key*=*{value}*. The standard set of global metadata keys consists of:

- `author`: principal author(s) of the exercise collection; also invokes the LaTeX command `\author`; will be written to pdf documents.
- `title`: title of the exercise collection; also invokes the LaTeX command `\title`; will be written to pdf documents.
- `date`: date of the exercise collection; also invokes the LaTeX command `\date`; will be written to pdf documents.
- `subject`: subject area of the exercise collection; will be written to pdf documents.
- `keyword`: keyword(s) for the exercise collection; will be written to pdf documents.
- `course`: title of the course (class, lecture, module, . . .) for the exercise collection.
- `institution`: institution (school, department, institute, university, . . .) offering the course or exercise collection.
- `instructor`: instructor(s) for the course or exercise; this field refers to person(s) who organise the corresponding course or exercises whereas `author` refers to the principal creator of the material.
- `period`: period (year, season, date, term identifier, . . .) of the corresponding course.
- `material`: type of material (exercises, homework assignments, exam, quizzes, solutions, . . .).

`\defexercisedata`   Additional custom fields for global metadata can be created with:

$$\verb|\defexercisedata{|\mathit{key}\verb|}|$$

`\getexercisedata`
`\exercisedataempty`   Global metadata should typically be specified somewhere at the top of the main document, and it can be inserted wherever needed. There are two commands to read and process metadata. To insert the value of metadata field *key* use:

$$\verb|\getexercisedata{|\mathit{key}\verb|}|$$

In some situations the output should depend on whether a metadata has been filled (e.g. to fill a default value or to display something else instead). This can be checked with the conditional:

$$\text{\textbackslash exercisedataempty}\{key\}\{empty\ code\}\{filled\ code\}$$

The *empty code* is executed if no value or an empty value has been specified; otherwise the *filled code* is executed.

sheet
problem
The package offers a similar mechanism to describe and use metadata for sheets and problems:

$$\text{\textbackslash begin\{sheet\}}[opts]$$
$$\text{\textbackslash begin\{problem\}}[opts]$$

The argument *opt* is a comma-separated list which can contain metadata specifications in the form *key*={*value*}. The standard set of metadata keys for sheets consists of:

- `due`: indication of the due date for the exercise sheet.
- `handout`: indication of the handout date for the exercise sheet.
- `title`: specifies a title for the sheet; when reading value (see below), returns composed title; untitled sheets will be displayed by their number; title will be written to pdf documents.
- `rawtitle` (for reading only): contains the raw title as specified by `title`.
- `author`: author(s) of the sheet; will be written to pdf documents.
- `editor`: editor(s) of the sheet; this field refers to a person who makes adjustments to the sheet whereas `author` refers to the creator of the sheet.
- `editdate`: indication of the date when the sheet was last edited.

The standard set of metadata keys for problems consists of:

- `title`: specifies a title for the problem; when reading value (see below), returns composed title; untitled problems will be displayed by their number.
- `rawtitle` (for reading only): contains the raw title as specified by `title`.

\defsheetdata
\setsheetdata
\getsheetdata
\sheetdataempty
\defproblemdata
\setproblemdata
\getproblemdata
\problemdataempty
Metadata for sheets can be used in the same way as the global metadata. The following directives are analogous to \defexercisedata, \exercisedata, \getexercisedata and \exercisedataempty:

$$\text{\textbackslash defsheetdata}\{key\}$$
$$\text{\textbackslash setsheetdata}\{data\}$$
$$\text{\textbackslash getsheetdata}\{key\}$$
$$\text{\textbackslash sheetdataempty}\{key\}\{empty\ code\}\{filled\ code\}$$

$$\text{\textbackslash defproblemdata}\{key\}$$
$$\text{\textbackslash setproblemdata}\{data\}$$
$$\text{\textbackslash getproblemdata}\{key\}$$
$$\text{\textbackslash problemdataempty}\{key\}\{empty\ code\}\{filled\ code\}$$

pdfdata
\writeexercisedata
The most relevant metadata can be written to the metadata section of pdf files (using pdfLaTeX and the package hyperref whenever loaded). This feature is configured by:

$$\text{\textbackslash exercisesetup\{pdfdata}[=auto|manual|sheet|off]\}$$

The option `auto` writes the global metadata `title`, `author`, `subject` and `keyword` to the corresponding fields in the pdf file. To make this work, these must be defined before the `\begin{document}` directive. The option `manual` allows to manually write these metadata by the command `\writeexercisedata`. It should be issued after the metadata have been set, but before any content is written to the pdf file. In other words, it can be anywhere in the document preamble directly after `\begin{document}`, or following a couple of content-free definitions at the beginning of the document body (in case the metadata should be set within the document body for some reason). The option `sheet` writes out the metadata at the beginning of the first `sheet` environment (which should follow `\begin{document}` without any content in between). This option is primarily for filling the `author` and `title` fields with metadata of a sheet rather than a collection of exercises. Note that if no `author` is defined for the sheet, the global metadata `author` is used. The option `off` disables all writing of metadata.

`problem`
`subproblem`
`solution`

There is an additional mechanism to keep track of metadata for problems, subproblems and solutions which can be displayed in the opening line of these entities. Displayed metadata serve two purposes: they are used to describe the quality of a problem or they are intended for internal documentation purposes. Their output can be controlled individually, e.g. only in development versions of a document. Note that specifying a key more than once will display the content multiple times in the order in which they are encountered. Displayed metadata are specified at the top of the corresponding environment:

$$\begin{array}{l} \texttt{\textbackslash begin\{problem\}[\textit{opts}]} \\ \texttt{\textbackslash begin\{subproblem\}[\textit{opts}]} \\ \texttt{\textbackslash begin\{solution\}[\textit{opts}]} \end{array}$$

The standard set of displayed metadata keys consists of:

- `author`: author(s) of the problem (or subproblem, solution).
- `editor`: editor(s) of the problem; this field refers to a person who has made adjustments to the problem whereas `author` refers to the creator of the problem.
- `source`: source of the problem; in case the problem has been taken from elsewhere (conceptually or literally).
- `difficulty`: indication of the level of difficulty of the problem.
- `keyword`: keyword(s) for the problem;
- `comment`: some comment on the problem.
- `optional` (display enabled by default): whether addressing the problem is mandatory or optional; by default the text will be displayed after the title in italic shape.

By default, only the `optional` items are displayed, all other types of items are hidden; controlling the display for each type of item is described below.

`extdata`
Further displayed metadata keys are defined by the package option `extdata`, see section 2.8:

- `review`: field to review the aspects of the problem (quality, length, appropriateness, difficulty, . . .).
- `recycle`: indication of previous instances where this problem was used.
- `timesolve`: indication of the time needed to solve this problem (or subproblem).
- `timepresent`: indication of the time needed to present this problem (or subproblem, solution).

`\showprobleminfo`
The display of the above metadata fields for a problem (or subproblem, solution) is controlled by:

$$\backslash\text{showprobleminfo}\{keys\}$$

Here *keys* is a comma-separated list of keys to be activated (*key* or *key*=`true`) or deactivated (*key*=`false`).

\defprobleminfo Displayable metadata can be defined or adjusted by:

$$\backslash\text{defprobleminfo}\{key\}\{code\}$$

Here *key* specifies the metadata field and *code* the code to display this type of metadata where the argument `#1` represents the data to be displayed.

insertprobleminfo
insertsubprobleminfo
insertsolutioninfo
\addprobleminfo
\addprobleminfo*

Additional information can be injected into the opening line of problems and solutions by the definitions:

$$\backslash\text{exerciseconfig}\{\text{insertprobleminfo}\}\{code\}$$
$$\backslash\text{exerciseconfig}\{\text{insertsubprobleminfo}\}\{code\}$$
$$\backslash\text{exerciseconfig}\{\text{insertsolutioninfo}\}\{code\}$$

The hook code *code* will be called after processing the environment arguments. Information can be added to the opening line by:

$$\backslash\text{addprobleminfo}\{info\}$$
$$\backslash\text{addprobleminfo}*\{info\}$$

The unstarred command adds information at the end of the opening line, the starred version at the beginning (but after the title or identifier).

2.4   Points

points Exercise problems or certain parts of them can be credited with points (credits, awards, ...). The package provides an interface to specify and manage such points. Points are declared by the option `points=`*points* for the environments `sheet`, `problem` and `subproblem`. These numbers will be printed to the opening line of problems and subproblems.

Note that the points should normally be integer numbers. Fractional points are permissible as well, but the internal storage by the TeX engine is somewhat limited, so that only fractions with powers of two as denominators (.5, multiples of .25, .125, .0625, ...) are reliable. More general fractional decimal numbers such as multiples of 0.2 will be subject to rounding errors and will not display nicely.

Bonus points can be specified in the format `points=[`*regular*`][+`*bonus*`]`. By default, such points will be printed as [*regular*][+*bonus*] where 0 components are omitted.

problempointsat
subproblempointsat
solutionpointsat

The location where points of problems and subproblems shall be displayed can be adjusted individually by:

```
\exercisestyle{problempointsat=start|start*|margin|end|manual|off}
\exercisestyle{subproblempointsat=start|start*|margin|end|manual|off}
\exercisestyle{solutionpointsat=start|start*|margin|end|manual|off}
```

The default values are `start` and `end` for problems and subproblems, respectively. The option `start` displays points at the very end of the opening line; the option `start*` displays them at the start of it. The option `end` displays points at the end of the problem or subproblem text. The option `margin` displays points in the margin. The option `manual` displays points at a manually chosen location specified by the directive `\showpoints`. Note that `\showpoints` can also be used for the option `end` to display the points prematurely

(e.g. if the text ends with a displayed equation, it may make sense to display the points just before the equation). The option `off` disables the display of points.

\getsheetdata   Points for sheets are only stored by the package; they must displayed manually. Within the corresponding `sheet` environment the points can be accessed by:

$$\verb|\getsheetdata{points}|$$

\getsheetpoints The package allows to read the point totals for other sheets and problems:
\getproblempoints
\getsubproblempoints
\getsolutionpoints
\extractpoints
\switchpoints

$$\verb|\getsheetpoints{|tag\verb|}|$$
$$\verb|\getproblempoints{|tag\verb|}|$$
$$\verb|\getsubproblempoints{}|$$
$$\verb|\getsolutionpoints{}|$$

Here *tag* is the tag assigned to the corresponding sheet or problem, see . An empty argument *tag* refers to the current sheet, (sub)problem or solution. If bonus points are used, the points will be returned in the format [*regular*][+*bonus*]; the components *regular* and *bonus* can be extracted from the returned expression by `\extractpoints` and `\extractpoints*`, respectively. A convenient case switch of the returned value can be performed by:

$$\verb|\switchpoints{|reg\verb|}{|bonus\verb|}{|both\verb|}{|none\verb|}{|val\verb|}|$$

Here *val* is the value returned from the points register, *reg* is displayed for purely regular points, *bonus* is displayed for purely bonus points, *both* is displayed for mixed points, *none* is displayed for no points. In each of the four expressions, `#1` will be replaced by the regular points and `#2` by the bonus points.

\awardpoints   Grading instructions with points to be awarded can be specified in the solution text by:

$$\verb|\awardpoints[|details\verb|]{|points\verb|}|$$
$$\verb|\awardpoints*[|details\verb|]{|points\verb|}|$$

Here *details* is an optional text with further details, e.g. to explain under which conditions these points are to be awarded. The starred form is used to specify optional points or alternative paths with alternative grading instructions. These points will be marked and not be used for the computation of a total.

warntext   The package attempts to add up the points of subproblems to the problem total and likewise the points of problems to the sheet total. The package also performs some sanity checks on the provided numbers: If points are specified for both subproblems and problems or for both problems and sheets, they will be compared. Also the points within solutions (excluding optional or alternative points) are added up and compared to the corresponding problem or subproblem. Furthermore the package checks whether points are defined for all subproblems within a problem or all problems within a sheet. Mismatches are reported as package warnings. As point mismatches can be rather severe, there is an option to write such warnings directly into the output document (to be removed before distribution):

$$\verb|\exercisesetup{warntext[=true|false]}|$$

fracpoints   The package offers pretty display of fractional points with denominators 2, 4 and 8 by writing the decimal part as a fraction, e.g. $1.75 \rightarrow 1^3/_4$. This feature is enabled by:

$$\verb|\exercisestyle{fracpoints}|$$

2.5 Labels and Tags

label LaTeX provides labels to make references to remote parts of the text. Labels can be set as usual by `\label{`*label*`}` within the problem, subproblems and sheet environments. Alternatively, they can be specified as the environment option:

$$\texttt{label=\{}label\texttt{\}}$$

tag The package provides an additional mechanism to tag sheets and problems. Each sheet
\sheettag and each problem can be assigned a unique tag *tag* by the environment option:
\problemtag

$$\texttt{tag=\{}tag\texttt{\}}$$

This tag is used for reading point totals as described in . Furthermore, the macro `\sheettag` or `\problemtag` is set to the tag *tag* within the current environment. If no tag is specified it matches the number of the sheet or problem; note that this number can change by reordering sheets and problems and therefore it should not be used to identify the entity from other parts of the document.

A useful application for tags is to encapsulate labels within individual sheets and problems which are part of a collection of exercises. Labels which are composed as `\sheettag`-*label* or `\problemtag`-*label* can be considered local and will not clash with labels defined within a different environment. Within the same sheet or problem, local labels can be accessed by the same construction. They can also be accessed from remote parts of the document by fully expanding `\sheettag` or `\problemtag` for the desired target environment.

autolabelsheet If unique tags are specified, the package can automatically create labels for sheets
autolabelproblem (sheet:*tag*) and problems (prob:*tag*) by:

$$\texttt{\textbackslash exercisesetup\{autolabelsheet[=true|false]\}}$$
$$\texttt{\textbackslash exercisesetup\{autolabelproblem[=true|false]\}}$$

2.6 Layout

The package provides a large number of parameters to adjust the display of exercises to a desired layout.

\exerciseconfig Configuration settings are declared and modified by the command:

$$\texttt{\textbackslash exerciseconfig\{}key\texttt{\}[}narg\texttt{]\{}value\texttt{\}}$$

Here *key* is a key and *value* is its assigned value. Configuration options can also be macros with arguments in which case *narg* is the number of arguments and *value* is the macro definition using arguments `#`*n*. The command `\exerciseconfig` therefore is analogous to `\(re)newcommand` except that the definitions are encapsulated by the package and any previous definition is overwritten without checking.

\exerciseconfigappend In some cases it may be useful to be able to append or prepend to a (parameterless) definition
\exerciseconfigprepend by:

$$\texttt{\textbackslash exerciseconfigappend\{}key\texttt{\}\{}value\texttt{\}}$$
$$\texttt{\textbackslash exerciseconfigprepend\{}key\texttt{\}\{}value\texttt{\}}$$

\getexerciseconfig Configuration definitions can be read by:
\exerciseconfigempty

$$\texttt{\textbackslash getexerciseconfig\{}key\texttt{\}[}arguments\texttt{]}$$

The number of arguments after {*key*} must match the optional argument *nargs* of the definition. Furthermore, it can be checked whether a configuration definition is empty:

$$\texttt{\textbackslash exerciseconfigempty\{}\textit{key}\texttt{\}\{}\textit{empty code}\texttt{\}\{}\textit{filled code}\texttt{\}}$$

The *empty code* is executed if no value or an empty value has been specified. Otherwise the *filled code* is executed.

The package defines numerous layout configuration options. They are listed along with their original definition and a brief description in . They include options to:

- adjust the language for the principal entities of this package like 'sheet(s)', 'problem(s)', 'solution(s)', 'points(s)';
- adjust the fonts styles of various parts of the text;
- adjust the spacing above, below, between various elements;
- define code to process data and insert text at various locations;
- compose text to be used in various situations;
- adjust the appearance of counters;
- adjust some other behaviour of the package.

The following will highlight only few examples.

insertsheettitle   An important setting is:

$$\texttt{\textbackslash exerciseconfig\{insertsheettitle\}\{}\textit{code}\texttt{\}}$$

The code *code* is meant to print the title or header of an exercise sheet. The minimalistic default code `\centerline{\getsheetdata{title}}` merely prints the sheet title "Sheet #" at the centre of a line. Commonly, one would replace this by a more elaborate header (potentially with some more information, appealing layout, logos, . . .). In order to design a header template, it makes sense to retrieve data via `\getexercisedata` and `\getsheetdata` described in . Likewise `\exercisedataempty` and `\sheetdataempty` can be used to display default values or alternative data if some particular data is not provided. An example is given by the `plainheader` extended style option defined in .

composetitleproblem   Another noteworthy example is `composetitleproblem` to compose the title for a problem. It takes two parameters, the number and the title. The (somewhat simplified) default declaration is:

```
\exerciseconfig{composetitleproblem}[2]{\exerciseifempty{#2}
  {\getexerciseconfig{termproblem}
   \getexerciseconfig{composeitemproblem}{#1}}
  {\getexerciseconfig{composeitemproblem}{#1} #2}}
```

This checks whether the title is empty. If no title is given use "Problem #.", otherwise use "#. *title*". Here the term "Problem" is made abstract by the configuration `termproblem` (e.g. to support internationalisation) and the problem number is further composed obtained by the configuration `composeitemproblem` which takes the bare number as argument and returns it followed by a dot.

\exerciseifempty   Handy conditionals command to check whether an expression *expr* is empty are:
\exerciseifnotempty

$$\texttt{\textbackslash exerciseifempty\{}\textit{expr}\texttt{\}\{}\textit{empty code}\texttt{\}\{}\textit{filled code}\texttt{\}}$$
$$\texttt{\textbackslash exerciseifnotempty\{}\textit{expr}\texttt{\}\{}\textit{filled code}\texttt{\}}$$

12

Their main purpose is to test whether some provided expression *expt* is empty. They expand to the common TeX constructs `\if&#1&#2\else#3\fi` and `\if&#1&\else#2\fi` which work assuming that *expr* is not too exotic (e.g. it should not start with the character '`&`' and other special TeX characters or macros are potentially dangerous).

2.7 Exercise Styles

The package provides a mechanism to define exercise styles which customise the display of exercises in some coordinated fashion.

`\exercisestyle` Style(s) are activated by the command:

$$\text{\texttt{\textbackslash exercisestyle}\{}styles\texttt{\}}$$

Here *styles* is a comma-separated list of styles, where each style is given by a pair *style*[`={`*argument*`}`]. The package defines a couple of standard styles:

- `solutionbelow=`*pos* (can take values `here`, `subproblem`, `subproblem*`, `problem`, `problem*`, `sheet` and `manual`; initially set to `subproblem`) – positions the solutions below the indicated environments; see section 2.2 for details.
- `problempointsat=`*pos* (can take values `start`, `start*`, `margin`, `end` and `manual`; initially set to `start`) – displays points in problems at the indicated location; see section 2.4 for details.
- `subproblempointsat=`*pos* (can take values `start`, `start*`, `margin`, `end` and `manual`; initially set to `end`) – displays points in subproblems at the indicated location; see section 2.4 for details.
- `solutionpointsat=`*pos* (can take values `start`, `start*`, `margin`, `end` and `manual`; initially set to `end`) – displays points in solutions at the indicated location; see section 2.4 for details.
- `problemby=`{*counter*} – number problems with the prefix *counter*, i.e. reset the problem counter whenever *counter* increases and use a composite label *counter.problem* to identify problems.
- `equationby=`{*counter*} – number the dedicated equation counters for sheets, problems and solutions with the prefix *counter*.
- `problembysheet` – number problems by sheet.
- `equationbysheet` – number dedicated equations for sheets, problems and solutions by sheet; note that the main equation counter is unaffected by this setting, it therefore makes sense to also activate the style `sheetequation` or use `\counterwithin{equation}{sheet}`.
- `pagebysheet` – number pages by sheet and denote pages by *sheet.page*; this style is useful to generate stable page numbers for a collection of sheets.
- `sheetequation`[`=true|false`] (no value implies `true`, initially set to `false`) – use a dedicated equation counter within sheets.
- `problemequation`[`=true|false`] (no value implies `true`, initially set to `false`) – use a dedicated equation counter within problems.
- `solutionequation`[`=true|false`] (no value implies `true`, initially set to `true`) – use a dedicated equation counter within solutions.
- `fracpoints`[`=true|false`] (no value implies `true`, initially set to `false`) – display fractional points for denominators 2, 4, 8; see section 2.4 for details.

- **twoside**[=true|false] (no value implies `true`, initially set to `false`) – enable/disable two-sided layout; in two-sided layout, sheets will start on odd pages and empty pages are added at the end of sheets to produce an even number of pages.

extstyle Further exercise styles are defined by the package option `extstyle`, see section 2.8:

- **plainheader** – define a plain sheet header to display some essential exercise and sheet data: `course`, `institution`, `instructor`, `period` (optional), sheet `title`, see section 2.3; the line below the header, font styles and spaces can be adjusted, see the definition in section C.4.
- **contents** – display sheets and problems in the table of contents (as sections and subsections).
- **solutionsf** – display solutions in sans serif font family.
- **solutiondimproblem** – dim the problem text whenever solutions are displayed.
- **solutionsep** – separate the solutions from the remaining text by horizontal lines.

\defexercisestyle Custom styles can be defined by:

$$\texttt{\textbackslash defexercisestyle}\{style\}\{init\}$$
$$\texttt{\textbackslash defexercisestylearg}[default]\{style\}\{init\}$$

This feature can be used to predefine certain aspects of the exercises layout. For example, different default page layouts could be declared in this way. The first version declares a style which is initialised by the code *item* upon activation by `\exercisestyle{`*style*[=true]`}`. Note that `\exercisestyle{`*style*=false`}` does nothing. The second version declares a style which is activated by `\exercisestyle{`*style*[={*arg*}]`}` and which calls *item* with the argument `#1` referring to *arg* (or *default* if no argument is given).

2.8 Package Options

\exercisesetup Features and options of general nature can be selected by the commands:

$$\texttt{\textbackslash usepackage}[opts]\texttt{\{exframe\}}$$
$$\text{or} \quad \texttt{\textbackslash PassOptionsToPackage}\{opts\}\texttt{\{exframe\}}$$
$$\text{or} \quad \texttt{\textbackslash exercisesetup}\{opts\}$$

`\PassOptionsToPackage` must be used before `\usepackage`; `\exercisesetup` must be used afterwards. *opts* is a comma-separated list of options.

The following options are available only when loading the package, i.e. they will not work within `\exercisesetup`:

- **extdata**[=true|false] (no value implies `true`, initially set to `false`) – define some more advanced metadata entries.
- **extstyle**[=true|false] (no value implies `true`, initially set to `false`) – define some more advanced styles.
- **problemenv**=*name* – redefine environment name `problem`. This and the following alike options may be useful in quickly adjusting existing sources to the exframe framework if the original framework works similarly and no special features are used. Otherwise, it is highly advisable to leave the names of environments and counters defined by the package untouched.
- **subproblemenv**=*name* – redefine environment name `subproblem`.

- solutionenv=*name* – redefine environment name `solution`.
- sheetenv=*name* – redefine environment name `sheet`.
- problemcounter=*name* – redefine counter name `problem`.
- subproblemcounter=*name* – redefine counter name `subproblem`.
- solutioncounter=*name* – redefine counter name `solution`.
- sheetcounter=*name* – redefine counter name `sheet`.

The following options can be specified by all three methods described above:

- solutions[=true|false] (no value implies `true`, initially set to `false`) – Enable/disable display of solutions. Sets the conditional `\ifsolutions` accordingly.
- pdfdata[=auto|manual|sheet|off] (no value implies `auto`, initially set to `auto`) – control writing most relevant metadata to pdf files; has no effect without package `hyperref`.
- lineno[=true|false] (no value implies `true`, initially set to `false`) – enable/disable writing of line numbers as comments into solution files.
- twoside[=true|false] (no value implies `true`, initially set to `false`) – enable/disable two-sided layout; see section 2.7 for details.
- solutionhref[=true|true] (no value implies `true`, initially set to `false`) – enable/disable use of hyper-references from solutions to the corresponding problems; has no effect without package `hyperref`.
- warntext[=true|false] (no value implies `true`, initially set to `false`) – enable/disable writing of relevant warning messages (points mismatch, point sums require update) into the document output for easier detection.
- autolabelsheet[=true|false] (no value implies `true`, initially set to `false`) – enable/disable automatically assigning labels (`sheet:\sheettag`; can be adjusted) to sheets according to their tag `\sheettag`.
- autolabelproblem[=true|false] (no value implies `true`, initially set to `false`) – enable/disable automatically assigning labels (`prob:\problemtag`; can be adjusted) to problems according to their tag `\problemtag`.
- solutionbuf[=true|false] (no value implies `true`, initially set to `true`) – enable/disable buffering for `solution` environments in order to control their display; disabling buffering can be helpful in debugging faulty `solution` environments; it might also resolve some tokenisation issues in special circumstances; note that the display of solutions cannot be suppressed with `\exercisesetup{solutions=false}` when buffering if disabled.
- problembuf[=true|false] (no value implies `true`, initially set to `false`) – enable/disable buffering for `problem` environments in order to control their display.

3 Information

3.1 Copyright

This work has the LPPL maintenance status 'maintained'.

The Current Maintainer of this work is Niklas Beisert.

This work consists of the files `README.txt`, `exframe.ins` and `exframe.dtx` as well as the derived files `exframe.sty`, `exfsamp.tex`, `exfserm.tex`, `exfser`*nn*`.tex` (*nn*=01, 02, 03, aa), `exfserpe.tex`, `exfserpf.tex`, `exfsermk.sh`, `exfsermk.mak` and `exframe.pdf`.

3.2 Files and Installation

The package consists of the files:

`README.txt`	readme file
`exframe.ins`	installation file
`exframe.dtx`	source file
`exframe.sty`	package file
`exfsamp.tex`	sample file
`exfserm.tex`	multipart sample main file
`exfser01.tex`	multipart sample sheet 1
`exfser02.tex`	multipart sample sheet 2
`exfser03.tex`	multipart sample sheet 3
`exfseraa.tex`	multipart sample unused problems
`exfserpe.tex`	multipart sample problem E
`exfserpf.tex`	multipart sample problem F
`exfsermk.sh`	multipart sample compile script
`exfsermk.mak`	multipart sample makefile
`exframe.pdf`	manual

The distribution consists of the files `README.txt`, `exframe.ins` and `exframe.dtx`.

- Run (pdf)LaTeX on `exframe.dtx` to compile the manual `exframe.pdf` (this file).
- Run LaTeX on `exframe.ins` to create the package `exframe.sty` and the samples consisting of `exfsamp.tex`, `exfserm.tex`, `exfser01.tex`, `exfser02.tex`, `exfser03.tex`, `exfseraa.tex`, `exfserpe.tex`, `exfserpf.tex`, `exfsermk.sh`, `exfsermk.mak`. Copy the file `exframe.sty` to an appropriate directory of your LaTeX distribution, e.g. *texmf-root*/`tex/latex/exframe`.

3.3 Related Packages

The package makes use of other packages available at CTAN:

- This package relies on some functionality of the package verbatim to read verbatim code from the LaTeX source without expansion of macros. Compatibility with the verbatim package has been tested with v1.5q (2014/10/28).
- This package uses the package xkeyval to process the options for the package, environments and macros. Compatibility with the xkeyval package has been tested with v2.7a (2014/12/03).
- This package can use the package hyperref to include hyperlinks between problems and solutions. Compatibility with the hyperref package has been tested with v6.88e (2018/11/30).
- This package can use the package amstext (which is automatically loaded by amsmath) to display text within equations. Compatibility with the amstext package has been tested with v2.01 (2000/06/29).

- This package uses the command `\currfilename` provided by the package currfile (if available and loaded) to indicate the LATEX source file in the generated metapost file. Compatibility with the currfile package has been tested with v0.7c (2015/04/23).

There are several other LATEX packages which offer a similar functionality varying largely in scope and sophistication:

- The package exsheets and its successor xsim provide a LATEX 3 style for typesetting exercises with solutions. They offer options to hide or delay solutions, print only specific problems, deal with points, specify metadata, handle exercise collections, as well as some more specific options. They allow to adjust the layout and choose among predefined ones.
- The package exercise provides a style for typesetting exercises with solutions. It offers many options to hide or delay solutions, print only specific problems, specify some metadata as well as some more specific options. It allows to customise the layout.
- The package exercises provides a style for typesetting exercises with solutions. It offers options to hide solutions and deal with points. It allows basic customisation of the layout.
- The package exam provides a document class for typesetting exams conveniently. It offers many options to hide solutions, deal with points and deal with other exam-specific tasks. It allows to adjust the layout and choose among predefined ones.
- The package probsoln provides a style for typesetting exercises with solutions which are stored in a collection. It offers options to hide solutions and to assemble problems from an external collection.
- The packages uebungsblatt, uassign, mathexam, exsol, homework, jhwhw provide basic functionality for somewhat more particular situations.

See CTAN categories exercise and exam for further up-to-date packages.

The philosophy of the present package is to define a low-level framework to describe exercises with solutions to be used in various situations. The aim is to provide the means to describe the content (problems, solutions, sheets) in a simple fashion and separate it from the various layout definitions and choices which will define the appearance of the content. The interface was designed to reduce potential conflict with other packages and definitions. The package itself does not define an elaborate layout, but it provides means to adjust it in many ways and to predefine custom layout schemes. The package offers most of the functionality of the above packages, but (presently) misses out on some more advanced features, see section 3.4.

3.4   Feature Suggestions

The following is a list of features which may be useful for future versions of this package:

- Add a section on useful combinations of customisation settings to achieve specific goals. Please send suggestions.
- Option to hide problem text while maintaining access to embedded solutions (for a version containing only solutions): this is difficult to implement because the problem environment cannot simply be discarded, but would have to be scanned very carefully for the embedded solution; instead process problems to some document and save solutions to file, then read solutions from different document.
- Define structures for multiple-choice questions.

3.5   Revision History

v3.31:   2020/01/11

- `onlysolutions` environment for solution mode content
- sample multipart setup streamlined

v3.3:   2019/06/15

- control display of `problem` environments via package option `problembuf`: manual display, write to file, disable individual problems
- `solutionbelow` mode `here*` superseded by package option `solutionbuf`
- display total points within solution: `solutionpointsat` (thanks to Till Bargheer for suggestion)
- read points for current sheet, (sub)problem and solution (thanks to Johannes Hahn for suggestion)
- case switch for bonus points (thanks to Johannes Hahn for suggestion)
- option to `disable` particular problems, control by hook function (thanks to Manuel Benz for suggestion)
- provided interface `\showfracpoints` and `\exerciseconfig{frac}` for fractional points display
- filename extensions configurable

v3.2:   2019/05/01

- bonus points can be specified as `points=`[*regular*][+*bonus*]
- `solutionbelow` mode `here*` added for direct processing of the solution environment
- multipart sample added

v3.11:   2019/04/15

- fix interaction with package calc (thanks to Johannes Hahn for bug report)
- fix style `fracpoints` in combination with some [`sub`]`problempointsat` choices (thanks to Johannes Hahn for bug report)
- fix spacing for [`sub`]`problempointsat=margin` (thanks to Johannes Hahn for bug report)

v3.1:   2019/01/21

- alternate placement modes for solutions
- fixed expansion of problem title
- reset font size for problem text

v3.0: 2019/01/16

- renamed to `exframe.sty`
- first version published on CTAN
- overhaul and streamline interface
- solution processing remodelled
- changed metadata handling
- changed and generalised points handling
- generalised sectioning layout
- changed layout specification model
- insert hyperlinks using `hyperref`
- manual, example and installation package added

v2.0 − v2.6: 2014/10/03 − 2018/11/05

- changed metadata interface
- broadened scope
- added more layout options
- added more metadata
- added sheet and problem tags
- add and remember points

v1.1 − v1.6: 2014/08/07 − 2014/09/14

- renamed to `nbprob.sty`
- added metadata
- added points
- added layout configuration
- removed specific macros

v1.0 − v1.02: 2011/09/23 − 2013/03/17

- first version as `problems.cls`
- dedicated layout and macros for author's exercise sheets

A   Standalone Sample

This section provides an example of how to use some of the exframe features. The resulting layout will be somewhat messy due to a random selection of features.

This example file describes a single exercise sheet. The other sheet of the series would be declared analogously in independent documents.

Preamble.   Standard document class:

```
1 \documentclass[12pt]{article}
```

Use package geometry to set the page layout; adjust the paragraph shape:

```
2 \usepackage{geometry}
3 \geometry{layout=a4paper}
4 \geometry{paper=a4paper}
5 \geometry{margin=2.5cm}
6 \parindent0pt
7 \parskip0.5ex
```

Include amsmath, hyperref and the exframe package:

```
8  \usepackage{amsmath}
9  \usepackage{hyperref}
10 \usepackage[extstyle]{exframe}
```

Solutions Switch.   It will be useful to have the switch to turn on/off the display of solutions near the top of the source file, potentially with the opposite setting commented out:

```
11 \exercisesetup{solutions=true}
12 %%\exercisesetup{solutions=false}
```

Layout Declarations.   The following layout declarations adjust the general layout of exercise sheets. They may as well be moved into an include file.

Declare a header for exercise sheets to display several relevant pieces of data; display points total:

```
13 \exercisestyle{plainheader}
14 \exerciseconfig{composeheaderbelowright}{\getsheetdata{points}}%
```

Redefine the appearance of some counters; sheets should be labelled by capital roman numerals, subproblems by lowercase roman numerals; declare the widest subproblem item to be expected:

```
15 \exerciseconfig{countersheet}{\Roman{sheet}}
16 \exerciseconfig{countersubproblem}{\roman{subproblem})}
17 \exerciseconfig{countersubproblemmax}{vii)}
```

Automatically display an asterisk for all subproblems with bonus points only; remove space to separate items:

```
18 \exerciseconfig{insertsubprobleminfo}{%
19 \switchpoints{}{\addprobleminfo*{%
20   \hspace{-\getexerciseconfig{skipsubprobleminfo}}*}}%
21 {}{}{\getsubproblempoints{}}}
```

Redefine the terms to be used for sheet(s); here, a German version:

```
22 \exerciseconfig{termsheet}{\"Ubungsblatt}
23 \exerciseconfig{termsheets}{\"Ubungsbl\"atter}
```

Display points for problems in the margin; change margin display to use the left margin; use the abbreviated form 'np.':

```
24 \exercisestyle{problempointsat=margin}
25 \reversemarginpar
```

```
26 \exerciseconfig{composepointsmargin}[1]{#1p.}
27 \exerciseconfig{composepointspairmargin}[2]{
28   \ifdim#2pt=0pt#1p.%
29   \else\ifdim#1pt=0pt+#2p.%
30   \else#1+#2p.%
31   \fi\fi}
```

Change the basic font style for all titles to be bold sans-serif:

```
32 \exerciseconfig{styletitle}{\sffamily\bfseries}
```

Add a significant amount of space below problems:

```
33 \exerciseconfig{skipproblembelow}{1.5cm}
```

Display half points as fractions:

```
34 \exercisestyle{fracpoints}
```

Show solutions below each problem (may try alternatives `subproblem` or `sheet`):

```
35 \exercisestyle{solutionbelow=problem}
```

Separate solutions by horizontal lines (extended style):

```
36 \exercisestyle{solutionsep}
```

Add course name to sheet title metadata:

```
37 \exerciseconfig{composemetasheet}[2]{\getexercisedata{course},
38   \exerciseifempty{#2}{\getexerciseconfig{termsheet} #1}{#2}}
```

Set title and author for pdf metadata:

```
39 \exercisesetup{pdfdata=sheet}
40 \exercisedata{title=%
41   {\getexercisedata{course}, \getexercisedata{material}}}
42 \exercisedata{author=%
43   {\getexercisedata{instructor}, \getexercisedata{institution}}}
```

Exercise Series Data.   Set some data on the current series:

```
44 \exercisedata{institution={Katharinen-Volksschule}}
45 \exercisedata{course={Mathematik}}
46 \exercisedata{instructor={J.\ G.\ B\"uttner}}
47 \exercisedata{period={ca.\ 1786}}
48 \exercisedata{material={\"Ubungsaufgaben}}
```

Body.

```
49 \begin{document}
```

Start sheet number 5:

```
50 \begin{sheet}[number=5]
```

Start a problem with a title:

```
51 \begin{problem}[title={Sums},points=99+4]
```

Some introduction to the problem:

```
52 This problem deals with sums and series.
```

A subproblem with a local label:

```
53 \begin{subproblem}[points=2,difficulty=simple,label={\problemtag-simplesum}]
54 Compute the sum
55 \showpoints
56 \begin{equation}
57 1+2+3.
58 \end{equation}
```

Provide a solution for the subproblem (within the subproblem environment):

```
59 \begin{solution}
60 The result is
61 \begin{equation}
62 1+2+3=6.
63 \end{equation}
64 \end{solution}
```

End subproblem:

```
65 \end{subproblem}
```

Another subproblem:

```
66 \begin{subproblem}[points=97+0.5,difficulty=lengthy]
67 Compute the sum
68 \begin{equation}
69 1+2+3+\ldots+98+99+100.
70 \end{equation}
71 Keep calm and calculate!
72 %%That ought to keep him occupied for a while
73 \end{subproblem}
```

Provide a solution for the previous subproblem (layout may differ slightly from declaration within); declare author:

```
74 \begin{solution}[author={C.\ F.\ Gau\ss}]
75 We use the result $1+2+3=6$ from part \ref{\problemtag-simplesum}
76 to jumpstart the calculation. The remaining sums yield
77 \awardpoints*[1 for each remaining sum]{97}
78 \begin{equation}
79 6+4+5+\ldots+99+100=5050.
80 \end{equation}
81 Alternatively the summands can be grouped into pairs as follows:
82 \begin{align}
83 1+100&=101,\\
84 2+99&=101,\\
85 3+98&=101,\\
86 \ldots &\nonumber\\
87 50+51&=101.
88 \end{align}
89 These amount to 50 times the same number 101.
90 Therefore the sum equals
91 \begin{equation}
92 1+2+\ldots+99+100=50\cdot 101=5050.
93 \end{equation}
94 \textit{Ligget se!} \awardpoints{97+0.5}
95 \end{solution}
```

Some text between subproblems:

```
96 You may give the final part a try:
```

Final subproblem; this one is optional:

```
 97 \begin{subproblem}[optional={optional},
 98   difficulty={requires inspiration},points={+3.5}]
 99 Compute the series
100 \showpoints
101 \begin{equation}
102 1+2+3+\ldots
103 \end{equation}
```

Provide a solution:

```
104 \begin{solution}
105 The series is divergent, so the result is $\infty$ \awardpoints{+1}.
106 \par
107 However, after subtracting the divergent part,
108 the result clearly is
109 \begin{equation}
110 \zeta(-1)=-\frac{1}{12}\,,
111 \end{equation}
112 where the zeta-function $\zeta(s)$ is defined by
113 \begin{equation}
114 \zeta(s):=\sum_{k=1}^\infty \frac{1}{k^s}\,.
115 \end{equation}
116 This definition holds only for $s>1$ where the sum is convergent,
117 but one can continue the complex analytic function to $s<0$
118 \awardpoints{+1.5}.
119 \par
120 Another way of understanding the result
121 is to use the indefinite summation formula
122 for arbitrary exponent $s$ in the summand
123 (which also follows from the Euler--MacLaurin formula)
124 \begin{equation}
125 \sum_n n^s
126 = \frac{n^{s+1}}{s+1}
127  -\sum_{j=0}^s  \frac{\zeta(j-s)\,s!}{(s-j)!\,j!}\,n^j
128 = \ldots - \zeta(-s)\,n^0.
129 \end{equation}
130 Curiously, the constant term with $j=0$ is just the desired result
131 but with the wrong sign
132 (in fact, the constant term of an indefinite sum is ambiguous;
133 for the claim we merely set $j=0$
134 in the expression which holds for others values of $j$)
135 \awardpoints{+0.5}.
136 In order to understand the sign,
137 we propose that the above formula describes the regularised result
138 for the sum with limits $+\infty$ and $n$
139 \begin{equation}
140 \sum_{k=+\infty}^n k^s
141 \simeq \frac{n^{s+1}}{s+1}
142  -\sum_{j=0}^s  \frac{\zeta(j-s)\,s!}{(s-j)!\,j!}\,n^j.
143 \end{equation}
144 Then we flip the summation limits of the desired sum
145 to bring it into the above form
146 \awardpoints{+0.5}
147 \begin{equation}
148 \sum_{k=1}^\infty k^s
149 = -\sum_{k=\infty}^0 k^s
150 \simeq \zeta(-s).
```

```
151 \end{equation}
152 \end{solution}
```

End subproblem:

```
153 \end{subproblem}
```

End problem:

```
154 \end{problem}
```

Another problem; this one is untitled:

```
155 \begin{problem}[points=1, difficulty=insane]
156 Show that the equation
157 \begin{equation}
158 a^3+b^3=c^3
159 \end{equation}
160 has no positive integer solutions.
161 \end{problem}
```

A solution can also follow a problem (but the layout may be slightly different, e.g. here the space below the problem will appear before the solution):

```
162 \begin{solution}
163 \normalmarginpar
164 This is beyond the scope of this example.
165 \marginpar{\footnotesize\raggedright does not fit here.\par}
166 \end{solution}
```

End sheet:

```
167 \end{sheet}
```

End of document body:

```
168 \end{document}
```

B Multipart Sample

The second example describes a series of exercise sheets which can be compiled as a collection or as individual sheets. This example describes a versatile setup with several convenient features; most of these features can be adjusted or removed easily as they mostly enhance the setup and do not interact with each other strongly.

B.1 Main File

The main source file is called `exfserm.tex`. It is referenced at several places within the setup, and when changing the name they need to be adjusted accordingly.

childdoc Mechanism. The setup uses the package childdoc to allow compilation of the series as a whole or in parts and with various sets of options:

```
169 \input{childdoc.def}
170 \childdocmain{exfserm}
```

The parameter of \childdocmain must match the main file name exfserm.

Compilation Switches. Define compilation switches and declare their default settings. `\printsol` controls whether solutions should be printed or not; by default solutions are activated for compilation of a part, but not for the complete document. `\draftver` controls whether the final version of the document is to be compiled; concretely this affects the compilations of metapost figures, see below:

```
171 \providecommand{\draftver}{y}
172 \ifchilddoc
173 \providecommand{\printsol}{y}
174 \else
175 \providecommand{\printsol}{n}
176 \fi
```

Preamble. Standard document class:

```
177 \documentclass[12pt]{article}
```

```
178 % \textsf{graphicx} package to display license logo:
179 \RequirePackage{graphicx}
```

hyperref Package. Use the hyperref package. Declare some options, e.g. use bookmarks only for complete document:

```
180 \PassOptionsToPackage{bookmarks=\ifchilddoc false\else true\fi}{hyperref}
181 \PassOptionsToPackage{bookmarksopen=true}{hyperref}
182 \RequirePackage{hyperref}
```

Use the hyperxmp package for inclusion of copyright metadata in the compiled pdf files:

```
183 \IfFileExists{hyperxmp.sty}{\RequirePackage{hyperxmp}}{}
```

exframe Package. Invoke exframe with extended data and styles:

```
184 \RequirePackage[extdata,extstyle]{exframe}
```

Set solutions switch and declare two-sided layout only if no solutions are printed:

```
185 \if\printsol n
186 \exercisesetup{solutions=false}
187 \exercisesetup{twoside=true}
188 \else
189 \exercisesetup{solutions=true}
190 \exercisesetup{twoside=false}
191 \fi
```

Might want to display some metadata (only for partial compile):

```
192 %%\if\printsol n\else\showprobleminfo{author,source,recycle}\fi
```

Set some options. Automatically assign labels to problems. Include sheets and problems in table of contents. Separate solutions by horizontal rules. Count problems, equations and pages by sheet (unless compiling single problem). Collect solutions below each problem. Write pdf metadata for sheet when compiling single sheet.

```
193 \exercisesetup{autolabelproblem}
194 \exercisestyle{contents,solutionsep}
195 \ifchilddocmanual\else
196 \exercisestyle{pagebysheet,problembysheet,equationbysheet,sheetequation}
197 \fi
198 \exercisestyle{solutionbelow={problem}}
199 \ifchilddoc\ifchilddocmanual\else\exercisesetup{pdfdata=sheet}\fi\fi
```

Layout Definitions. Set page dimensions and layout:

```
200 \RequirePackage[a4paper,margin=2.5cm]{geometry}
201 \pagestyle{plain}
```

Remove paragraph indentation:

```
202 \setlength\parindent{0pt}
203 \setlength\parskip{\smallskipamount}
```

Show overfull lines:

```
204 \setlength\overfullrule{5pt}
```

Define turn page over mark; hide when printing solutions:

```
205 \newcommand{\turnover}{\ifsolutions\else\vfill%
206   \hfill{\mathversion{bold}$\longrightarrow$}\newpage\fi}
```

Sheet Banner. Use standard sheet banner; show sheet `editdate` below banner (if declared); when compiling single sheet, display sheet due date instead:

```
207 \exercisestyle{plainheader}
208 \exerciseconfig{composeheaderbelowright}
209  {\sheetdataempty{editdate}{}{version: \getsheetdata{editdate}}}
210 \ifchilddoc\ifsolutions\else
211 \exerciseconfig{composeheaderbelowright}
212  {\sheetdataempty{due}{}{due: \getsheetdata{due}}}
213 \fi\fi
```

Lorem. Define a macro `\lorem` to write out some paragraph of text for the example:

```
214 \def\lorem{Lorem ipsum dolor sit amet, consectetur adipisici elit,
215   sed eiusmod tempor incidunt ut labore et dolore magna aliqua.
216   Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
217   nisi ut aliquid ex ea commodi consequat.
218   Quis aute iure reprehenderit in voluptate velit esse
219   cillum dolore eu fugiat nulla pariatur.
220   Excepteur sint obcaecat cupiditat non proident,
221   sunt in culpa qui officia deserunt mollit anim id est laborum.\par}
```

metapost Setup. Use package mpostinl (if available) to include some metapost figures within the source of the problems:

```
222 \IfFileExists{mpostinl.sty}{\RequirePackage{mpostinl}}{}
223 \ifdefined\mpostsetup
```

Setup mpostinl. Use checksums to invoke metapost only when figures change. Process all figures individually and immediately when in draft mode (avoids a second compilation pass). Number figures within sheet to provide a stable numbering upon insertion/deletion of new figures or partial compilation. Do not warn about unused figures when preparing without solutions (figures for solutions should be declared outside `solution` environments):

```
224 \mpostsetup{checksum}
225 \if\draftver y\mpostsetup{now,nowall}\fi
226 \ifchilddocmanual\else\mpostsetup{numberwithin={sheet}}\fi
227 \ifsolutions\else\mpostsetup{warnunused=false}\fi
```

Global metapost definitions. Define some latex macro to demonstrate usage of latex for typesetting labels. Define some global metapost variables, `paths` is an array of path variables, `xu` serves as a length unit to scale individual figures (use `interim xu:=...;` to set scale for local figure only):

```
228 \mpostsetup{globaldef=true}
229 \begin{mposttex}
230 \def\figure{figure}
231 \end{mposttex}
232 \begin{mpostdef}
233 path paths[];
234 newinternal numeric xu;
235 xu:=1cm;
236 \end{mpostdef}
237 \mpostsetup{globaldef=false}
```

Close optional mpostinl processing:

```
238 \fi
```

Document Data. Set document data for series of problem sheets:

```
239 \exercisedata{course={exframe package samples}}
240 \exercisedata{instructor={N.\ Beisert}}
241 \exercisedata{institution={exframe academy}}
242 \exercisedata{period={spring 2019}}
243 \exercisedata{date={2019}}
```

Assemble some entries from given data:

```
244 \exercisedata{material={\ifsolutions\getexerciseconfig{termsolutions}%
245   \else\getexerciseconfig{termsheets}\fi}}
246 \exerciseconfig{composetitlesheet}[2]{\exerciseifempty{#2}%
247   {\ifsolutions\getexerciseconfig{termsolutions}\else%
248    \getexerciseconfig{termsheet}\fi\ #1}%
249   {\ifsolutions\getexerciseconfig{termsolutions}\fi\ #2}}
250 \exerciseconfig{composemetasheet}[2]{\if\draftver yDRAFT: \fi%
251   \getexercisedata{course},
252   \getexerciseconfig{composetitlesheet}{#1}{#2}}
```

Set general purpose metadata:

```
253 \exercisedata{title={\if\draftver yDRAFT: \fi%
254   \getexercisedata{course}, \getexercisedata{material}}}
255 \exercisedata{author={Niklas Beisert, \getexercisedata{institution}}}
256 \exercisedata{subject={lecture series at \getexercisedata{institution},
257   \getexercisedata{period}}}
```

Redefine some terms:

```
258 \exerciseconfig{termsheet}{sheet}
259 \exerciseconfig{termsheets}{sample sheets}
260 \exerciseconfig{termsolution}{Solution}
261 \exerciseconfig{termsolutions}{solutions}
```

License. It is always useful to specify a copyright line and a license for the document. Define some restrictive default text.

```
262 \def\copyrightmessage
263   {This document as well as its parts is protected by copyright.}
```

```
264 \def\licensemessage
265   {Reproduction of any part of this work in any form
266   without prior written consent of \getexercisedata{institution}
267   is not permissible.}
```

Apply Creative Commons BY-SA license under certain conditions (no solutions, final version):

```
268 \ifsolutions\else\if\draftver y\else
269 \def\licensecctype{by-sa}
270 \def\licenseccname{Attribution-ShareAlike \licenseccver{} International}
271 \fi\fi
```

Write applicable Creative Commons license information. Use Creative Commons logos included in package doclicense:

```
272 \ifdefined\licensecctype
273 \ifdefined\licenseccver\else\def\licenseccver{4.0}\fi
274 \def\licenseurl
275   {https://creativecommons.org/licenses/\licensecctype/\licenseccver}
276 \def\licensemessage{\texorpdfstring
277   {This work is licensed under the Creative Commons ``\licenseccname'' License
278   (CC \MakeUppercase{\licensecctype} \licenseccver).\par
279   \IfFileExists{doclicense.sty}{
280   \begin{center}\includegraphics{doclicense-CC-\licensecctype}\end{center}}{}
281   To view a copy of this license, visit: \url{\licenseurl}}
282   {This work is licensed under the Creative Commons \licenseccname{} License.}}
283 \fi
```

Write copyright and license as pdf metadata (using package hyperxmp if available):

```
284 \ifdefined\xmptilde
285 \hypersetup{pdfcopyright={Copyright \getexercisedata{date}
286   \getexercisedata{author}. \copyrightmessage{} \licensemessage}}
287 \ifdefined\sourceurl\hypersetup{pdflicenseurl={\sourceurl}}\else
288 \ifdefined\licenseurl\hypersetup{pdflicenseurl={\licenseurl}}\fi\fi
289 \hypersetup{keeppdfinfo=true}
290 \hypersetup{pdfsource={}}
291 \XMPLangAlt{en}{pdfcopyright={Copyright \getexercisedata{date}
292   \getexercisedata{author}. \copyrightmessage{} \licensemessage}}
293 \fi
```

Body.   Start document body:

```
294 \begin{document}
```

Single Problem Display.   The following code handles the compilation of individual problems from their own source file. Make sure to leave the conditional before issuing \end{document}:

```
295 \def\tmp{}
296 \ifchilddocmanual
297 \def\tmp{\end{document}}
298 \input{\childdocname}
299 \fi\tmp
```

Frontmatter.   Do not print frontmatter for individual sheets. Define a plain page counter for frontmatter:

```
300 \setcounter{section}{-1}
301 \begingroup\ifchilddoc\else
302 \renewcommand{\thepage}{\arabic{page}}
```

Prepare a title page to display some relevant data:

```
303 \pdfbookmark[1]{Title Page}{title}
304 \thispagestyle{empty}
305 \vspace*{\fill}
306 \begin{center}
307 \begingroup\bfseries\LARGE\getexercisedata{course}\par\endgroup
308 \vspace{0.5cm}
309 \begingroup\large\getexercisedata{material}\par\endgroup
310 \vspace{0.5cm}
311 \begingroup\large\getexercisedata{institution},
312   \getexercisedata{period}\par\endgroup
313 \vspace{2cm}
314 \begingroup\scshape\Large\getexercisedata{instructor}\par\endgroup
315 \end{center}
316 \vspace*{\fill}\vspace*{\fill}
317 \newpage
```

Prepare a copyright and license page using data specified above:

```
318 \phantomsection\pdfbookmark[1]{Copyright}{copyright}
319 \thispagestyle{empty}
320 \vspace*{\fill}\vspace*{\fill}
321 \begin{center}
322 \begin{minipage}{11cm}\raggedright
323 {\copyright} \getexercisedata{date} \getexercisedata{author}
324 \par\medskip
325 \copyrightmessage{}
326 \licensemessage
327 \ifdefined\sourcemessage\par\medskip\sourcemessage\fi
328 \ifdefined\attributionmessage\par\medskip\attributionmessage\fi
329 \end{minipage}\end{center}
330 \vspace*{\fill}\vspace*{\fill}\vspace*{\fill}
331 \newpage
```

Print table of contents:

```
332 \makeatletter\renewcommand\@pnumwidth{2.4em}\makeatother
333 \setcounter{tocdepth}{2}
334 \phantomsection\pdfbookmark[1]{Contents}{contents}
335 {\parskip0pt\tableofcontents}
336 \exercisecleardoublepage\setcounter{page}{1}
```

End of frontmatter:

```
337 \fi\endgroup
```

Include Sheets. Include problem sheets:

```
338 \include{exfser01}
339 \include{exfser02}
340 \include{exfser03}
```

Include sheet to collect unused problems (only process for individual sheets, i.e. itself):

```
341 \def\jobnameunused{exfseraa}
342 \ifx\childdocname\jobnameunused\include{\jobnameunused}\fi
```

End. End of document body:

```
343 \end{document}
```

B.2 Sheet File

Provide some source files `exfser01.tex`, `exfser02.tex`, `exfser03.tex`, `exfseraa.tex` for problem sheets.

childdoc Mechanism. Instruct the package `childdoc` to compile only the present sheet if source is compiled by latex:

```
344 %%\providecommand{\printsol}{n}
345 \input{childdoc.def}
346 \childdocof{exfserm}
```

The parameter of `\childdocof` must match the main file name `exfserm`. Uncommenting the commented line suppressed printing of the solution.

Sheet Environment. Declare a sheet with intended due date:

```
347 \begin{sheet}[due={2019-04-29}]
```

Adjust due date for each sheet. For sheet containing unused problems `exfseraa.tex`, declare a sheet `title={unused problems}` instead of due date.

Problems. Start a problem:

```
348 \begin{problem}[title={Sample A}]
```

Let us declare a figure using `mpostinl` (if available). Denote it by the label *tag*-`fig`, where *tag* is the tag of the problem (in order to avoid potential conflicts with other problems; *tag* is assigned automatically or by specifying the option `tag` for the `problem` environment):

```
349 \ifdefined\mpostuse
350 \begin{mpostfig}[label={\problemtag-fig}]
351 interim xu:=1.5cm;
352 paths[1]:=fullcircle scaled 1xu;
353 fill paths[1] withgreyscale 0.7;
354 draw paths[1] withpen pencircle scaled 1pt;
355 label(btex \figure etex, center(paths[1]));
356 \end{mpostfig}
357 \fi
```

Write a problem body with figure, some subproblems and a solution:

```
358 \lorem
359
360 \begin{subproblem}
361 \lorem
362 \begin{center}
363 \ifdefined\mpostuse\mpostuse{\problemtag-fig}\else figure\fi
364 \end{center}
365 \lorem
366 \end{subproblem}
367
368 \begin{solution}
369 \lorem
```

```
370 \end{solution}
371
372 \begin{subproblem}
373 \lorem
374 \end{subproblem}
375
376 \begin{solution}
377 \lorem
378 \end{solution}
379
380 \lorem
381
382 \begin{subproblem}
383 \lorem
384 \end{subproblem}
385
386 \begin{solution}
387 \lorem
388 \end{solution}
```

End the problem:

```
389 \end{problem}
```

Start new page:

```
390 \turnover
```

Write a second problem to accompany the first one:

```
391 \begin{problem}[title={Sample B}]
```

Problem body without a figure; this time the `solution` environments are included in the `subproblem` environments:

```
392 \lorem
393
394 \begin{subproblem}
395 \lorem
396 \begin{solution}
397 \lorem
398 \end{solution}
399 \end{subproblem}
400
401 \begin{subproblem}
402 \lorem
403 \begin{solution}
404 \lorem
405 \end{solution}
406 \end{subproblem}
```

End the problem:

```
407 \end{problem}
```

End Sheet. End the sheet:

```
408 \end{sheet}
```

B.3 Individual Problem Files

It may be more convenient to define each problem in an individual file, so that a sheet can be composed by including the appropriate problem files. In such a setup, the childdoc mechanism allows to compile each problem individually.

To that end, prepare a file exfserp*nn*.tex containing the problem environment. This file should start with:

```
409 %%\providecommand{\printsol}{n}
410 \input{childdoc.def}
411 \childdocby{exfserm}
```

and end with \endinput. Then compose the problem sheet by including the appropriate set of problem files via \input{exfserp*nn*}.

In the example, a sheet file exfser03.tex includes two problem files exfserpe.tex and exfserpf.tex.

B.4 Make Scripts

The setup allows the compilation in various modes for editing purposes. In order to generate and update a complete set of documents for distribution (all individual sheets and a collection of sheets; with and without solutions), it makes sense to use the software development utility make. The compilation of individual components is simplified by a bash shell script.

Compile Script. The bash shell script exfsermk.sh compiles one part of the collection of problem sheets in a given mode.

Shebang for bash script:

```
412 #!/bin/bash
```

Configure and declare variables with default values. srcmain defines the name of the main source file; srcsec*nn* defines the name of the sheet source file; trglist defines the target file names; trgsol defines the target modes; sheets is a list of allowable sheet identifiers *nn*:

```
413 srcmain="exfserm"
414 srcsec="exfser"
415 trglist=(Problems Solutions)
416 trgsol=(n y)
417 secnum="01 02 03 aa"
```

Display usage:

```
418 if [ -z $1 ]
419 then
420   echo "Usage:
421 $0 number [version]
422    number: number of sheet, 0 for combined document
423    version: 0 for problems, 1 for solutions
424 $0 filename
425    filename: target file to be compiled"
426   exit 1
427 fi
```

Configure and declare variables with default values. num takes the sheet number; ver takes the compile mode;

```
428 num="$1"
429 ver="$2"
430 nl=$'\n'
431 secokay=""
432 make=".pdf"
```

Check if the parameter matches any of the acceptable output file names:

```
433 for v in "${trglist[@]}"
434 do
435   if [[ $num =~ ^$v ]]
436   then
437     ver=$v
438     num=${num#$v}
439     if [[ $num =~ ^.*\.tex$ ]]; then make=".tex"; fi
440     num=${num%%.*}
441   fi
442 done
```

Ensure that `num` is a two-digit number, prepend '0' otherwise:

```
443 if [[ $num =~ ^[0-9]$ ]]; then num="0$num"; fi
444 if [[ $num == "00" ]]; then num=""; fi
```

Check whether `num` is acceptable:

```
445 if [[ -z $num ]]; then secokay="okay"; fi
446 for v in $secnum
447 do
448   if [[ "$num" == "$v" ]]; then secokay="okay"; fi
449 done
```

Otherwise display error message and exit:

```
450 if [[ -z $secokay ]]
451 then
452   echo "error: unknown sheet"
453   exit 1
454 fi
```

Disable newline character for command line tex code:

```
455 if [[ "$make" == ".pdf" ]]; then nl=""; fi
```

Function to compile a component. Set up `childdoc` mechanism according to desired component. Compile two passes, first in `-draftmode`. Suppress messages by `mpost`. Display warning messages in log file:

```
456 function docompile
457 {
458   if [[ -z $num ]]
459   then
460     job="$srcmain"
461     fwd="\\childdocforward{$srcmain}"
462   else
463     job="$srcsec$num"
464     fwd="\\childdocforward[$srcmain]{$srcsec$num}"
465   fi
466   body="\\def\\jobname{$job}$optdef\\input{childdoc.def}$fwd"
467   for pass in first main
468   do
469     par="";
```

```
470    if [[ "$pass" == "first" ]]; then par="-draftmode"; fi
471    drop="This is|entering extended mode|\\write18"
472    drop="$drop|Preloading the plain mem file|mpost\.mp|plain\.mp"
473    pdflatex -shell-escape -interaction=batchmode $par \
474      -jobname "$trg" "$body" | grep -vE "$drop"
475    if [[ "$pass" != "main" ]]; then continue; fi
476    if ! (grep -E -q "may have changed|rerunfilecheck Warning" "$trg.log"); then break; fi
477  done
478  grep -E "^! |Warning|Error|Undefined|Overfull|Underfull" "$trg.log"
479 }
```

Function to generate a **childdoc** compile file with specific options:

```
480 function writesource
481 {
482   if [[ -z $num ]]
483   then
484     fwd="\\childdocforward{$srcmain}"
485   else
486     fwd="\\childdocforwardprefix[$srcmain]{$target}{$srcsec}"
487   fi
488   body="$optdef\\input{childdoc.def}$nl$fwd"
489   echo "$body" > $trg.tex
490 }
```

Translate versions to parameter values, configure variables and select appropriate function for intended task:

```
491 for i in "${!trglist[@]}"
492 do
493   if [[ -z $ver || "$ver" == "${trglist[$i]}" || $ver = $i ]]
494   then
495     target="${trglist[$i]}"
496     sol="${trgsol[$i]}"
497     trg="$target$num"
498     optdef="\\def\\draftver{n}$nl\\def\\printsol{$sol}$nl"
499     if [[ "$make" == ".pdf" ]]; then docompile; else writesource; fi
500   fi
501 done
```

Finish with blank line:

```
502 echo
```

Makefile. Define a make file `exfsermk.mak` for the project. The compilation is then started by `make -f exfsermk.mak` …. More conveniently, in a single-project setup within the directory, this file would be called `Makefile` in which case it suffices to just run `make` ….

Configuration definitions:

```
503 SRCMAIN = exfserm
504 SRCSEC = exfser
505 SRCPRB = exfserp
506 SCRIPT = exfsermk.sh
507 MAKEFILE = exfsermk.mak
508 TRGLIST = Problems Solutions
509 SECNUM = 01 02 03 aa
510 PREREQS = $(SRCMAIN).tex
511
```

```
512 SRCSECFILES = $(SECNUM:%=$(SRCSEC)%.tex)
513 TRGMAINFILES = $(foreach trg,$(TRGLIST),$(trg).pdf)
514 TRGSECFILES = $(foreach trg,$(TRGLIST),$(trg).pdf $(SECNUM:%=$(trg)%.pdf))
515 GENFILES = $(foreach trg,$(TRGLIST),$(trg).tex $(SECNUM:%=$(trg)%.tex))
516 BAKFILES = $(PREREQS) $(SRCSECFILES) $(GENFILES) $(MAKEFILE) $(SCRIPT) $(SRCPRB)*
```

Define some abstract targets; `default` is the default target when no parameters are given:

```
517 default: sheets ;
518 main: $(TRGMAINFILES) ;
519 sheets: $(TRGSECFILES) ;
520 sheet%: $(foreach trg,$(TRGLIST),$(trg)%.pdf) ;
521 all: main sheets ;
522 sources: $(GENFILES) ;
```

Compile particular files via `exfsermk.sh` bash script. Note that command lines have to start with a tab character (represented by 8 spaces here):

```
523 $(TRGMAINFILES): $(SRCSECFILES) $(PREREQS)
524         bash ./$(SCRIPT) $@
525 $(word 1,$(TRGLIST))%.pdf: $(SRCSEC)%.tex $(PREREQS)
526         bash ./$(SCRIPT) $@
527 $(word 2,$(TRGLIST))%.pdf: $(SRCSEC)%.tex $(PREREQS)
528         bash ./$(SCRIPT) $@
529 $(GENFILES):
530         bash ./$(SCRIPT) $@
```

Touch main file for recompile:

```
531 touch:
532         touch $(SRCMAIN).tex
```

Define `clean` target to remove all intermediate compilation files:

```
533 clean:
534         rm -f $(foreach ext,.aux .log,$(SECNUM:%=$(SRCSEC)%$(ext)))
535         rm -f $(foreach trg,$(TRGLIST),$(SECNUM:%=$(trg)%.log) $(trg).log)
536         rm -f $(foreach ext,.aux .log .out .toc,$(SRCMAIN)$(ext))
537         rm -f $(foreach ext,.mp .mpx -*.mps -tmp.log -tmp.mp -tmp.mpx,$(SRCMAIN)$(ext))
538         rm -f mpxerr.tex mpxerr.log mpxerr.dvi texput.log
539         rm -f $(patsubst %,$(SRCPRB)*%,.aux .log .mp .mpx -*.mps)
540         rm -f $(patsubst %,$(SRCPRB)*%,-tmp.log -tmp.mp -tmp.mpx)
```

Define `clean-bak` target to remove all backup files ending in '~' or '.bak':

```
541 clean-bak:
542         rm -f $(BAKFILES:%=%~) $(BAKFILES:%=%.bak)
```

Define `clean-all` target to remove all generated files for a clean source directory:

```
543 clean-all:  clean
544         rm -f $(TRGSECFILES) $(TRGMAINFILES) $(GENFILES)
545         rm -f $(SECNUM:%=$(SRCSEC)%.pdf) $(SRCMAIN).pdf
546         rm -f $(SRCPRB)*.pdf
```

C   Implementation

This section describes the package `exframe.sty`.

C.1 General Definitions

Required Packages. The package loads the package verbatim and xkeyval if not yet present. verbatim is used for solution environment reading and xkeyval is used for extended options processing:

```
547 \RequirePackage{verbatim}
548 \RequirePackage{xkeyval}
```

General Definitions.

\exf@empty   Define an empty macro for comparison by \ifx:

```
549 \def\exf@empty{}
```

\exf@tmpdim   Define a length for temporary storage:

```
550 \newlength\exf@tmpdim
```

\exf@exptwo   A macro to conveniently expand the third token in line:

```
551 \def\exf@exptwo#1{\expandafter#1\expandafter}
```

\exf@exparg   A macro to conveniently expand the first token of an argument following arbitrary code:

```
552 \long\def\exf@expswitch#1#2{#2{#1}}
553 \long\def\exf@exparg#1#2{\exf@exptwo\exf@expswitch{#2}{#1}}
```

\exf@csdo
\exf@csdotwo

Some macros to conveniently expand \csname arguments before expanding the macro:

```
554 \def\exf@csdo#1#2{\expandafter#1\csname#2\endcsname}
555 \def\exf@csdotwo#1#2#3{\exf@exptwo#1#2\csname#3\endcsname}
```

\exf@append@def
\exf@prepend@def

Add definitions to macros (after or before original content):

```
556 \long\def\exf@append@def#1#2{\exf@exptwo\def#1\expandafter{#1#2}}
557 \long\def\exf@prepend@switch#1#2#3{#2{#3#1}}
558 \long\def\exf@prepend@def#1#2{\exf@exptwo\exf@prepend@switch{#1}{\def#1}{#2}}
```

\exf@expsetkeys   A version of \setkeys from xkeyval which expands first:

```
559 \newcommand{\exf@expsetkeys}[2]{\edef\exf@tmp{#2}%
560   \exf@exparg{\setkeys{#1}}{\exf@tmp}}
```

\exf@isif   Execute #3 if content of macro #1 equals #2:

```
561 \newcommand{\exf@ifis}[3]%
562   {\def\exf@tmp{#2}\ifx#1\exf@tmp#3\fi}
```

\exf@href   Display text with hyperreference passed by macro #1 (in case hyperref is loaded and the reference is defined and not empty):

```
563 \newcommand{\exf@href}[2]{%
564   \ifdefined#1\ifx#1\exf@empty#2\else%
565   \ifdefined\hyperlink\protect\hyperlink{#1}{#2}\else#2\fi\fi\else#2\fi}
```

\exf@text
\exf@ensuretext

Two macros to display text in math mode. exf@text is a wrapper for \text of amstext in case the latter package is loaded. exf@ensuretext makes sure the text is set in text mode or within an \mbox in math math:

```
566 \newcommand{\exf@text}[1]{\ifdefined\text\text{#1}\else#1\fi}
567 \newcommand{\exf@ensuretext}[1]{\ifmmode\mbox{#1}\else#1\fi}
```

exf@addcontentsline   Add a line to the table of contents unless macro in argument `#1` is empty:

```
568 \newcommand{\exf@addcontentsline}[2]{%
569   \ifx#1\exf@empty\else\addcontentsline{toc}{#1}{#2}\fi}
```

C.2   Package Setup

Initialisation Options.

exframe.sty   Some setup options are available while loading the package only.

Configure names of main environments and counters:

```
570 \def\exf@problemname{problem}
571 \def\exf@subproblemname{sub\exf@problemname}
572 \def\exf@solutionname{solution}
573 \def\exf@sheetname{sheet}
574 \def\exf@problemcounter{problem}
575 \def\exf@subproblemcounter{sub\exf@problemcounter}
576 \def\exf@solutioncounter{solution}
577 \def\exf@sheetcounter{sheet}
578 \define@key{exframe.sty}{problemenv}{\def\exf@problemname{#1}}
579 \define@key{exframe.sty}{subproblemenv}{\def\exf@subproblemname{#1}}
580 \define@key{exframe.sty}{solutionenv}{\def\exf@solutionname{#1}}
581 \define@key{exframe.sty}{sheetenv}{\def\exf@sheetname{#1}}
582 \define@key{exframe.sty}{problemcounter}{\def\exf@problemcounter{#1}}
583 \define@key{exframe.sty}{subproblemcounter}{\def\exf@subproblemcounter{#1}}
584 \define@key{exframe.sty}{solutioncounter}{\def\exf@solutioncounter{#1}}
585 \define@key{exframe.sty}{sheetcounter}{\def\exf@sheetcounter{#1}}
```

Whether to provide some extended configuration options (available while loading only):

```
586 \define@boolkey{exframe.sty}[exf@]{extdata}[true]{}
587 \define@boolkey{exframe.sty}[exf@]{extstyle}[true]{}
```

Setup Options.

exf@setup   All remaining setup options are available also when the package is already loaded.

Main switch for solutions:

```
588 \define@boolkey{exf@setup}[]{solutions}[true]{}
```

Switch for writing pdf metadata:

```
589 \define@choicekey{exf@setup}{pdfdata}%
590   {auto,manual,sheet,off}[auto]{\def\exf@metadata{#1}}
591 \def\exf@metadata{auto}
```

Write line number indicators to output file:

```
592 \define@boolkey{exf@setup}[exf@]{lineno}[true]{}
```

Prepare two-sided sheets:

```
593 \define@boolkey{exf@setup}[exf@]{twoside}[true]{}
```

Generate hyperreferences from solutions to corresponding problems:

```
594 \define@boolkey{exf@setup}[exf@]{solutionhref}[true]{}
595 \exf@solutionhreftrue
```

Automatically generate labels for sheets and problems:

```
596 \define@boolkey{exf@setup}[exf@]{autolabelsheet}[true]{}
597 \define@boolkey{exf@setup}[exf@]{autolabelproblem}[true]{}
```

Write warning message to document for better detection of inconsistencies:

```
598 \define@boolkey{exf@setup}[exf@]{warntext}[true]{}
```

Activate buffering of solutions:

```
599 \define@boolkey{exf@setup}[exf@]{solutionbuf}[true]{}
600 \exf@solutionbuftrue
```

Activate buffering of problems:

```
601 \define@boolkey{exf@setup}[exf@]{problembuf}[true]{}
```

Processing.   Process global options while loading package:

```
602 \ProcessOptionsX<exframe.sty,exf@setup>
```

\exercisesetup   Configure package when package is already loaded:

```
603 \newcommand{\exercisesetup}[1]{\exf@expsetkeys{exf@setup}{#1}}
```

Solutions Only Processing.

onlysolutions   Process block only in solutions mode:

```
604 \newenvironment{onlysolutions}%
605  {\ifsolutions\else%
606    \let\endonlysolutions\endcomment%
607    \expandafter\comment\fi}%
608  {}
```

C.3   Configuration

This section defines and describes the various configuration options provided by the package. It also serves as a manual, and most code can be recycled and adjusted for individual configurations:

Definitions.

\exerciseconfig   Set a configuration macro; store definition in `exf@config@#2`; use \newcommand for macros with arguments, but (non-long) \def for plain definitions:

```
609 \newcommand{\exerciseconfig}[1]{%
610  \@ifnextchar[{\exf@configopt{#1}}{\exf@confignoopt{#1}}}
611 \long\def\exf@configopt#1[#2]#3{%
612  \exf@csdo\def{exf@config@#1}{}%
613  \exf@csdo\renewcommand{exf@config@#1}[#2]{#3}}%
614 \long\def\exf@confignoopt#1#2{\exf@csdo\def{exf@config@#1}{#2}}
```

Append to a (parameterless) configuration macro:

```
615 \newcommand{\exerciseconfigappend}[2]{%
616   \exf@csdo\exf@append@def{exf@config@#1}{#2}}
617 \newcommand{\exerciseconfigprepend}[2]{%
618   \exf@csdo\exf@prepend@def{exf@config@#1}{#2}}
```

\getexerciseconfig Get configuration macro:

```
619 \newcommand{\getexerciseconfig}[1]{\csname exf@config@#1\endcsname}
```

exerciseconfigempty Test whether configuration macro #1 is empty; execute #2 if empty, otherwise execute #3:

```
620 \newcommand{\exerciseconfigempty}[3]{\exf@csdo\ifx{exf@config@#1}\exf@empty%
621   #2\else#3\fi}
```

\exerciseifemtpy Code to test whether #1 (expanded) is empty; execute #2 if empty, otherwise execute #3:
\exerciseifnotempty

```
622 \long\def\exerciseifempty#1#2#3{\if&#1&#2\else#3\fi}
623 \long\def\exerciseifnotempty#1#2{\if&#1&\else#2\fi}
```

Terms.

term... Terms for sheet, problem, solution and points (for adjustment or internationalisation):

```
624 \exerciseconfig{termsheet}{Sheet}
625 \exerciseconfig{termsheets}{Sheets}
626 \exerciseconfig{termproblem}{Problem}
627 \exerciseconfig{termproblems}{Problems}
628 \exerciseconfig{termsolution}{Solution}
629 \exerciseconfig{termsolutions}{Solutions}
630 \exerciseconfig{termpoint}{point}
631 \exerciseconfig{termpoints}{points}
```

Formatting Styles.

style... Formatting styles to be applied for various parts of text. Different styles will be applied in sequence from more general to more specific.

Basic style for all exercise text:

```
632 \exerciseconfig{styletext}{\normalsize\normalfont}
```

Style for problems:

```
633 \exerciseconfig{styletextproblem}{}
```

Style for solutions:

```
634 \exerciseconfig{styletextsolution}{\footnotesize}
```

Basic style for titles:

```
635 \exerciseconfig{styletitle}{\bfseries}
```

Style for problem titles:

```
636 \exerciseconfig{styletitleproblem}{\large}
```

Style for subproblem titles:

```
637 \exerciseconfig{styletitlesubproblem}{}
```

Style for solution titles:

638 \exerciseconfig{styletitlesolution}{}

Style for problem section title in solution block:

639 \exerciseconfig{styletitlesolutionsproblem}{\small}

Style for solution block title:

640 \exerciseconfig{styletitlesolutions}{\normalsize}

Style for problem block title:

641 \exerciseconfig{styletitleproblems}{\Large}

Spacing.

skip...   Spaces related to various elements. Vertical space is typically combined with space declared elsewhere using \addvspace.

Space above problem environment:

642 \exerciseconfig{skipproblemabove}{3.25ex plus 1ex minus 1.5ex}

Space below problem environment:

643 \exerciseconfig{skipproblembelow}{3pt plus 1pt minus 1pt}

Space below or after problem title; positive numbers generate vertical space (problem body is started in new paragraph), negative numbers generate horizontal space (problem body continues on opening line):

644 \exerciseconfig{skipproblemtitle}{3pt plus 1pt minus 1pt}

Horizontal space between items in the problem opening line:

645 \exerciseconfig{skipprobleminfo}{0.5em}

Space for problem item and indentation; 0pt means no indentation and direct display of title; positive numbers define an absolute amount; -1pt (or any negative number) computes the amount of indentation from the width of (standard) item plus separator:

646 \exerciseconfig{skipproblemitem}{0pt}

Spaces related to subproblem environment; analogous to spaces related to problem environment, see above:

647 \exerciseconfig{skipsubproblemabove}{1.5ex plus 0.5ex minus 1ex}
648 \exerciseconfig{skipsubproblembelow}{1.5ex plus 0.5ex minus 1ex}
649 \exerciseconfig{skipsubproblemtitle}{-1em}
650 \exerciseconfig{skipsubprobleminfo}{0.25em}
651 \exerciseconfig{skipsubproblemitem}{-1pt}

Spaces related to solution environment; analogous to spaces related to problem environment,

652 \exerciseconfig{skipsolutionabove}{0ex}
653 \exerciseconfig{skipsolutionbelow}{1.5ex plus 0.5ex minus 1ex}
654 \exerciseconfig{skipsolutiontitle}{-0.5em}
655 \exerciseconfig{skipsolutioninfo}{0.25em}

skipsolutionitem and skipsolutionitemsub are analogous to skipproblemitem described above; they apply to solutions corresponding to problems and subproblems, respectively:

```
656 \exerciseconfig{skipsolutionitem}{0pt}
657 \exerciseconfig{skipsolutionitemsub}{0pt}
```

Spaces related to solution blocks; space above and below a solution block:

```
658 \exerciseconfig{skipsolutionsabove}{1.5ex plus 0.5ex minus 1ex}
659 \exerciseconfig{skipsolutionsbelow}{1.5ex plus 0.5ex minus 1ex}
```

Space above problem titles in a solution block:

```
660 \exerciseconfig{skipsolutionsproblemabove}{1.0ex plus 0ex minus 0.5ex}
```

Space following problem titles in a solution block (with legacy definition):

```
661 \exerciseconfig{skipsolutionsproblemtitle}{1.0ex plus 0ex minus 0.5ex}
662 \exerciseconfig{skipsolutionsproblem}{\exf@config@skipsolutionsproblemtitle}
```

Space following title of a solution block:

```
663 \exerciseconfig{skipsolutionstitle}{1.0ex plus 0ex minus 0.5ex}
```

Spaces related to problem blocks; space above and below a problem block:

```
664 \exerciseconfig{skipproblemsabove}{1.5ex plus 0.5ex minus 1ex}
665 \exerciseconfig{skipproblemsbelow}{1.5ex plus 0.5ex minus 1ex}
```

Space following title of a problem block:

```
666 \exerciseconfig{skipproblemstitle}{1.0ex plus 0ex minus 0.5ex}
```

Hook Code.

insert...   Code to process data and to insert text at various points of processing.

Code to generate the title for a sheet; minimalistic default to display the sheet title:

```
667 \exerciseconfig{insertsheettitle}{\centerline{\getsheetdata{title}}}
```

Code to clear the page at the start and at the end of a new sheet:

```
668 \exerciseconfig{insertsheetclearpage}{\exercisecleardoublepage}
```

Code to insert before a sheet is displayed:

```
669 \exerciseconfig{insertsheetbefore}{}
```

Code to insert after a sheet is displayed:

```
670 \exerciseconfig{insertsheetafter}{}
```

Code to insert before a solution block is displayed:

```
671 \exerciseconfig{insertsolutionsbefore}{}
```

Code to insert after a solution block is displayed:

```
672 \exerciseconfig{insertsolutionsafter}{}
```

Code to insert before a problem block is displayed:

```
673 \exerciseconfig{insertproblemsbefore}{}
```

Code to insert after a problem block is displayed:

```
674 \exerciseconfig{insertproblemsafter}{}
```

Code to insert before a problem is displayed:

```
675 \exerciseconfig{insertproblembefore}{}
```

Code to insert after a problem is displayed:

```
676 \exerciseconfig{insertproblemafter}{}
```

Code to insert between a problem and its solution (if style `solutionbelow` is `problem`):

```
677 \exerciseconfig{insertproblemsolution}{}
```

Code to insert text into the problem opening line by means of `\addprobleminfo`:

```
678 \exerciseconfig{insertprobleminfo}{}
```

Code to determine the display of problems:

```
679 \exerciseconfig{insertproblemselect}[1]{}
```

Analogous definitions for subproblems:

```
680 \exerciseconfig{insertsubproblembefore}{}
681 \exerciseconfig{insertsubproblemafter}{}
682 \exerciseconfig{insertsubprobleminfo}{}
683 \exerciseconfig{insertsubproblemsolution}{}
```

Analogous definitions for solutions:

```
684 \exerciseconfig{insertsolutionbefore}{}
685 \exerciseconfig{insertsolutionafter}{}
686 \exerciseconfig{insertsolutioninfo}{}
```

Text Composition for Environments.

compose... Macros to generate text for various situations. Preferably the output is plain text without formatting, but in some situations it may be required to address formatting in these macros.

Default separator for items:

```
687 \exerciseconfig{composeitemsep}{\ }
```

Compose sheet title; arguments are sheet number and raw title (empty if not specified); default is "Sheet #1" or given title "#2":

```
688 \exerciseconfig{composetitlesheet}[2]%
689   {\exerciseifempty{#2}{\getexerciseconfig{termsheet} #1}{#2}}
```

Compose sheet title for pdf metadata; arguments are sheet number and raw title:

```
690 \exerciseconfig{composemetasheet}[2]%
691   {\getexerciseconfig{composetitlesheet}{#1}{#2}}
```

Compose sheet title for table of contents; arguments are sheet number and raw title:

```
692 \exerciseconfig{composetocsheet}[2]%
693   {\exerciseifempty{#2}{\getexerciseconfig{termsheet} #1}{#1. #2}}
```

Compose problem item; argument is problem number:

```
694 \exerciseconfig{composeitemproblem}[1]{#1.}
```

Problem item separator:

```
695 \exerciseconfig{composeitemproblemsep}%
696   {\getexerciseconfig{composeitemsep}}
```

Compose problem title; arguments are problem number (empty if item is split off) and raw title (empty if not specified); default is "Problem #1." or "#1. #2":

```
697 \exerciseconfig{composetitleproblem}[2]{\exerciseifempty{#1}%
698    {\exerciseifempty{#2}{}{#2}}%
699    {\exerciseifempty{#2}{\getexerciseconfig{termproblem}\ %
700      \getexerciseconfig{composeitemproblem}{#1}}%
701     {\getexerciseconfig{composeitemproblem}{#1} #2}}}
```

Compose problem title for table of contents; arguments are problem number and raw title:

```
702 \exerciseconfig{composetocproblem}[2]%
703    {\exerciseifempty{#2}{\getexerciseconfig{termproblem} #1}{#1. #2}}
```

Compose subproblem item; argument is subproblem number:

```
704 \exerciseconfig{composeitemsubproblem}[1]{#1}
```

Subproblem item separator:

```
705 \exerciseconfig{composeitemsubproblemsep}%
706    {\getexerciseconfig{composeitemsep}}
```

Compose subproblem title; argument is subproblem number:

```
707 \exerciseconfig{composetitlesubproblem}[1]{#1}
```

Compose solution item; arguments are problem and subproblem number:

```
708 \exerciseconfig{composeitemsolution}[2]{#1.}
709 \exerciseconfig{composeitemsolutionsub}[2]{#2}
```

Solution item separator:

```
710 \exerciseconfig{composeitemsolutionsep}%
711    {\getexerciseconfig{composeitemsep}}
```

Compose title for single solution; arguments are corresponding problem and subproblem number:

```
712 \exerciseconfig{composetitlesolutionsingle}[2]%
713    {\getexerciseconfig{termsolution}:}
```

Compose title for one out of several solutions; arguments are corresponding problem and subproblem number:

```
714 \exerciseconfig{composetitlesolutionmulti}[2]{#2}
```

Compose table of contents line for solution; arguments are problem number and raw title:

```
715 \exerciseconfig{composetocsolution}[2]%
716    {\getexerciseconfig{composetocproblem}{#1}{#2}}
```

Compose title for solution block:

```
717 \exerciseconfig{composetitlesolutions}%
718    {\getexerciseconfig{termsolutions}}
```

Compose title for problem block:

```
719 \exerciseconfig{composetitleproblems}%
720    {\getexerciseconfig{termproblems}}
```

Compose table of contents line for solution block:

```
721 \exerciseconfig{composetocsolutions}%
722    {\getexerciseconfig{composetitlesolutions}}
```

Compose table of contents line for problem block:

```
723 \exerciseconfig{composetocproblems}%
724   {\getexerciseconfig{composetitleproblems}}
```

Compose sectional title for solution following a single problem; arguments are problem number and raw title:

```
725 \exerciseconfig{composetitlesolutionsproblemsingle}[2]%
726   {\getexerciseconfig{termsolution}}
```

Compose sectional title for solution of one problem within a solution block; arguments are problem number and raw title:

```
727 \exerciseconfig{composetitlesolutionsproblemmulti}[2]%
728   {\exerciseifempty{#2}{\getexerciseconfig{termproblem} #1}{#1. #2}}
```

Compose label:

```
729 \exerciseconfig{composeitemsolutionlabel}[2]{#1#2}
```

Points.   Compose number of points:

```
730 \exerciseconfig{composepointsnum}[1]{#1}
```

Compose number of points followed by 'points'; use singular 'point' for 1:

```
731 \exerciseconfig{composepoints}[1]{\getexerciseconfig{composepointsnum}{#1}~%
732   \ifdim #1pt=1pt\getexerciseconfig{termpoint}%
733   \else\getexerciseconfig{termpoints}\fi}
```

Compose points declaration for use in opening line:

```
734 \exerciseconfig{composepointsstart}[1]{(\getexerciseconfig{composepoints}{#1})}
```

Compose points declaration for use in margin:

```
735 \exerciseconfig{composepointsmargin}[1]{\getexerciseconfig{composepoints}{#1}}
```

Compose points declaration for use in text:

```
736 \exerciseconfig{composepointsbody}[1]{(\getexerciseconfig{composepoints}{#1})}
```

Compose points declaration for use in sheet data:

```
737 \exerciseconfig{composepointssheet}[1]{%
738   \exerciseifnotempty{#1}{\getexerciseconfig{composepoints}{#1}}}
```

Compose points declaration for solution with comment:

```
739 \exerciseconfig{composepointsaward}[2]%
740   {(\getexerciseconfig{composepoints}{#1}\exerciseifnotempty{#2}{; #2})}
```

Compose alternative points for solution declaration with comment:

```
741 \exerciseconfig{composepointsawardalt}[2]%
742   {(\getexerciseconfig{composepoints}{#1}*\exerciseifnotempty{#2}{; #2})}
```

Compose pairs of points; omit 0 components:

```
743 \exerciseconfig{composepointspair}[2]{%
744   \ifdim#2pt=0pt%
745     \getexerciseconfig{composepoints}{#1}%
746   \else\ifdim#1pt=0pt%
747     +\getexerciseconfig{composepoints}{#2}%
```

```
748    \else%
749    \getexerciseconfig{composepointsnum}{#1}+%
750    \getexerciseconfig{composepointsnum}{#2}~%
751    \getexerciseconfig{termpoints}%
752  \fi\fi}
```

Compose pairs of points for designated use; recycle plain definition if no bonus points given:

```
753 \exerciseconfig{composepointspairbody}[2]{%
754    \ifdim#2pt=0pt\getexerciseconfig{composepointsbody}{#1}\else%
755    (\getexerciseconfig{composepointspair}{#1}{#2})\fi}
756 \exerciseconfig{composepointspairstart}[2]{%
757    \ifdim#2pt=0pt\getexerciseconfig{composepointsstart}{#1}\else%
758    (\getexerciseconfig{composepointspair}{#1}{#2})\fi}
759 \exerciseconfig{composepointspairmargin}[2]{%
760    \ifdim#2pt=0pt\getexerciseconfig{composepointsmargin}{#1}\else%
761    \getexerciseconfig{composepointspair}{#1}{#2}\fi}
762 \exerciseconfig{composepointspairsheet}[2]{%
763    \ifdim#2pt=0pt\getexerciseconfig{composepointssheet}{#1}\else%
764    \getexerciseconfig{composepointspair}{#1}{#2}\fi}
765 \exerciseconfig{composepointspairaward}[3]{%
766    \ifdim#2pt=0pt\getexerciseconfig{composepointsaward}{#1}{#3}\else%
767    (\getexerciseconfig{composepointspair}{#1}{#2}%
768    \exerciseifnotempty{#3}{; #3})\fi}
769 \exerciseconfig{composepointspairawardalt}[3]{%
770    \ifdim#2pt=0pt\getexerciseconfig{composepointsawardalt}{#1}{#3}\else%
771    (\getexerciseconfig{composepointspair}{#1}{#2}*%
772    \exerciseifnotempty{#3}{; #3})\fi}
```

Compose pairs of points for designated situations:

```
773 \exerciseconfig{composepointspairbodyproblem}[2]{%
774    \getexerciseconfig{composepointspairbody}{#1}{#2}}
775 \exerciseconfig{composepointspairbodysubproblem}[2]{%
776    \getexerciseconfig{composepointspairbody}{#1}{#2}}
777 \exerciseconfig{composepointspairbodysolution}[2]{%
778    \getexerciseconfig{composepointspairbody}{#1}{#2}}
779 \exerciseconfig{composepointspairstartproblem}[2]{%
780    \getexerciseconfig{composepointspairstart}{#1}{#2}}
781 \exerciseconfig{composepointspairstartsubproblem}[2]{%
782    \getexerciseconfig{composepointspairstart}{#1}{#2}}
783 \exerciseconfig{composepointspairstartsolution}[2]{%
784    \getexerciseconfig{composepointspairstart}{#1}{#2}}
```

Display points in the margin:

```
785 \exerciseconfig{insertpointsmargin}[1]{\marginpar{\footnotesize #1}}
```

Display warning about points mismatch:

```
786 \exerciseconfig{insertwarnpoints}[3]
787    {\textbf{points mismatch for #1 (#2 determined vs.\ #3 given)}}
```

Display warning about points changed:

```
788 \exerciseconfig{insertwarnpointsrerun}[1]
789    {\textbf{points changed for #1 (please recompile)}}
```

Counters. Define counter display via configuration interface:

```
790 \exerciseconfig{countersheet}{\arabic{\exf@sheetcounter}}
```

```
791 \exerciseconfig{counterproblem}{\arabic{\exf@problemcounter}}
792 \exerciseconfig{counterproblemmax}{10}
793 \exerciseconfig{countersubproblem}{\alph{\exf@subproblemcounter})}
794 \exerciseconfig{countersubproblemmax}{m)}
795 \exerciseconfig{countersheetequation}{\arabic{equation}}
796 \exerciseconfig{counterproblemequation}{P\arabic{equation}}
797 \exerciseconfig{countersolutionequation}{S\arabic{equation}}
```

Further Definitions.

labelsheet   Templates for automatic generation of labels from tags:
labelproblem
```
798 \exerciseconfig{labelsheet}[1]{sheet:#1}
799 \exerciseconfig{labelproblem}[1]{prob:#1}
```

toclevel...   Table of contents levels for sheets, problems, solutions of problems and solution blocks;
empty means no writing to table of contents:
```
800 \exerciseconfig{toclevelsheet}{}
801 \exerciseconfig{toclevelproblem}{}
802 \exerciseconfig{toclevelproblems}{}
803 \exerciseconfig{toclevelsolution}{}
804 \exerciseconfig{toclevelsolutions}{}
```

extsolutions   Filename extension for solution and problem blocks:
extproblems
```
805 \exerciseconfig{extsolutions}{.sol}
806 \exerciseconfig{extproblems}{.prb}
```

C.4   Styles

Styles are meant as a way to adjust several configuration options at the same time to achieve a
consistent layout in some regard. Useful examples can be found among the extended exercise
styles. They can serve a starting point for further custom styles.

Exercise Styles Code.

defexercisestylearg   Define a style with an argument:
```
807 \newcommand{\defexercisestylearg}[3][]{%
808   \def\exf@tmp{#1}\ifx\exf@tmp\exf@empty%
809     \define@key{exf@style}{#2}{#3}\else%
810     \define@key{exf@style}{#2}[#1]{#3}\fi}
```

\defexercisestyle   Define a style with a boolean argument; execute code onlf if true:
```
811 \newcommand{\defexercisestyle}[2]{%
812   \exf@csdotwo\long\def{exf@style@code@#1}{#2}%
813   \exf@exparg{\define@boolkey{exf@style}[exf@style@]{#1}[true]}%
814     {\csname ifexf@style@#1\endcsname\csname exf@style@code@#1\endcsname\fi}}
```

\exercisestyle   Process styles:
```
815 \newcommand{\exercisestyle}[1]{\exf@expsetkeys{exf@style}{#1}}
```

Default Exercise Styles.

problemmanual   Delay display of problems:

```
816 \define@boolkey{exf@style}[exf@]{problemmanual}[true]{}
817 \exf@problemmanualfalse
```

solutionbelow   Choose location for solutions:

```
818 \def\exf@solutionbelow{subproblem}
819 \define@choicekey{exf@style}{solutionbelow}%
820   {here,subproblem,subproblem*,problem,problem*,sheet,manual}%
821   {\ifexf@solfile@open\else\gdef\exf@solutionbelow{#1}\fi}
```

sheetequation
problemequation
solutionequation   Use separate equation counters for sheets, problems and solutions:

```
822 \defexercisestyle{sheetequation}{}
823 \defexercisestyle{problemequation}{}
824 \defexercisestyle{solutionequation}{}
825 \exf@style@solutionequationtrue
```

problempointsat
subproblempointsat
solutionpointsat   Choose where points of (sub)problems and solutions are displayed:

```
826 \def\exf@pointsat{start}
827 \define@choicekey{exf@style}{problempointsat}%
828   {start,start*,margin,end,manual,off}{\def\exf@pointsat{#1}}
829 \define@choicekey{exf@style}{pointsat}%
830   {start,start*,margin,end,manual,off}{\def\exf@pointsat{#1}}
831 \def\exf@subpointsat{end}
832 \define@choicekey{exf@style}{subproblempointsat}%
833   {start,start*,margin,end,manual,off}{\def\exf@subpointsat{#1}}
834 \define@choicekey{exf@style}{subpointsat}%
835   {start,start*,margin,end,manual,off}{\def\exf@subpointsat{#1}}
836 \def\exf@solpointsat{off}
837 \define@choicekey{exf@style}{solutionpointsat}%
838   {start,start*,margin,end,manual,off}{\def\exf@solpointsat{#1}}
839 \define@choicekey{exf@style}{solpointsat}%
840   {start,start*,margin,end,manual,off}{\def\exf@solpointsat{#1}}
```

problemby
equationby   Declare problems or equations as subcounter of other counter:

```
841 \defexercisestylearg{problemby}{\exf@numberproblemwithin{#1}}
842 \defexercisestylearg{equationby}{\exf@numberequationwithin{#1}}
```

pagebysheet
problembysheet
equationbysheet   Number pages, problems or equations by sheet:

```
843 \defexercisestyle{pagebysheet}{%
844   \def\thepage{\csname the\exf@sheetcounter\endcsname.\arabic{page}}%
845   \def\theHpage{\csname theH\exf@sheetcounter\endcsname.\arabic{page}}%
846   \exerciseconfigappend{insertsheetbefore}{\setcounter{page}{1}}}
847 \defexercisestyle{problembysheet}%
848   {\exf@numberproblemwithin{\exf@sheetcounter}}
849 \defexercisestyle{equationbysheet}%
850   {\exf@numberequationwithin{\exf@sheetcounter}}
```

fracpoints   Use vulgar fractions to display binary fractional points:

```
851 \defexercisestyle{fracpoints}%
852   {\exerciseconfig{composepointsnum}[1]{\protect\showfracpoints{##1}}}
```

twoside Use two-sided layout for sheets:

```
853 \defexercisestylearg[true]{twoside}{\exercisesetup{twoside={#1}}}
```

Extended Exercise Styles. Declare more specific styles:

```
854 \ifexf@extstyle
```

contents Add sheets and problems to table of contents:

```
855 \defexercisestyle{contents}{%
856   \exerciseconfig{toclevelsheet}{section}%
857   \exerciseconfig{toclevelproblem}{subsection}}
```

solutionsf Use sans serif font for solutions:

```
858 \defexercisestyle{solutionsf}{%
859   \exerciseconfigappend{styletextsolution}{\sffamily\let\itshape\slshape}}
```

solutiondimproblem Dim problem text if solutions are displayed:

```
860 \defexercisestyle{solutiondimproblem}{%
861   \RequirePackage{color}%
862   \exerciseconfigappend{styletextsolution}{\color[gray]{0}}%
863   \exerciseconfigappend{styletextproblem}{\color[gray]{0.2}}}
```

solutionsep Separate solutions by horizontal lines:

```
864 \defexercisestyle{solutionsep}{%
865   \exerciseconfig{insertsolutionsbefore}{\hrule\nopagebreak[3]\vspace{0.5ex}}%
866   \exerciseconfig{insertsolutionsafter}%
867    {\removelastskip\nopagebreak[3]\vspace{1.0ex}\hrule}}
```

plainheader Declare a simple sheet header with some configurable options; the configuration options `styleheader...` define font styles, `skipheaderbelow` the space below the header and `composeheaderbelow...` some auxiliary text to be displayed on the line below the header:

```
868 \defexercisestyle{plainheader}{%
869   \exerciseconfig{styleheadertitle}{\Large\bfseries}%
870   \exerciseconfig{styleheadercourse}{\sffamily}%
871   \exerciseconfig{styleheaderbelow}{\footnotesize}%
872   \exerciseconfig{skipheaderbelow}{3ex}%
873   \exerciseconfig{composeheaderbelowleft}{}%
874   \exerciseconfig{composeheaderbelowright}{}%
875   \exerciseconfig{composeheaderbelowcenter}{}%
876   \exerciseconfig{insertsheettitle}{\noindent%
877    \begin{minipage}{\textwidth}%
878    {\getexerciseconfig{styleheadertitle}%
879     \makebox[0pt][l]{\getexercisedata{course}}%
880     \hfill\makebox[0pt][r]{\getsheetdata{title}}\par}%
881    {\getexerciseconfig{styleheadercourse}%
882     \makebox[0pt][l]{\getexercisedata{institution}%
883      \exercisedataempty{period}{}{, \getexercisedata{period}}}%
884     \hfill\makebox[0pt][r]{\getexercisedata{instructor}}%
885     \vphantom{g}\par}%
886    \hrule%
887    {\def\tmp{}%
888     \exerciseconfigempty{composeheaderbelowleft}{}{\def\tmp{.}}%
889     \exerciseconfigempty{composeheaderbelowcenter}{}{\def\tmp{.}}%
```

```
890    \exerciseconfigempty{composeheaderbelowright}{}{\def\tmp{.}}%
891    \exerciseifnotempty{\tmp}%
892     {\getexerciseconfig{styleheaderbelow}\vphantom{\^A}%
893       \makebox[0pt][l]{\getexerciseconfig{composeheaderbelowleft}}%
894       \hfill\makebox[0pt][c]{\getexerciseconfig{composeheaderbelowcenter}}%
895       \hfill\makebox[0pt][r]{\getexerciseconfig{composeheaderbelowright}}%
896       \vspace*{-\baselineskip}\vspace*{-\parskip}\par}}%
897    \end{minipage}%
898    \par\addvspace{\getexerciseconfig{skipheaderbelow}}}}
```

Done with extended styles:

```
899 \fi
```

C.5   Metadata

Global Metadata Code.

\defexercisedata   Declare global metadata field by defining a key *key* in category exf@data that stores the chosen value in \exf@data@*key*:

```
900 \newcommand{\defexercisedata}[1]{%
901   \exf@csdo\def{exf@data@#1}{}%
902   \define@key{exf@data}{#1}%
903     {\exf@csdo\gdef{exf@data@#1}{##1}}}
```

\exercisedata   Process key-value pairs:

```
904 \newcommand{\exercisedata}[1]{\setkeys{exf@data}{#1}}
```

\getexercisedata   Read global metadata:

```
905 \newcommand{\getexercisedata}[1]{\csname exf@data@#1\endcsname}
```

\exercisedataempty   Check whether the field is empty:

```
906 \newcommand{\exercisedataempty}[3]{\exf@csdo\ifx{exf@data@#1}\exf@empty%
907   #2\else#3\fi}
```

Global Metadata Declarations.   Declare fields corresponding to standard pdf metadata:

```
908 \defexercisedata{author}
909 \defexercisedata{title}
910 \defexercisedata{subject}
911 \defexercisedata{keyword}
```

Declare additional general purpose fields:

```
912 \defexercisedata{date}
```

Declare metadata related to courses:

```
913 \defexercisedata{instructor}
914 \defexercisedata{course}
915 \defexercisedata{institution}
916 \defexercisedata{period}
917 \defexercisedata{material}
```

Overwrite standard definitions for `author`, `title`, `date` to also fill ordinary LaTeX structures:

```
918 \define@key{exf@data}{author}{\gdef\exf@data@author{#1}\author{#1}}
919 \define@key{exf@data}{title}{\gdef\exf@data@title{#1}\title{#1}}
920 \define@key{exf@data}{date}{\gdef\exf@data@date{#1}\date{#1}}
```

Sheet Metadata.

\defsheetdata   Declare sheet metadata field by defining a key *key* in category `exf@sheet` that stores the chosen value in `\exf@data@sheet@`*key*:

```
921 \newcommand{\defsheetdata}[1]{%
922   \exf@csdo\def{exf@data@sheet@#1}{}%
923   \define@key{exf@sheet}{#1}%
924     {\exf@csdo\def{exf@data@sheet@#1}{##1}}}
```

\setsheetdata   Set sheet metadata:

```
925 \newcommand{\setsheetdata}[1]{\setkeys{exf@sheet}{#1}}
```

\getsheetdata   Read sheet metadata:

```
926 \newcommand{\getsheetdata}[1]{\csname exf@data@sheet@#1\endcsname}
```

\sheetdataempty   Check whether the field is empty:

```
927 \newcommand{\sheetdataempty}[3]{\exf@csdo\ifx{exf@data@sheet@#1}\exf@empty%
928   #2\else#3\fi}
```

Declare general purpose fields:

```
929 \defsheetdata{due}
930 \defsheetdata{handout}
931 \defsheetdata{editdate}
932 \defsheetdata{author}
933 \defsheetdata{editor}
```

Special title processing:

```
934 \def\exf@data@sheet@rawtitle{}
935 \define@key{exf@sheet}{title}{\def\exf@data@sheet@rawtitle{#1}}
936 \def\exf@data@sheet@title{\exf@config@composetitlesheet%
937   {\csname the\exf@sheetcounter\endcsname}{\exf@data@sheet@rawtitle}}%
```

Special points processing:

```
938 \def\exf@data@sheet@points{\ifdefined\exf@sheet@points%
939   \expandafter\exf@config@composepointspairsheet\exf@sheet@points\fi}%
```

Problem Metadata.

\defproblemdata   Declare problem metadata field by defining a key *key* in category `exf@problem` that stores the chosen value in `\exf@data@problem@`*key*:

```
940 \newcommand{\defproblemdata}[1]{%
941   \exf@csdo\def{exf@data@problem@#1}{}%
942   \define@key{exf@problem}{#1}%
943     {\exf@csdo\def{exf@data@problem@#1}{##1}}}
```

50

\setproblemdata   Set problem metadata:

```
944 \newcommand{\setproblemdata}[1]{\setkeys{exf@problem,exf@scanproblem}{#1}}
```

\getproblemdata   Read problem metadata:

```
945 \newcommand{\getproblemdata}[1]{\csname exf@data@problem@#1\endcsname}
```

\problemdataempty   Check whether the field is empty:

```
946 \newcommand{\problemdataempty}[3]{\exf@csdo\ifx{exf@data@problem@#1}\exf@empty%
947   #2\else#3\fi}
```

Special title processing:

```
948 \def\exf@data@problem@rawtitle{}
949 \define@key{exf@problem}{title}{\def\exf@data@problem@rawtitle{#1}}
950 \def\exf@data@problem@title{\exf@config@composetitleproblem{%
951   \csname the\exf@problemcounter\endcsname}{\exf@data@problem@rawtitle}}%
```

Problem Environment Code.

\exf@addmargin   Define a length for environment margin:

```
952 \newlength\exf@addmargin
```

\exf@section   Write out problem opening line followed by some amount of skip (positive dimensions add vertical space, negative dimensions add horizontal space); protected expand argument if in horizontal mode (because it will be held until text is output and some definitions may become invalid):

```
953 \newcommand{\exf@section}[2]{\setlength\exf@tmpdim{#1}%
954   \ifdim\exf@tmpdim<0pt%
955     \protected@edef\exf@tmp{#2}%
956   \else%
957     \def\exf@tmp{#2}%
958   \fi%
959   \exf@exparg{\@startsection{}{}{0pt}{0pt}{#1}{}*}{\exf@tmp}}
```

\exf@init@block   Clean info buffer, define amount of skip between items:

```
960 \newcommand{\exf@init@block}[1]{%
961   \def\exf@intro{}\def\exf@intro@skip{#1}%
962   \exf@addmargin0pt\def\exf@introitem{}}
```

\exf@append@intro   Append to info buffer:

```
963 \newcommand{\exf@append@intro}[1]%
964   {\exf@append@def\exf@intro{#1\hspace{\exf@intro@skip}}}
```

\exf@prepend@intro   Prepend to info buffer:

```
965 \newcommand{\exf@prepend@intro}[1]%
966   {\exf@prepend@def\exf@intro{#1\hspace{\exf@intro@skip}}}
```

\exf@open@block   Open environment, set margin, compose opening line:

```
967 \newcommand{\exf@open@block}[1]{%
968   \advance\leftskip\exf@addmargin%
```

```
969    \advance\linewidth-\exf@addmargin%
970    \advance\@totalleftmargin\exf@addmargin%
971    \ifx\exf@intro\exf@empty%
972     \exf@section{0pt}{\exf@introitem}%
973    \else%
974     \exf@section{#1}{\exf@introitem\exf@intro\unskip}%
975    \fi}%
```

\exf@close@block   Close environment, undo margin:

```
976 \newcommand{\exf@close@block}{%
977    \advance\leftskip-\exf@addmargin%
978    \advance\linewidth\exf@addmargin%
979    \advance\@totalleftmargin-\exf@addmargin}%
```

\addprobleminfo   Interface to append or prepend to info buffer:

```
980 \newcommand{\addprobleminfo}{\@ifstar\exf@prepend@intro\exf@append@intro}
```

\exf@addinfoswitch   Add a switch for displaying problem info:

```
981 \newcommand{\exf@addinfoswitch}[1]%
982    {\define@boolkey{exf@infoswitch}[exf@showdata@]{#1}[true]{}}
```

\defprobleminfo   Declare a problem info field, add corresponding info switch, process key-value pair if switch activated:

```
983 \newcommand{\defprobleminfo}[2]{%
984    \exf@addinfoswitch{#1}%
985    \exerciseconfig{compose@probleminfo@#1}[1]{#2}%
986    \exf@exparg{\define@key{exf@probleminfo}{#1}}%
987     {\csname ifexf@showdata@#1\endcsname\exf@append@intro{%
988      \csname exf@config@compose@probleminfo@#1\endcsname{##1}}\fi}}
```

\showprobleminfo   Process info switches, expand argument first:

```
989 \newcommand{\showprobleminfo}[1]{\exf@expsetkeys{exf@infoswitch}{#1}}
```

Problem Info Declarations.   Declare general purpose fields:

```
990 \defprobleminfo{optional}{\emph{#1:}}
991 \showprobleminfo{optional}
992 \defprobleminfo{difficulty}{(#1)}
```

Declare fields for internal information (mostly):

```
993 \defprobleminfo{comment}{#1}
994 \defprobleminfo{author}{$\langle$#1$\rangle$}
995 \defprobleminfo{editor}{$\{$#1$\}$}
996 \defprobleminfo{source}{[#1]}
997 \defprobleminfo{keyword}{\#(#1)}
```

Declare more specific fields:

```
998 \ifexf@extdata
999 \defprobleminfo{review}{#1}
1000 \defprobleminfo{recycle}{[[#1]]}
1001 \defprobleminfo{timesolve}{\{#1\}}
1002 \defprobleminfo{timepresent}{\{\!\{#1\}\!\}}
1003 \fi
```

Write Metadata to PDF Files.

\exf@writemetadata Write Metadata to PDF Files in case hyperref is available:

```
1004 \newcommand{\exf@writemetadata}{%
1005   \ifdefined\hypersetup%
```

Write author, title, subject and keywords:

```
1006     \ifx\exf@data@author\exf@empty\else%
1007      \hypersetup{pdfauthor={\exf@data@author}}\fi%
1008     \ifx\exf@data@title\exf@empty\else%
1009      \hypersetup{pdftitle={\exf@data@title}}\fi%
1010     \ifx\exf@data@subject\exf@empty\else%
1011      \hypersetup{pdfsubject={\exf@data@subject}}\fi%
1012     \ifx\exf@data@keyword\exf@empty\else%
1013      \hypersetup{pdfkeywords={\exf@data@keyword}}\fi%
1014   \fi%
```

Do not write again:

```
1015   \gdef\exf@metadata{off}}
```

Automatic writing at \begin{document}:

```
1016 \AtBeginDocument{\exf@ifis\exf@metadata{auto}{\exf@writemetadata}}
```

\writeexercisedata Write metadata manually:

```
1017 \newcommand{\writeexercisedata}{%
1018   \exf@ifis\exf@metadata{manual}{\exf@writemetadata}}
```

C.6 Counters

sheet Define main counters (with customised names if necessary) and equation counters:
problem
subproblem
solution
```
1019 \newcounter{\exf@sheetcounter}
1020 \newcounter{\exf@problemcounter}
1021 \newcounter{\exf@subproblemcounter}[\exf@problemcounter]
1022 \newcounter{\exf@solutioncounter}[\exf@problemcounter]
1023 \newcount\exf@eqsav
1024 \newcounter{exf@sheetequation}
1025 \newcounter{exf@problemequation}
1026 \newcounter{exf@solutionequation}
```

Implement counter display; take care of corresponding hyperref labels:

```
1027 \exf@csdo\def{the\exf@sheetcounter}{\exf@config@countersheet}
1028 \exf@csdo\def{the\exf@problemcounter}{\exf@config@counterproblem}
1029 \exf@csdo\def{the\exf@subproblemcounter}{\exf@config@countersubproblem}
1030 \def\theexf@sheetequation{\exf@config@countersheetequation}
1031 \def\theHexf@sheetequation{sheet.\arabic{equation}}
1032 \def\theexf@problemequation{\exf@config@counterproblemequation}
1033 \def\theHexf@problemequation{prob.\arabic{equation}}
1034 \def\theexf@solutionequation{\exf@config@countersolutionequation}
1035 \def\theHexf@solutionequation{sol.\arabic{equation}}
```

numberproblemwithin Declare problem counter as subcounter of #1:

```
1036 \newcommand{\exf@numberproblemwithin}[1]{%
1037   \@addtoreset{\exf@problemcounter}{#1}%
```

```
1038    \exf@csdo\def{the\exf@problemcounter}%
1039      {\csname the#1\endcsname.\exf@config@counterproblem}}
```

numberequationwithin   Declare various new equation counters as subcounter of `#1`; take care of corresponding hyperref labels:

```
1040 \newcommand{\exf@numberequationwithin}[1]{%
1041    \@addtoreset{exf@sheetequation}{#1}%
1042    \def\theexf@sheetequation%
1043      {\csname the#1\endcsname.\exf@config@countersheetequation}%
1044    \def\theHexf@sheetequation%
1045      {\csname theH#1\endcsname.sheet.\arabic{equation}}%
1046    \@addtoreset{exf@problemequation}{#1}%
1047    \def\theexf@problemequation%
1048      {\csname the#1\endcsname.\exf@config@counterproblemequation}%
1049    \def\theHexf@problemequation%
1050      {\csname theH#1\endcsname.prob.\arabic{equation}}%
1051    \@addtoreset{exf@solutionequation}{#1}%
1052    \def\theexf@solutionequation%
1053      {\csname the#1\endcsname.\exf@config@countersolutionequation}%
1054    \def\theHexf@solutionequation%
1055      {\csname theH#1\endcsname.sol.\arabic{equation}}}
```

C.7   Buffers

File Output.

`\ifexf@solfile@open`   Conditional whether output files are presently in use:

```
1056 \newif\ifexf@solfile@open\exf@solfile@openfalse
1057 \newif\ifexf@probfile@open\exf@probfile@openfalse
```

`\exf@solfile`   Reserve file handles:

```
1058 \newwrite\exf@solfile
1059 \newwrite\exf@probfile
```

`\exf@writeline`   Write a line to the file:

```
1060 \newcommand{\exf@writeline}[2]{\immediate\write#1{#2}}
```

`\exf@linesep`   Return a separator line:

```
1061 \newcommand{\exf@linesep}%
1062    {\@percentchar------------------------------------}
```

`\exf@lineno`   Return current position in source file; display line number and source file name (if available via package currfile):

```
1063 \newcommand{\exf@lineno}{\@percentchar%
1064    \ifdefined\currfilename\currfilename\space\fi%
1065    l.\the\inputlineno}
```

`\exf@start@solfile`   Open a new solution file `#1.sol` (do nothing if already open); indicate source, switch to manual solution display mode:

```
1066 \newcommand{\exf@start@solfile}[1]{%
1067    \ifexf@solfile@open\else%
```

```
1068    \exercisestyle{solutionbelow=manual}%
1069    \global\exf@solfile@opentrue%
1070    \edef\exf@tmp{#1}%
1071    \immediate\openout\exf@solfile\exf@tmp\exf@config@extsolutions\relax%
1072    \exf@writeline\exf@solfile{\@percentchar%
1073     generated from file '\jobname' by exframe.sty}%
1074    \ifexf@lineno\exf@writeline\exf@solfile{\exf@lineno}\fi%
1075    \exf@writeline\exf@solfile{}%
1076  \fi}
```

\exf@close@solfile   Close solution file (if open); indicate position, close and reset variables:

```
1077 \newcommand{\exf@close@solfile}{%
1078   \ifexf@solfile@open%
1079    \ifexf@lineno\exf@writeline\exf@solfile{\exf@linesep}%
1080     \exf@writeline\exf@solfile{\exf@lineno}\fi%
1081    \exf@writeline\exf@solfile{\@backslashchar endinput}%
1082    \immediate\closeout\exf@solfile%
1083    \global\exf@solfile@openfalse%
1084  \fi}
```

\exf@start@probfile   Open a new problem file #1.prb (do nothing if already open); indicate source, switch to manual problem display mode:

```
1085 \newcommand{\exf@start@probfile}[1]{%
1086   \ifexf@probfile@open\else%
1087    \global\exf@probfile@opentrue%
1088    \edef\exf@tmp{#1}%
1089    \immediate\openout\exf@probfile\exf@tmp\exf@config@extproblems\relax%
1090    \exf@writeline\exf@probfile{\@percentchar%
1091     generated from file '\jobname' by exframe.sty}%
1092    \ifexf@lineno\exf@writeline\exf@probfile{\exf@lineno}\fi%
1093    \exf@writeline\exf@probfile{}%
1094  \fi}
```

\exf@close@probfile   Close problem file (if open); indicate position, close and reset variables:

```
1095 \newcommand{\exf@close@probfile}{%
1096   \ifexf@probfile@open%
1097    \ifexf@lineno\exf@writeline\exf@probfile{\exf@linesep}%
1098     \exf@writeline\exf@probfile{\exf@lineno}\fi%
1099    \exf@writeline\exf@probfile{\@backslashchar endinput}%
1100    \immediate\closeout\exf@probfile%
1101    \global\exf@probfile@openfalse%
1102  \fi}
```

Make sure to properly close files at the end:

```
1103 \AtEndDocument{\exf@close@solfile\exf@close@probfile}
```

Buffers.

\exf@solbuf   Declare token buffers for storing problems and solutions and conditionals indicating whether
\exf@probbuf   the buffers have been used:
\ifexf@solbuf@clean
\ifexf@probbuf@clean
```
1104 \newtoks\exf@solbuf
1105 \newtoks\exf@probbuf
1106 \newif\ifexf@solbuf@clean\exf@solbuf@cleantrue
1107 \newif\ifexf@probbuf@clean\exf@probbuf@cleantrue
```

| `\exf@clear@solbuf` | Clear a buffer and mark clean: |
| `\exf@clear@probbuf` | |

```
1108 \def\exf@clear@solbuf{\global\exf@solbuf@cleantrue\global\exf@solbuf={}}
1109 \def\exf@clear@probbuf{\global\exf@probbuf@cleantrue\global\exf@probbuf={}}
```

`\exf@append@buf` Append tokens to buffer:

```
1110 \def\exf@append@buf#1#2{\global#1=\expandafter{\the#1#2}}
```

`\exf@addline` Add a protected expanded line to the buffer:

```
1111 \def\exf@addline#1#2{{\protected@edef\exf@tmp{#2}%
1112   \exf@exparg{\exf@append@buf#1}{\exf@tmp^^J}}}
```

`\exf@source@buf` Source a buffer into the document:

```
1113 \def\exf@source@buf#1{\exf@exptwo\scantokens{\the#1}}
```

`\exf@write@buf` Write the buffer into the solution file:

```
1114 \def\exf@write@buf#1#2{\exf@writeline#1{\the#2}}
```

Verbatim Processing.

`\exf@verbatim` Start reading the buffer from the environment body:

```
1115 \newcommand{\exf@verbatim}{%
1116   \begingroup%
1117   \@bsphack%
1118   \let\do\@makeother\dospecials%
1119   \catcode'\^^M\active%
1120   \def\verbatim@processline{\exf@exptwo\exf@verbatim@process%
1121     {\the\verbatim@line^^J}}%
1122   \verbatim@start}
```

`\exf@endverbatim` Stop reading the buffer:

```
1123 \newcommand{\exf@endverbatim}{\@esphack\endgroup}
```

`\exf@scanblock` Scan an optional argument from a verbatim environment; allow for an empty environment and an empty first line; argument #1 is macro to be called eventually:

```
1124 \def\exf@scanblock#1{%
```

Check for empty first line:

```
1125   \@ifnextchar\par{\exf@scanblock@par{#1}}{\exf@scanblock@sel{#1}}}
```

Handle empty first line, implies no optional argument:

```
1126 \long\def\exf@scanblock@par#1\par{\exf@scanblock@sel{#1}[]}
```

Check for optional argument (`[`) and for environment ending (`\end`):

```
1127 \def\exf@scanblock@sel#1{\@ifnextchar[{\exf@scanblock@opt{#1}}%
1128   {\@ifnextchar\end{\exf@scanblock@end{#1}}{\exf@scanblock@noopt{#1}}}}
```

Handle empty environment, hopefully environment matches (otherwise?!):

```
1129 \def\exf@scanblock@end#1\end#2{%
1130   \def\exf@tmp{#2}\ifx\exf@tmp\@currenvir%
```

```
1131      \def\exf@verbatim{}\def\exf@endverbatim{}%
1132    \fi%
1133    #1{}{\scantokens{\end{#2}}}}
```

Pass on without and with optional argument; pass on optional argument and any token scanned prematurely:

```
1134 \def\exf@scanblock@noopt#1#2{#1{}{\scantokens#2}}
1135 \def\exf@scanblock@opt#1[#2]{#1{#2}{}}
```

C.8   Points

Points Arithmetic.

\exf@addtopoints   Add points #2+#3 to macro #1 using metric register:

```
1136 \newcommand{\exf@addtopoints}[3]{%
1137    \ifdefined#1\else\def#1{{0}{0}}\fi%
1138    \setlength\exf@tmpdim{\expandafter\@firstoftwo#1pt}%
1139    \addtolength\exf@tmpdim{#2pt}%
1140    \edef\exf@tmp{\strip@pt\exf@tmpdim}%
1141    \setlength\exf@tmpdim{\expandafter\@secondoftwo#1pt}%
1142    \addtolength\exf@tmpdim{#3pt}%
1143    \xdef#1{{\exf@tmp}{\strip@pt\exf@tmpdim}}}
```

\exf@pointsmismatch   Execute #3 if points disagree:

```
1144 \newcommand{\exf@pointsmismatch}[3]{%
1145    \let\exf@tmp\@undefined%
1146    \ifdim\expandafter\@firstoftwo#1pt=\expandafter\@firstoftwo#2pt\else%
1147      \def\exf@tmp{}\fi%
1148    \ifdim\expandafter\@secondoftwo#1pt=\expandafter\@secondoftwo#2pt\else%
1149      \def\exf@tmp{}\fi%
1150    \ifdefined\exf@tmp#3\fi}
```

Points Expansion.

\exf@outpoints   If points macro #3 is set, expand #3, pass on as #1{\protect#2}#3 and clear #3 globally:

```
1151 \def\exf@outpoints#1#2#3{\ifdefined#3%
1152    \exf@exptwo\exf@outpoints@switch{#3}{#1}{#2}%
1153    \global\let#3\@undefined\fi}
1154 \def\exf@outpoints@switch#1#2#3{#2{\protect#3#1}}
```

\exf@scanpoints   Call as \exf@scanpoints#1[*regular*][+*bonus*]++& to write {*regular*}{*bonus*} into #1; fill with 0 if empty:

```
1155 \def\exf@scanpoints#1#2+#3+#4&{%
1156    \edef#1{\if&#2&0\else#2\fi}%
1157    \edef#1{{#1}{\if&#3&0\else#3\fi}}}
```

\exf@formatpoints   Format as [#1][+#2]; remove 0 components:

```
1158 \def\exf@formatpoints#1#2{\ifdim#2pt=0pt#1\else%
1159    \ifdim#1pt=0pt+#2\else#1+#2\fi\fi}
```

\extractpoints   Extract main (plain) or bonus (starred) part from saved points register:

```
1160 \newcommand{\extractpoints}{\@ifstar{\exf@extractpoints\@secondoftwo}%
1161   {\exf@extractpoints\@firstoftwo}}
1162 \newcommand{\exf@extractpoints}[2]{\edef\exf@tmp{#2}%
1163   \exf@exptwo\exf@scanpoints\exf@tmp\exf@tmp++&%
1164   \expandafter#1\exf@tmp}
```

\switchpoints Extract main (plain) and bonus (starred) part from points, and execute one of three:

```
1165 \newcommand{\switchpoints}[5]{\edef\exf@tmp{#5}%
1166   \exf@exptwo\exf@scanpoints\exf@tmp\exf@tmp++&%
1167   \expandafter\exf@switchpoints\exf@tmp{#1}{#2}{#3}{#4}}
1168 \newcommand{\exf@switchpoints}[6]{%
1169   \ifdim#2pt=0pt\ifdim#1pt=0pt\def\exf@tmp##1##2{#6}%
1170   \else\def\exf@tmp##1##2{#3}\fi%
1171   \else\ifdim#1pt=0pt\def\exf@tmp##1##2{#4}%
1172   \else\def\exf@tmp##1##2{#5}\fi\fi\exf@tmp{#1}{#2}}
```

Tools.

xf@makepointsmargin Combination to typeset points in margin:

```
1173 \newcommand{\exf@makepointsmargin}[2]{%
1174   \exf@config@insertpointsmargin{\exf@config@composepointspairmargin{#1}{#2}}}
```

\exf@warnmismatch If points #3 and #4 are defined and disagree, issue a warning message:

```
1175 \newcommand{\exf@warnmismatch}[4]{%
1176   \ifdefined#4\ifdefined#3\exf@pointsmismatch#3#4{%
1177   \let\exf@tmp\PackageWarning%
1178   \ifx#1\exf@solutionname\let\exf@tmp\PackageWarningNoLine\fi%
1179   \exf@tmp{exframe}{points mismatch %
1180     (\expandafter\exf@formatpoints#3 determined %
1181     vs. \expandafter\exf@formatpoints#4 given) %
1182     for #1 \csname the#2\endcsname}%
1183   \ifexf@warntext\edef\exf@tmp{%
1184     {\expandafter\exf@formatpoints#3}{\expandafter\exf@formatpoints#4}}%
1185   \exf@exptwo\exf@config@insertwarnpoints#1\exf@tmp\fi}%
1186   \fi\fi}
```

\exf@warnrerun If points #3 and #4 are defined and disagree, issue advice to recompile:

```
1187 \newcommand{\exf@warnrerun}[4]{%
1188   \ifdefined#4\ifdefined#3\exf@pointsmismatch#3#4{%
1189   \PackageWarning{exframe}{points changed %
1190     for #1 \csname the#2\endcsname; rerun to fix}%
1191   \ifexf@warntext\exf@config@insertwarnpointsrerun#1\fi}%
1192   \fi\fi}
```

Binary Rational Numbers.

\exf@splitsign Split a decimal float number into sign, integer and fractional part:
\exf@splitdecimal

```
1193 \def\exf@splitsign#1-#2-#3&{\def\exf@splitnum{#1#2}\def\exf@splitminus{#3}}
1194 \def\exf@splitdecimal#1.#2.#3&{\def\exf@splitint{#1}\def\exf@splitdec{#2}}
```

\showfracpoints Display a float number as a fraction with denominators 2, 4 or 8 when possible; first split number, complete missing zeros and handle cases:

```
1195 \newcommand{\showfracpoints}[1]{%
1196     \edef\exf@tmp{#1}%
1197     \expandafter\exf@splitsign\exf@tmp--&%
1198     \expandafter\exf@splitdecimal\exf@splitnum..&%
1199     \if&\exf@splitint&\def\exf@splitint{0}\fi%
1200     \if&\exf@splitdec&\def\exf@splitdec{0}\fi%
1201     \def\exf@tmp{\exf@splitint.\exf@splitdec}%
1202     \ifnum\exf@splitdec=0\def\exf@tmp{\exf@splitint}\fi%
1203     \ifnum\exf@splitdec=5\def\exf@tmp{\exf@config@frac{\exf@splitint}{1}{2}}\fi%
1204     \ifnum\exf@splitdec=25\def\exf@tmp{\exf@config@frac{\exf@splitint}{1}{4}}\fi%
1205     \ifnum\exf@splitdec=75\def\exf@tmp{\exf@config@frac{\exf@splitint}{3}{4}}\fi%
1206     \ifnum\exf@splitdec=125\def\exf@tmp{\exf@config@frac{\exf@splitint}{1}{8}}\fi%
1207     \ifnum\exf@splitdec=375\def\exf@tmp{\exf@config@frac{\exf@splitint}{3}{8}}\fi%
1208     \ifnum\exf@splitdec=625\def\exf@tmp{\exf@config@frac{\exf@splitint}{5}{8}}\fi%
1209     \ifnum\exf@splitdec=875\def\exf@tmp{\exf@config@frac{\exf@splitint}{7}{8}}\fi%
1210     \ifx\exf@splitminus\exf@empty\else$\exf@splitminus$\fi\exf@tmp%
1211 }
```

`\exf@config@frac` Display a vulgar fraction such as $12^3/_4$:

```
1212 \newcommand{\exf@config@frac}[3]{%
1213     \ifnum#1=0\else#1\fi%
1214     \ifnum#2=0\else$%
1215         ^{\exf@text{#2}}%
1216         \mskip-4mu/\mskip-2mu%
1217         _{\exf@text{#3}}$\fi}
```

Sheet Points Code.

`exf@notesheetpoints` Store a sheet point number in a macro:

```
1218 \newcommand{\exf@notesheetpoints}[2]{%
1219     \exf@csdo\gdef{exf@sheetpoints@#1}{#2}}
```

Make sure the macros in code written to the `.aux` file exist:

```
1220 \AtBeginDocument{\immediate\write\@auxout{%
1221     \string\providecommand{\string\exf@notesheetpoints}[2]{}}}
```

`xf@writesheetpoints` Write sheet points to the `.aux` file:

```
1222 \newcommand{\exf@writesheetpoints}[2]%
1223     {\immediate\write\@auxout{\string\exf@notesheetpoints{\sheettag}%
1224         {\exf@formatpoints{#1}{#2}}}}
```

`\getsheetpoints` Read points for current sheet or from `.aux` file:

```
1225 \newcommand{\getsheetpoints}[1]{\if&#1&%
1226     \ifdefined\exf@sheet@points%
1227         \expandafter\exf@formatpoints\exf@sheet@points\else 0\fi%
1228     \else\ifcsname exf@sheetpoints@#1\endcsname%
1229         \csname exf@sheetpoints@#1\endcsname\else 0\fi\fi}
```

Problem Points Code.

`f@noteproblempoints` Store a problem point number in a macro:

```
1230 \newcommand{\exf@noteproblempoints}[2]{%
1231     \exf@csdo\gdef{exf@problempoints@#1}{#2}}
```

Make sure the macros written to the `.aux` file exist:

```
1232 \AtBeginDocument{\immediate\write\@auxout{%
1233   \string\providecommand{\string\exf@noteproblempoints}[2]{}}}
```

`@writeproblempoints` Write problem points to the `.aux` file:

```
1234 \newcommand{\exf@writeproblempoints}[2]%
1235   {\immediate\write\@auxout{\string\exf@noteproblempoints{\problemtag}%
1236     {\exf@formatpoints{#1}{#2}}}}
```

`\getproblempoints` Read points for current problem or from `.aux` file:

```
1237 \newcommand{\getproblempoints}[1]{\if&#1&%
1238   \ifdefined\exf@problem@points%
1239     \expandafter\exf@formatpoints\exf@problem@points\else 0\fi%
1240   \else\ifcsname exf@problempoints@#1\endcsname%
1241     \csname exf@problempoints@#1\endcsname\else 0\fi\fi}
```

`getsubproblempoints` Read points for current subproblem:

```
1242 \newcommand{\getsubproblempoints}[1]{\if&#1&%
1243   \ifdefined\exf@subproblem@points%
1244     \expandafter\exf@formatpoints\exf@subproblem@points\else 0\fi%
1245   \else 0\fi}
```

`\showpoints` Show points within a problem or subproblem:

```
1246 \newcommand{\showpoints}{%
1247   \ifdefined\exf@in@solution\exf@outpoints{\exf@ensuretext}%
1248     {\exf@config@composepointspairbodysolution}{\exf@solution@points@show}%
1249   \else\ifdefined\exf@in@subproblem\exf@outpoints{\exf@ensuretext}%
1250     {\exf@config@composepointspairbodysubproblem}{\exf@subproblem@points@show}%
1251   \else\ifdefined\exf@in@problem\exf@outpoints{\exf@ensuretext}%
1252     {\exf@config@composepointspairbodyproblem}{\exf@problem@points@show}%
1253   \fi\fi\fi}
```

Solution Points Code.

`\exf@awardpointsalt` Award points for alternative or optional solution; does not count towards solution total:

```
1254 \newcommand{\exf@awardpointsalt}[2][]{\exf@scanpoints\exf@tmp#2++&%
1255   \exf@exptwo\exf@ensuretext{%
1256     \expandafter\exf@config@composepointspairawardalt\exf@tmp{#1}}}
```

`\exf@awardpointsreg` Award points for regular solution; counts towards solution total:

```
1257 \newcommand{\exf@awardpointsreg}[2][]{\exf@scanpoints\exf@tmp#2++&%
1258   \exf@exptwo\exf@addtopoints\exf@solution@points@total\exf@tmp%
1259   \exf@scanpoints\exf@tmp#2++&%
1260   \exf@exptwo\exf@ensuretext{%
1261     \expandafter\exf@config@composepointspairaward\exf@tmp{#1}}}
```

`\awardpoints` Award points within solution with optional starred form:

```
1262 \newcommand{\awardpoints}{\@ifstar\exf@awardpointsalt\exf@awardpointsreg}
```

`\getsolutionpoints` Read points for current solution:

```
1263 \newcommand{\getsolutionpoints}[1]{\if&#1&%
1264   \ifdefined\exf@solution@points%
1265     \expandafter\exf@formatpoints\exf@solution@points\else 0\fi%
1266   \else 0\fi}
```

C.9   Sheet Environment

exf@sheet   Define options for `sheet` environment:

```
1267 \define@key{exf@sheet}{points}{\exf@scanpoints\exf@sheet@points#1++&}
1268 \define@key{exf@sheet}{number}{\setcounter{\exf@sheetcounter}{#1}}
1269 \define@key{exf@sheet}{label}{\def\exf@label{#1}}
1270 \define@key{exf@sheet}{tag}{\def\sheettag{#1}}
```

sheet   Define `sheet` environment (potentially using custom name):

```
1271 \newenvironment{\exf@sheetname}[1][]{%
```

Insert hook code to clear page, step counter:

```
1272   \exf@config@insertsheetclearpage%
1273   \refstepcounter{\exf@sheetcounter}%
```

Use equation counter for sheets:

```
1274   \ifexf@style@sheetequation%
1275   \exf@eqsav\value{equation}\relax%
1276   \setcounter{equation}{\value{exf@sheetequation}}%
1277   \let\theequation\theexf@sheetequation%
1278   \let\theHequation\theHexf@sheetequation%
1279   \fi%
```

Reset optional arguments, process arguments:

```
1280   \let\exf@sheet@points\@undefined%
1281   \def\sheettag{\csname the\exf@sheetcounter\endcsname}%
1282   \let\exf@sheet@points@total\@undefined%
1283   \let\exf@label\@undefined%
1284   \setkeys{exf@sheet}{#1}%
```

Process automatic and manual labels:

```
1285   \ifexf@autolabelsheet\label{\exf@config@labelsheet{\sheettag}}\fi%
1286   \ifdefined\exf@label\label{\exf@label}\fi%
```

Set points from explicit input or from `.aux` storage:

```
1287   \ifdefined\exf@sheet@points%
1288   \let\exf@sheet@points@given\exf@empty%
1289   \else%
1290   \let\exf@sheet@points@given\@undefined%
1291   \ifcsname exf@sheetpoints@\sheettag\endcsname%
1292     \exf@csdotwo\let\exf@tmp{exf@sheetpoints@\sheettag}%
1293     \exf@exptwo\exf@scanpoints\exf@sheet@points\exf@tmp++&%
1294   \fi\fi%
```

Process metadata:

```
1295   \exf@ifis\exf@metadata{sheet}{{%
1296   \ifx\exf@data@sheet@author\exf@empty\else%
1297     \let\exf@data@author\exf@data@sheet@author\fi%
1298     \def\exf@data@title{\exf@config@composemetasheet%
```

```
1299        {\csname the\exf@sheetcounter\endcsname}{\exf@data@sheet@rawtitle}}%
1300     \exf@writemetadata}}%
```

Insert hook code:

```
1301     \exf@config@insertsheetbefore%
```

Add table of contents line:

```
1302     \ifx\exf@config@toclevelsheet\exf@empty\else%
1303     \ifdefined\phantomsection\phantomsection\fi\fi%
1304   \exf@addcontentsline{\exf@config@toclevelsheet}%
1305     {\exf@config@composetocsheet{\csname the\exf@sheetcounter\endcsname}%
1306       {\exf@data@sheet@rawtitle}}%
```

Write sheet title:

```
1307     \exf@config@insertsheettitle}%
```

End of environment; perform sanity check on total points if given explicitly:

```
1308   {\ifdefined\exf@sheet@points@given%
1309     \exf@warnmismatch{\exf@sheetname}{\exf@sheetcounter}%
1310       {\exf@sheet@points@total}{\exf@sheet@points}%
```

Test whether points have changed since last compile:

```
1311     \else%
1312     \exf@warnrerun{\exf@sheetname}{\exf@sheetcounter}%
1313       {\exf@sheet@points@total}{\exf@sheet@points}%
```

Store points:

```
1314     \let\exf@sheet@points\exf@sheet@points@total%
1315     \fi%
```

Write sheet points to `.aux` file:

```
1316     \ifdefined\exf@sheet@points%
1317     \expandafter\exf@writesheetpoints\exf@sheet@points%
1318     \fi%
```

Insert solutions:

```
1319     \exf@ifis\exf@solutionbelow{sheet}{\insertsolutions}%
```

Insert hook code:

```
1320     \exf@config@insertsheetafter%
1321     \exf@config@insertsheetclearpage%
```

Restore original equation counter:

```
1322     \ifexf@style@sheetequation%
1323     \setcounter{exf@sheetequation}{\value{equation}}%
1324     \setcounter{equation}{\exf@eqsav}%
1325     \fi%
```

Done:

```
1326     \ignorespacesafterend}
```

ciscleardoublepage Clear the current page, clear even page with a totally empty page:

```
1327 \newcommand{\exercisecleardoublepage}{%
1328   \clearpage\ifexf@twoside\ifodd\value{page}\else%
1329   \thispagestyle{empty}\hbox{}\newpage\fi\fi}
```

C.10 Problem Environment

Print Problems.

exf@problem Define options for `problem` environment:

```
1330 \define@key{exf@problem}{points}{\exf@scanpoints\exf@problem@points#1++&}
1331 \define@key{exf@problem}{label}{\def\exf@label{#1}}
1332 \define@key{exf@problem}{tag}{\def\problemtag{#1}}
1333 \define@key{exf@problem}{sollabel}{\xdef\exf@sollabel{#1}}
```

printproblem Define `printproblem` environment:

```
1334 \newenvironment{printproblem}[1]{%
```

Start with new paragraph, set text style, add vspace:

```
1335    \par\exf@config@styletext\addvspace{\exf@config@skipproblemabove}%
```

Step problem counter:

```
1336    \refstepcounter{\exf@problemcounter}%
```

Insert hook code:

```
1337    \exf@config@insertproblembefore%
```

Begin inner group, mark in problem:

```
1338    \begingroup%
1339    \def\exf@in@problem{}%
```

Use equation counter for problems:

```
1340    \ifexf@style@problemequation%
1341    \exf@eqsav\value{equation}\relax%
1342    \setcounter{equation}{\value{exf@problemequation}}%
1343    \let\theequation\theexf@problemequation%
1344    \let\theHequation\theHexf@problemequation%
1345    \fi%
```

Initialise variables, process arguments:

```
1346    \exf@init@block{\exf@config@skipprobleminfo}%
1347    \def\problemtag{\csname the\exf@problemcounter\endcsname}%
1348    \let\exf@problem@points\@undefined%
1349    \let\exf@label\@undefined%
1350    \global\let\exf@sollabel\@undefined%
1351    \let\exf@problem@points@total\@undefined%
1352    \setkeys{exf@problem,exf@probleminfo,exf@scanproblem}{#1}%
```

Process automatic and manual labels:

```
1353    \ifexf@autolabelproblem\label{\exf@config@labelproblem{\problemtag}}\fi%
1354    \ifdefined\exf@label\label{\exf@label}\fi%
```

Mark for new solution section; remember problem counter, title:

```
1355    \gdef\exf@problem@solnewsec{}%
1356    \xdef\exf@prevprob{\csname the\exf@problemcounter\endcsname}%
1357    \ifcsname theH\exf@problemcounter\endcsname%
1358    \xdef\exf@prevprobhref{\exf@problemcounter.%
1359      \csname theH\exf@problemcounter\endcsname}%
1360    \fi%
```

```
1361    \ifx\exf@data@problem@rawtitle\exf@empty%
1362     \global\let\exf@prevprobtitle\@undefined%
1363    \else%
1364     \protected@xdef\exf@prevprobtitle{\exf@data@problem@rawtitle}%
1365    \fi%
1366    \global\let\exf@prevsubprob\@undefined%
1367    \global\let\exf@prevsubprobhref\@undefined%
```

Set points from explicit input or from `.aux` storage:

```
1368    \ifdefined\exf@problem@points%
1369     \let\exf@problem@points@given\exf@empty%
1370    \else%
1371     \let\exf@problem@points@given\@undefined%
1372     \ifcsname exf@problempoints@\problemtag\endcsname%
1373      \exf@csdotwo\let\exf@tmp{exf@problempoints@\problemtag}%
1374      \exf@exptwo\exf@scanpoints\exf@problem@points\exf@tmp++&%
1375     \fi\fi%
1376    \global\let\exf@prevpoints\exf@problem@points%
1377    \let\exf@problem@points@show\@undefined%
1378    \ifdefined\exf@problem@points%
1379     \let\exf@problem@points@show\exf@problem@points%
1380    \fi%
```

Disable points display if desired:

```
1381    \exf@ifis\exf@pointsat{off}{\let\exf@problem@points@show\@undefined}%
```

Display points in opening line if desired; expand points into argument and remove points:

```
1382    \exf@ifis\exf@pointsat{start}{\exf@outpoints{\exf@append@intro}%
1383     {\exf@config@composepointspairstartproblem}{\exf@problem@points@show}}%
1384    \exf@ifis\exf@pointsat{start*}{\exf@outpoints{\exf@prepend@intro}%
1385     {\exf@config@composepointspairstartproblem}{\exf@problem@points@show}}%
```

Insert hook code, set problem body style:

```
1386    \exf@config@insertprobleminfo%
1387    \exf@config@styletextproblem%
```

Write title without item:

```
1388    \ifdim\exf@config@skipproblemitem=0pt%
1389     \exf@prepend@intro{{%
1390      \exf@config@styletitle\exf@config@styletitleproblem%
1391      \exf@config@composetitleproblem{\csname the\exf@problemcounter\endcsname}%
1392       {\exf@data@problem@rawtitle}}}%
```

Write item with fixed total width or item width plus space:

```
1393    \else%
1394     \ifdim\exf@config@skipproblemitem>0pt%
1395      \setlength\exf@addmargin{\exf@config@skipproblemitem}%
1396     \else%
1397      \settowidth\exf@addmargin{%
1398       \exf@config@styletitle\exf@config@styletitleproblem%
1399       \exf@config@composeitemproblem{\exf@config@counterproblemmax}%
1400       \exf@config@composeitemproblemsep}%
1401     \fi%
```

Define item label:

```
1402    \def\exf@introitem{\makebox[0cm][r]{%
```

```
1403    \exf@config@styletitle\exf@config@styletitleproblem%
1404    \exf@config@composeitemproblem{\csname the\exf@problemcounter\endcsname}%
1405     \exf@config@composeitemproblemsep}}%
```

Compose title:

```
1406    \ifx\exf@data@problem@rawtitle\exf@empty\else%
1407    \exf@prepend@intro{{%
1408      \exf@config@styletitle\exf@config@styletitleproblem%
1409      \exf@config@composetitleproblem{\exf@empty}{\exf@data@problem@rawtitle}}}%
1410    \fi%
1411   \fi%
```

Write points into margin if desired; expand points into argument and remove points:

```
1412   \exf@ifis\exf@pointsat{margin}{%
1413    \exf@outpoints{\exf@prepend@def\exf@introitem}%
1414    {\exf@makepointsmargin}{\exf@problem@points@show}}%
```

Write out opening line:

```
1415   \exf@open@block{\exf@config@skipproblemtitle}%
```

Add table of contents line:

```
1416   \exf@addcontentsline{\exf@config@toclevelproblem}%
1417    {\exf@config@composetocproblem{\csname the\exf@problemcounter\endcsname}%
1418     {\exf@data@problem@rawtitle}}%
```

Done:

```
1419   \@afterindentfalse}%
```

End environment, show points if desired:

```
1420  {\exf@ifis\exf@pointsat{end}{\showpoints}%
```

Perform sanity checks on total points if given explicitly:

```
1421   \ifdefined\exf@problem@points@given%
1422    \exf@warnmismatch{\exf@problemname}{\exf@problemcounter}%
1423    {\exf@problem@points@total}{\exf@problem@points}%
```

Warn if calculated total points have changed:

```
1424   \else%
1425    \exf@warnrerun{\exf@problemname}{\exf@problemcounter}%
1426    {\exf@problem@points@total}{\exf@problem@points}%
```

Read computed total points:

```
1427    \let\exf@problem@points\exf@problem@points@total%
1428   \fi%
```

Write points to `.aux` file; add to sheet total:

```
1429   \ifdefined\exf@problem@points%
1430    \expandafter\exf@writeproblempoints\exf@problem@points%
1431    \exf@exptwo\exf@addtopoints\exf@sheet@points@total\exf@problem@points%
1432   \fi%
1433   \ifdefined\exf@problem@points\else\ifdefined\exf@sheet@points@total%
1434    \PackageWarning{exframe}{no points defined for \exf@problemname}%
1435   \fi\fi%
```

Solutions to subproblems must be declared within problem environment:

```
1436    \global\let\exf@prevsubprob\@undefined%
1437    \global\let\exf@prevsubprobhref\@undefined%
```

End paragraph and environment:

```
1438    \par\exf@close@block%
```

Display solution if desired:

```
1439    \exf@ifis\exf@solutionbelow{problem}{%
1440      \exf@config@insertproblemsolution%
1441      \exf@showsolutions{\exf@config@composetitlesolutionmulti}{}}%
```

Restore original equation counter:

```
1442    \ifexf@style@problemequation%
1443      \setcounter{exf@problemequation}{\value{equation}}%
1444      \setcounter{equation}{\exf@eqsav}%
1445    \fi%
```

End inner group:

```
1446    \endgroup%
```

Insert hook code, vertical skip:

```
1447    \exf@config@insertproblemafter%
1448    \addvspace{\exf@config@skipproblembelow}%
```

Display solution if desired:

```
1449    \exf@ifis\exf@solutionbelow{problem*}{%
1450      \exf@showsolutions{\exf@config@composetitlesolutionmulti}{}}%
```

Done:

```
1451    \ignorespacesafterend}
```

Read Problem Environment.

exf@scanproblem Define options for `problem` environment:

```
1452 \define@boolkey{exf@scanproblem}[exf@scanproblem@]{disable}[true]{}
```

exf@problem@direct Define direct output version of `problem` environment; pass on to `printproblem` environment:

```
1453 \newenvironment{exf@problem@direct}[1][]%
1454   {\printproblem{#1}}{\endprintproblem\ignorespacesafterend}
```

exf@problem@scan
exf@scanproblem Define scan version of `problem` environment; use `\exf@scanblock` to properly parse optional argument and pass on to `exf@scanproblem`:

```
1455 \newenvironment{exf@problem@scan}%
1456   {\exf@scanblock{\exf@scanproblem}}{\endexf@scanproblem}%
1457 \newenvironment{exf@scanproblem}[2]{%
```

Determine problem display:

```
1458    \exf@scanproblem@disablefalse%
1459    \setkeys*{exf@scanproblem}{#1}%
1460    \exf@config@insertproblemselect{#1}%
```

Write separator and `printproblem` environment to buffer:

```
1461   \ifexf@scanproblem@disable%
1462   \def\exf@verbatim@process{\@gobble}%
1463   \else%
1464   \ifexf@lineno\exf@addline\exf@probbuf{\exf@linesep}%
1465    \exf@addline\exf@probbuf{\exf@lineno}\fi%
1466   \exf@addline\exf@probbuf%
1467    {\@backslashchar begin{printproblem}{#1}}%
1468   \def\exf@verbatim@process{\exf@append@buf\exf@probbuf}%
1469   \fi%
```

Start verbatim processing:

```
1470   \exf@verbatim#2}%
```

End verbatim processing; close `printproblem` environment:

```
1471  {\exf@endverbatim%
1472   \ifexf@scanproblem@disable\else%
1473   \exf@addline\exf@probbuf{\@backslashchar end{printproblem}}%
1474   \global\exf@probbuf@cleanfalse%
1475   \fi%
```

Write buffer to file if output file open:

```
1476   \ifexf@probfile@open%
1477    \exf@write@buf\exf@probfile\exf@probbuf%
1478    \exf@clear@probbuf%
1479   \fi%
```

Output buffer immediately:

```
1480   \ifexf@problemmanual\else%
1481    \exf@source@buf\exf@probbuf%
1482    \exf@clear@probbuf%
1483   \fi%
```

Done:

```
1484   \ignorespacesafterend}
```

problem Define `problem` environment (potentially using custom name) to choose between direct and buffered version:

```
1485 \newenvironment{\exf@problemname}%
1486 {\ifexf@problembuf\let\exf@tmp\exf@problem@scan%
1487  \else\let\exf@tmp\exf@problem@direct\fi%
1488  \exf@tmp}%
1489 {\ifexf@problembuf\let\exf@tmp\endexf@problem@scan%
1490  \else\let\exf@tmp\endexf@problem@direct\fi%
1491  \exf@tmp}
```

C.11   Problem Blocks

Problem Block Handling.

\exf@problemstitle   Compose the title for a problem block section:

```
1492 \newcommand{\exf@problemstitle}{%
```

Check whether title is empty:

```
1493  \protected@edef\exf@problemstitleexp{\exf@config@composetitleproblems}%
1494  \ifx\exf@problemstitleexp\exf@empty\else%
```

Output section line:

```
1495    \exf@section{\exf@config@skipproblemstitle}%
1496      {\exf@config@styletitle\exf@config@styletitleproblems%
1497       \exf@problemstitleexp}%
1498    \exf@addcontentsline{\exf@config@toclevelproblems}%
1499      {\exf@config@composetocproblems}%
1500    \fi}
```

\exf@showproblemsin Output problem block intro:

```
1501 \newcommand{\exf@showproblemsin}{%
```

Set problem body style; add vertical space; insert hook code:

```
1502    \par\exf@config@styletext\addvspace{\exf@config@skipproblemsabove}%
1503    \exf@config@insertproblemsbefore}
```

\exf@showproblemsout Output problem block outro:

```
1504 \newcommand{\exf@showproblemsout}{%
```

Insert hook code; close paragraph; add vertical space:

```
1505    \exf@config@insertproblemsafter%
1506    \par\exf@config@styletext\addvspace{\exf@config@skipproblemsbelow}}
```

\exf@showproblems Output problem block in buffer:

```
1507 \newcommand{\exf@showproblems}{%
```

Do nothing if buffer is empty (avoid titles):

```
1508    \ifexf@probbuf@clean\else\begingroup%
```

Execute output problem block intro:

```
1509      \exf@showproblemsin%
1510      \exf@problemstitle%
```

Source and clear buffer:

```
1511      \exf@source@buf\exf@probbuf%
1512      \exf@clear@probbuf%
```

Execute output problem block outro:

```
1513      \exf@showproblemsout%
1514    \endgroup\fi}
```

Problems Buffer Interface.

\writeproblems Open a file #1.prb for writing problems; default is present main file name:

```
1515 \newcommand{\writeproblems}[1][\jobname]{%
1516    \exf@close@probfile\exf@start@probfile{#1}}
```

\closeproblems Close problems output file (if open):

```
1517 \newcommand{\closeproblems}{\exf@close@probfile}
```

\readproblems Read problems from file `#1.prb`; default is present main file name; switch layout and add heading:

```
1518 \newcommand{\readproblems}[1][\jobname]{\exf@close@probfile%
1519   \begingroup%
1520    \exf@config@styletext\exf@config@styletextproblem%
1521    \exf@problemstitle%
1522    \input{#1\exf@config@extproblems}%
1523   \endgroup}
```

\insertproblems Show problems buffer:

```
1524 \newcommand{\insertproblems}{\exf@showproblems}
```

C.12   Subproblem Environment

exf@subproblem Define options for `subproblem` environment:

```
1525 \define@key{exf@subproblem}{points}{\exf@scanpoints\exf@subproblem@points#1++&}
1526 \define@key{exf@subproblem}{label}{\def\exf@label{#1}}
```

subproblem Define `subproblem` environment (potentially using custom name):

```
1527 \newenvironment{\exf@subproblemname}[1][]{%
```

Start with new paragraph, set text style, add vspace and step counter:

```
1528   \par{\exf@config@styletext\addvspace{\exf@config@skipsubproblemabove}}%
1529   \refstepcounter{\exf@subproblemcounter}%
```

Insert hook code:

```
1530   \exf@config@insertsubproblembefore%
```

Start inner group, mark in subproblem:

```
1531   \begingroup%
1532   \def\exf@in@subproblem{}%
```

Initialise variables, process arguments:

```
1533   \exf@init@block{\exf@config@skipsubprobleminfo}%
1534   \let\exf@subproblem@points\@undefined%
1535   \let\exf@label\@undefined%
1536   \setkeys{exf@subproblem,exf@probleminfo}{#1}%
```

Process manual label:

```
1537   \ifdefined\exf@label\label{\exf@label}\fi%
```

Remember subproblem counter for solution:

```
1538   \xdef\exf@prevsubprob{\csname the\exf@subproblemcounter\endcsname}%
1539   \ifcsname theH\exf@subproblemcounter\endcsname%
1540    \xdef\exf@prevsubprobhref{\exf@subproblemcounter.%
1541      \csname theH\exf@subproblemcounter\endcsname}%
1542   \fi%
```

Remeber points for display; disable points display if desired:

```
1543    \let\exf@subproblem@points@show\@undefined%
1544    \ifdefined\exf@subproblem@points%
1545     \let\exf@subproblem@points@show\exf@subproblem@points\fi%
1546    \exf@ifis\exf@subpointsat{off}{\let\exf@subproblem@points@show\@undefined}%
```

Add given points to problem total:

```
1547    \ifdefined\exf@subproblem@points%
1548     \global\let\exf@prevpoints\exf@subproblem@points%
1549     \exf@exptwo\exf@addtopoints\exf@problem@points@total\exf@subproblem@points%
```

Warn if no points given for present subproblem but previously:

```
1550    \else%
1551     \ifdefined\exf@problem@points@total%
1552      \PackageWarning{exframe}{no points defined for \exf@subproblemname}%
1553     \fi%
1554    \fi%
```

Display points in opening line if desired; expand points into argument and remove points:

```
1555    \exf@ifis\exf@subpointsat{start}{\exf@outpoints{\exf@append@intro}%
1556     {\exf@config@composepointspairstartsubproblem}{\exf@subproblem@points@show}}%
1557    \exf@ifis\exf@subpointsat{start*}{\exf@outpoints{\exf@prepend@intro}%
1558     {\exf@config@composepointspairstartsubproblem}{\exf@subproblem@points@show}}%
```

Insert hook code:

```
1559    \exf@config@insertsubprobleminfo%
```

Write opening line without item:

```
1560    \ifdim\exf@config@skipsubproblemitem=0pt%
1561     \exf@prepend@intro{{%
1562      \exf@config@styletitle\exf@config@styletitlesubproblem%
1563      \exf@config@composetitlesubproblem{%
1564       \csname the\exf@subproblemcounter\endcsname}}}%
```

Write item with fixed total width or item width plus space:

```
1565    \else%
1566     \ifdim\exf@config@skipsubproblemitem>0pt%
1567      \setlength\exf@addmargin{\exf@config@skipsubproblemitem}%
1568     \else%
1569      \settowidth\exf@addmargin{%
1570       \exf@config@styletitle\exf@config@styletitlesubproblem%
1571       \exf@config@composeitemsubproblem{\exf@config@countersubproblemmax}%
1572       \exf@config@composeitemsubproblemsep}%
1573     \fi%
```

Define item label:

```
1574     \def\exf@introitem{\makebox[0cm][r]{%
1575      \exf@config@styletitle\exf@config@styletitlesubproblem%
1576      \exf@config@composeitemsubproblem%
1577       {\csname the\exf@subproblemcounter\endcsname}%
1578      \exf@config@composeitemsubproblemsep}}%
1579    \fi%
```

Write points into margin if desired; expand points into argument and remove points:

```
1580    \exf@ifis\exf@subpointsat{margin}{%
```

```
1581    \exf@outpoints{\exf@prepend@def\exf@introitem}%
1582    {\exf@makepointsmargin}{\exf@subproblem@points@show}}%
```

Write out opening line:

```
1583    \exf@open@block{\exf@config@skipsubproblemtitle}%
```

Done:

```
1584    \@afterindentfalse}%
```

End environment, show points if desired:

```
1585  {\exf@ifis\exf@subpointsat{end}{\showpoints}%
```

End paragraph and environment:

```
1586    \par\exf@close@block%
```

Display solution if desired:

```
1587    \exf@ifis\exf@solutionbelow{subproblem*}{%
1588      \exf@config@insertsubproblemsolution%
1589      \exf@showsolutions{\exf@config@composetitlesolutionsingle}{}}%
```

End inner group:

```
1590    \endgroup%
```

Insert hook code, vertical skip:

```
1591    \exf@config@insertsubproblemafter%
1592    {\exf@config@styletext\addvspace{\exf@config@skipsubproblembelow}}%
```

Display solution if desired:

```
1593    \exf@ifis\exf@solutionbelow{subproblem}{%
1594      \exf@showsolutions{\exf@config@composetitlesolutionsingle}{}}%
```

Done:

```
1595    \ignorespacesafterend}
```

C.13   Solution Environment

Print Solutions.

exf@solution   Define options for `solution` environment:

```
1596 \define@key{exf@solution}{prob}{\def\exf@solprob{#1}}
1597 \define@key{exf@solution}{subprob}{\def\exf@solsubprob{#1}}
1598 \define@key{exf@solution}{problemtag}{\def\problemtag{#1}}
1599 \define@key{exf@solution}{sheettag}{\def\sheettag{#1}}
1600 \define@key{exf@solution}{href}{\def\exf@solhref{#1}}
1601 \define@key{exf@solution}{label}{\def\exf@label{#1}}
1602 \define@key{exf@solution}{points}{\exf@scanpoints\exf@solution@points#1++&}
1603 \define@key{exf@solution}{probtitle}{\def\exf@solprobtitle{#1}}
```

printsolution   Define `printsolution` environment to display a previously read `solution` environment; this
works analogously to `problem` and `subproblem`:

```
1604 \newenvironment{printsolution}[1]{%
```

Start new paragraph, add vertical space:

```
1605    \par{\exf@config@styletext\addvspace{\exf@config@skipsolutionabove}}%
```

Insert hook code:

```
1606    \exf@config@insertsolutionbefore%
```

Use equation counter for solutions:

```
1607    \ifexf@style@solutionequation%
1608    \exf@eqsav\value{equation}\relax%
1609    \setcounter{equation}{\value{exf@solutionequation}}%
1610    \let\theequation\theexf@solutionequation%
1611    \let\theHequation\theHexf@solutionequation%
1612    \fi%
```

Start a group, initialise variables, process arguments:

```
1613    \begingroup%
1614    \def\exf@in@solution{}%
1615    \def\exf@solprob{}%
1616    \def\exf@solsubprob{}%
1617    \let\exf@label\@undefined%
1618    \let\exf@solution@points\@undefined%
1619    \let\exf@solution@points@total\@undefined%
1620    \def\exf@solhref{}%
1621    \exf@init@block{\exf@config@skipsolutioninfo}%
1622    \setkeys{exf@solution,exf@probleminfo}{#1}%
```

Set solution counter to reflect associated problem:

```
1623    \exf@csdo\def{the\exf@solutioncounter}%
1624    {\exf@config@composeitemsolutionlabel{\exf@solprob}{\exf@solsubprob}}%
1625    \refstepcounter{\exf@solutioncounter}%
```

Set label:

```
1626    \ifdefined\exf@label\label{\exf@label}\fi%
```

Remeber points for display; disable points display if desired:

```
1627    \let\exf@solution@points@show\@undefined%
1628    \ifdefined\exf@solution@points%
1629    \let\exf@solution@points@show\exf@solution@points\fi%
1630    \exf@ifis\exf@solpointsat{off}{\let\exf@solution@points@show\@undefined}%
```

Display points in opening line if desired; expand points into argument and remove points:

```
1631    \exf@ifis\exf@solpointsat{start}{\exf@outpoints{\exf@append@intro}%
1632    {\exf@config@composepointspairstartsolution}{\exf@solution@points@show}}%
1633    \exf@ifis\exf@solpointsat{start*}{\exf@outpoints{\exf@prepend@intro}%
1634    {\exf@config@composepointspairstartsolution}{\exf@solution@points@show}}%
```

Insert hook code, set solution body style:

```
1635    \exf@config@insertsolutioninfo%
1636    \exf@config@styletext\exf@config@styletextsolution%
```

Determine solution for problem or subproblem:

```
1637    \ifx\exf@solsubprob\exf@empty%
1638    \let\exf@tmp\exf@config@skipsolutionitem%
1639    \else%
```

```
1640    \let\exf@tmp\exf@config@skipsolutionitemsub%
1641  \fi%
```

Write title without item:

```
1642  \ifdim\exf@tmp=0pt%
1643    \protected@edef\exf@solution@title{%
1644     \exf@composetitle{\exf@solprob}{\exf@solsubprob}}%
1645    \ifx\exf@solution@title\exf@empty\else%
1646     \exf@prepend@intro{{%
1647      \exf@config@styletitle\exf@config@styletitlesolution%
1648      \ifexf@solutionhref\exf@href{\exf@solhref}%
1649        {\exf@solution@title}\else\exf@solution@title\fi}}%
1650    \fi%
```

Write item with fixed total width or item width plus space:

```
1651  \else%
1652  \ifdim\exf@tmp>0pt%
1653    \setlength\exf@addmargin{\exf@tmp}%
1654  \else%
1655    \settowidth\exf@addmargin{%
1656     \exf@config@styletitle\exf@config@styletitlesolution%
1657     \ifx\exf@solsubprob\exf@empty%
1658      \exf@config@composeitemsolution{\exf@config@counterproblemmax}%
1659        {\exf@config@countersubproblemmax}%
1660     \else%
1661      \exf@config@composeitemsolutionsub{\exf@config@counterproblemmax}%
1662        {\exf@config@countersubproblemmax}%
1663     \fi\exf@config@composeitemsolutionsep}%
1664  \fi%
```

Set item label depending on problem or subproblem:

```
1665  \ifx\exf@solsubprob\exf@empty%
1666    \protected@edef\exf@solution@item%
1667     {\exf@config@composeitemsolution{\exf@solprob}{\exf@empty}}%
1668  \else%
1669    \protected@edef\exf@solution@item%
1670     {\exf@config@composeitemsolutionsub{\exf@solprob}{\exf@solsubprob}}%
1671  \fi%
```

Define item label:

```
1672  \def\exf@introitem{\makebox[0cm][r]{%
1673    \exf@config@styletitle\exf@config@styletitlesubproblem%
1674    \ifexf@solutionhref\exf@href{\exf@solhref}{\exf@solution@item}%
1675    \else\exf@solution@item\fi%
1676    \exf@config@composeitemproblemsep}}%
1677  \fi%
```

Write points into margin if desired; expand points into argument and remove points:

```
1678  \exf@ifis\exf@solpointsat{margin}{%
1679    \exf@outpoints{\exf@prepend@def\exf@introitem}%
1680    {\exf@makepointsmargin}{\exf@solution@points@show}}%
```

Write out opening line:

```
1681  \exf@open@block{\exf@config@skipsolutiontitle}%
```

Done:

```
1682    \@afterindentfalse}%
```

End environment, show points if desired, perform sanity check:

```
1683  {\exf@ifis\exf@solpointsat{end}{\showpoints}%
1684   \exf@warnmismatch{\exf@solutionname}{\exf@solutioncounter}%
1685     {\exf@solution@points@total}{\exf@solution@points}%
```

End paragraph and envionment:

```
1686   \par\exf@close@block%
```

Restore original equation counter:

```
1687   \ifexf@style@solutionequation%
1688    \setcounter{exf@solutionequation}{\value{equation}}%
1689    \setcounter{equation}{\exf@eqsav}%
1690   \fi%
```

End inner group:

```
1691   \endgroup%
```

Vertical skip, insert hook code:

```
1692   {\exf@config@styletext\addvspace{\exf@config@skipsolutionbelow}}%
1693   \exf@config@insertsolutionafter%
```

Done:

```
1694   \ignorespacesafterend}
```

\solutionssection Define a section for a problem within a solution block:

```
1695 \newcommand{\solutionssection}[1]{\begingroup%
```

Initialise variables, process arguments:

```
1696   \def\exf@solprob{}%
1697   \def\exf@solsubprob{}%
1698   \def\exf@solprobtitle{}%
1699   \let\exf@label\@undefined%
1700   \let\exf@solhref\@undefined%
1701   \setkeys{exf@solution}{#1}%
```

Select title (and table of contents entry) corresponding to multiple problems vs. single problem:

```
1702   \let\exf@composetitle\exf@config@composetitlesolutionsproblemmulti%
1703   \exf@ifis\exf@solutionbelow{problem}{\let\exf@composetitle%
1704     \exf@config@composetitlesolutionsproblemsingle}%
1705   \exf@ifis\exf@solutionbelow{problem*}{\let\exf@composetitle%
1706     \exf@config@composetitlesolutionsproblemsingle}%
1707   \def\exf@solutionstoc{\exf@addcontentsline{\exf@config@toclevelsolution}%
1708     {\exf@config@composetocsolution{\exf@solprob}{\exf@solprobtitle}}}%
```

Write section line:

```
1709   \addvspace{\exf@config@skipsolutionsproblemabove}%
1710   \exf@solutionssection{\exf@config@styletitlesolutionsproblem}%
1711     {\exf@composetitle{\exf@solprob}{\exf@solprobtitle}}%
1712     {\exf@config@skipsolutionsproblemtitle}%
1713     {\exf@solutionstoc}{\exf@label}{\exf@solhref}%
1714   \endgroup}
```

Read Solution Environment.

If this is the first solution within a new section, write section heading to buffer:

```
1715 \newcommand{\exf@process@solnewsec}{%
1716  \ifdefined\exf@problem@solnewsec%
1717  \def\exf@probarg{\ifdefined\exf@prevprob prob={\exf@prevprob}\fi%
1718   \ifdefined\exf@prevprobtitle,probtitle={\exf@prevprobtitle}\fi%
1719   \ifdefined\exf@prevprobhref,href={\exf@prevprobhref}\fi%
1720   \ifdefined\exf@sollabel,label={\exf@sollabel}\fi}%
1721  \exf@ifis\exf@solutionbelow{here}{\let\exf@probarg\@undefined}%
1722  \exf@ifis\exf@solutionbelow{subproblem}{\let\exf@probarg\@undefined}%
1723  \exf@ifis\exf@solutionbelow{subproblem*}{\let\exf@probarg\@undefined}%
1724  \ifdefined\exf@probarg%
1725   \ifexf@lineno\exf@addline\exf@solbuf{\exf@linesep}%
1726    \exf@addline\exf@solbuf{\exf@lineno}\fi%
1727   \exf@addline\exf@solbuf{\@backslashchar solutionssection{\exf@probarg}}%
1728   \exf@addline\exf@solbuf{}%
1729  \fi%
1730  \global\let\exf@problem@solnewsec\@undefined%
1731  \fi}%
```

Declare additional arguments to `printsolution` to describe corresponding problem and tags:

```
1732 \newcommand{\exf@generate@solprobarg}{%
1733  \edef\exf@solprobarg{%
1734   \ifdefined\exf@prevprob prob={\exf@prevprob},\fi%
1735   \ifdefined\exf@prevsubprob subprob={\exf@prevsubprob},%
1736    \ifdefined\exf@prevsubprobhref href={\exf@prevsubprobhref},\fi%
1737   \else%
1738    \ifdefined\exf@prevprobhref href={\exf@prevprobhref},\fi%
1739   \fi%
1740   \ifdefined\exf@prevpoints points=%
1741    {\expandafter\exf@formatpoints\exf@prevpoints},\fi%
1742   \ifdefined\sheettag sheettag={\sheettag},\fi%
1743   \ifdefined\problemtag problemtag={\problemtag},\fi}%
```

Clean up:

```
1744  \global\let\exf@prevsubprob\@undefined%
1745  \global\let\exf@prevsubprobhref\@undefined%
1746  \global\let\exf@prevpoints\@undefined}%
```

Define direct output version of `solution` environment; pass on to `printsolution` environment:

```
1747 \newenvironment{exf@solution@direct}[1][]%
1748 {\showpoints%
1749  \global\let\exf@problem@solnewsec\@undefined%
1750  \exf@generate@solprobarg%
1751  \exf@showsolutionsin%
1752  \let\exf@composetitle\exf@config@composetitlesolutionsingle%
1753  \exf@exptwo\printsolution{\exf@solprobarg#1}}%
1754 {\endprintsolution%
1755  \exf@showsolutionsout%
1756  \ignorespacesafterend}%
```

Define scan version of `solution` environment; use `\exf@scanblock` to properly parse op-

tional argument and pass on to `exf@scansolution`:

```
1757 \newenvironment{exf@solution@scan}%
1758  {\exf@scanblock{\exf@scansolution}}{\endexf@scansolution}%
1759 \newenvironment{exf@scansolution}[2]{%
```

If solution is to be displayed immediately, make sure to display points first; insert solution section heading in buffer; compose additional arguments to `printsolution`:

```
1760  \exf@ifis\exf@solutionbelow{here}{\showpoints}%
1761  \exf@process@solnewsec%
1762  \exf@generate@solprobarg%
```

Write separator and `printsolution` environment to buffer:

```
1763  \ifexf@lineno\exf@addline\exf@solbuf{\exf@linesep}%
1764   \exf@addline\exf@solbuf{\exf@lineno}\fi%
1765  \exf@addline\exf@solbuf%
1766   {\@backslashchar begin{printsolution}{\exf@solprobarg#1}}%
```

Start verbatim processing:

```
1767  \def\exf@verbatim@process{\exf@append@buf\exf@solbuf}%
1768  \exf@verbatim#2}%
```

End verbatim processing; close `printsolution` environment:

```
1769  {\exf@endverbatim%
1770  \exf@addline\exf@solbuf{\@backslashchar end{printsolution}}%
1771  \global\exf@solbuf@cleanfalse%
```

Write buffer to file if output file open:

```
1772  \ifexf@solfile@open%
1773   \exf@write@buf\exf@solfile\exf@solbuf%
1774   \exf@clear@solbuf%
1775  \fi%
```

Drop buffer if solutions not to be displayed:

```
1776  \ifsolutions\else\exf@clear@solbuf\fi%
```

Display solution immediately in various cases:

```
1777  \exf@ifis\exf@solutionbelow{here}{\exf@showsolutions%
1778   {\exf@config@composetitlesolutionsingle}{}}%
1779  \ifdefined\exf@in@subproblem\else%
1780   \exf@ifis\exf@solutionbelow{subproblem}{%
1781    \exf@showsolutions{\exf@config@composetitlesolutionsingle}{}}%
1782   \exf@ifis\exf@solutionbelow{subproblem*}{%
1783    \exf@showsolutions{\exf@config@composetitlesolutionsingle}{}}\fi%
1784  \ifdefined\exf@in@problem\else%
1785   \exf@ifis\exf@solutionbelow{problem}{%
1786    \exf@showsolutions{\exf@config@composetitlesolutionmulti}{}}%
1787   \exf@ifis\exf@solutionbelow{problem*}{%
1788    \exf@showsolutions{\exf@config@composetitlesolutionmulti}{}}\fi%
```

Done:

```
1789  \ignorespacesafterend}
```

solution Define `solution` environment (potentially using custom name) to choose between direct and buffered version:

```
1790 \newenvironment{\exf@solutionname}%
1791 {\ifexf@solutionbuf\let\exf@tmp\exf@solution@scan%
1792  \else\let\exf@tmp\exf@solution@direct\fi%
1793  \exf@tmp}%
1794 {\ifexf@solutionbuf\let\exf@tmp\endexf@solution@scan%
1795  \else\let\exf@tmp\endexf@solution@direct\fi%
1796  \exf@tmp}
```

C.14   Solution Blocks

Solution Block Handling.

<div style="margin-left:-3em">xf@solutionssection</div> Output solution block section:

```
1797 \newcommand{\exf@solutionssection}[6]{%
```

Check whether title is empty:

```
1798  \protected@edef\exf@solutionstitleexp{#2}%
1799  \ifx\exf@solutionstitleexp\exf@empty\else%
```

Define a label:

```
1800   \ifdefined#5%
1801   \exf@csdo\def{the\exf@solutioncounter}%
1802    {\exf@config@composeitemsolutionlabel{\exf@solprob}{\exf@solsubprob}}%
1803   \refstepcounter{\exf@solutioncounter}\label{#5}%
1804   \fi%
```

Output section line:

```
1805   \exf@section{#3}{\exf@config@styletitle\exf@config@styletitlesolution#1%
1806    \ifexf@solutionhref\exf@href{#6}{\exf@solutionstitleexp}%
1807    \else\exf@solutionstitleexp\fi}#4%
1808   \fi}
```

\exf@solutionstitle   Compose the title for a solution block:

```
1809 \newcommand{\exf@solutionstitle}{\exf@solutionssection%
1810  {\exf@config@styletitlesolutions}{%
1811   \exf@config@composetitlesolutions}{\exf@config@skipsolutionstitle}%
1812  {\exf@addcontentsline{\exf@config@toclevelsolutions}%
1813   {\exf@config@composetocsolutions}}{\@undefined}{\@undefined}}
```

exf@showsolutionsin   Output solution block intro:

```
1814 \newcommand{\exf@showsolutionsin}{%
```

Set solution body style; add vertical space; insert hook code:

```
1815  \par\exf@config@styletext\addvspace{\exf@config@skipsolutionsabove}%
1816  \exf@config@styletextsolution%
1817  \exf@config@insertsolutionsbefore}
```

xf@showsolutionsout   Output solution block outro:

```
1818 \newcommand{\exf@showsolutionsout}{%
```

Insert hook code; close paragraph; add vertical space:

```
1819  \exf@config@insertsolutionsafter%
1820  \par\exf@config@styletext\addvspace{\exf@config@skipsolutionsbelow}}
```

\exf@showsolutions Output solution block in buffer:

```
1821 \newcommand{\exf@showsolutions}[2]{%
```

Do nothing if buffer is empty (avoid titles):

```
1822   \ifexf@solbuf@clean\else\begingroup%
```

Execute output solution block intro:

```
1823     \exf@showsolutionsin%
1824     \let\exf@composetitle#1%
1825     #2%
```

Source and clear buffer:

```
1826     \exf@source@buf\exf@solbuf%
1827     \exf@clear@solbuf%
```

Execute output solution block outro:

```
1828     \exf@showsolutionsout%
1829   \endgroup\fi}
```

Solutions Buffer Interface.

\writesolutions Open a file `#1.sol` for writing solutions; default is present main file name:

```
1830 \newcommand{\writesolutions}[1][\jobname]{%
1831   \exf@close@solfile\exf@start@solfile{#1}}
```

\closesolutions Close solutions output file (if open):

```
1832 \newcommand{\closesolutions}{\exf@close@solfile}
```

\readsolutions Read solutions from file `#1.sol`; default is present main file name; switch layout and add heading:

```
1833 \newcommand{\readsolutions}[1][\jobname]{\exf@close@solfile%
1834   \ifsolutions\begingroup%
1835     \exf@config@styletext\exf@config@styletextsolution%
1836     \let\exf@composetitle\exf@config@composetitlesolutionmulti%
1837     \exf@solutionstitle%
1838     \input{#1\exf@config@extsolutions}%
1839   \endgroup\fi}
```

\insertsolutions Show solutions buffer:

```
1840 \newcommand{\insertsolutions}{\exf@showsolutions%
1841   {\exf@config@composetitlesolutionmulti}{\exf@solutionstitle}}
```