

# The `exframe` Package

## Source Code Documentation

Niklas Beisert

Institut für Theoretische Physik  
Eidgenössische Technische Hochschule Zürich  
Wolfgang-Pauli-Strasse 27, 8093 Zürich, Switzerland  
`nbeisert@itp.phys.ethz.ch`

v3.5.3: 2026-03-23

<https://ctan.org/pkg/exframe>  
<https://github.com/nbeisert/latex-pkg-nb>

### Abstract

`exframe` is a  $\text{\LaTeX} 2_{\epsilon}$  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

<b>1</b>	<b>Standalone Sample</b>	<b>2</b>
<b>2</b>	<b>Multipart Sample</b>	<b>8</b>
2.1	Main File . . . . .	8
2.2	Sheet File . . . . .	13
2.3	Individual Problem Files . . . . .	15
2.4	Make Scripts . . . . .	16
<b>3</b>	<b>Implementation</b>	<b>19</b>
3.1	General Definitions . . . . .	19
3.2	Package Setup . . . . .	21
3.3	Configuration . . . . .	23
3.4	Styles . . . . .	32
3.5	Metadata . . . . .	35
3.6	Counters . . . . .	39
3.7	Buffers . . . . .	40
3.8	Points . . . . .	43
3.9	Tag Lists . . . . .	47
3.10	Sheet Environment . . . . .	48
3.11	Problem Environment . . . . .	50
3.12	Problem Blocks . . . . .	55

3.13 Subproblem Environment . . . . .	56
3.14 Solution Environment . . . . .	60
3.15 Solution Blocks . . . . .	66
3.16 Interaction with metastr . . . . .	67

## 1 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 `amstext`, `hyperref` packages:

```
8 \usepackage{amstext}
9 \usepackage{hyperref}
```

May include `metastr` package; below code adjusts to whether or not present:

```
10 \PassOptionsToPackage{loadlang=en|de}{metastr}
11 \PassOptionsToPackage{course=true}{metastr}
12 %%\usepackage{metastr}
13 %%\metasetlang{de}
```

Include `exframe` package:

```
14 \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:

```
15 \exercisestart{solutions=false}
16 %%\exercisestart{solutions=true}
```

Automatically put labels for (sub)problems:

```
17 \exercisestart{autolabelproblem=true}
```

**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:

```

18 \exercisestyle{plainheader}
19 \exerciseconfig{composeheaderbelowright}{\getsheetdata{points}}%

```

Make the delimiter ‘)’ part of the subproblem item rather than the subproblem label:

```

20 \exercisestyle{subproblemdelitem}

```

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:

```

21 \exerciseconfig{countersheet}{\Roman{sheet}}
22 \exerciseconfig{countersubproblem}{\roman{subproblem}}
23 \exerciseconfig{countersubproblemmax}{vii}

```

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

```

24 \exerciseconfig{insertsubprobleminfo}{%
25 \switchpoints{}{\addprobleminfo*{%
26 \hspace{-\getexerciseconfig{skipsubprobleminfo}*}}}%
27 {}{\getsubproblempoints{}}

```

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

```

28 \ifdefined\metaset
29 \metasetterm[en]{sheet}{Exercise Sheet}
30 \metasetterm[en]{sheets}{Exercise Sheets}
31 \metasetterm[de]{sheet}{\ "Ubungsblatt}
32 \metasetterm[de]{sheets}{\ "Ubungsbl\ "atter}
33 \else
34 \exerciseconfig{termsheet}{\ "Ubungsblatt}
35 \exerciseconfig{termsheets}{\ "Ubungsbl\ "atter}
36 \fi

```

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

```

37 \exercisestyle{problempointsat=margin}
38 \reversemarginpar
39 \exerciseconfig{composepointsmargin}[1]{#1p.}
40 \exerciseconfig{composepointspairmargin}[2]{%
41 \ifdim#2pt=0pt#1p.%
42 \else\ifdim#1pt=0pt+#2p.%
43 \else#1+#2p.%
44 \fi\fi}

```

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

```

45 \exerciseconfig{styletitle}{\sffamily\bfseries}

```

Add a significant amount of space below problems:

```

46 \exerciseconfig{skipproblembelow}{1.5cm}

```

Display half points as fractions:

```

47 \exercisestyle{fracpoints}

```

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

```

48 \exercisestyle{solutionbelow=problem}

```

Indent solutions for subproblems:

```
49 \exerciseconfig{skipsolutionitemsub}{-1pt}
```

Separate solutions by horizontal lines (extended style):

```
50 \exercisestyle{solutionsep}
```

Write metadata for sheet:

```
51 \exercisecsetup{pdfdata=sheet}
```

Set title and author for pdf metadata; `title` is course plus material or sheet title; `author` is instructor or sheet author plus institution:

```
52 \ifdefined\metaset
53 \metaset{sep}{subtitle}{, }
54 \metaset{subtitle}{\ifsolutions\metatranslate[#1]{solutions} \fi%
55 \metaif[use]{sheettitle}
56 {\metapick[#1]{sheettitle}}
57 {\metapick[#1]{material}}}
58 \metaset{author}{\exerciseifempty{\getsheetdata{author}}%
59 {\metapick[#1]{instructor}}{\metapick[#1]{sheetauthor}},
60 \metapick[#1]{institution}}
61 \else
62 \exerciseconfig{composemetasheet}[2]{\getexercisedata{course},
63 \ifsolutions\getexerciseconfig{termsolutions} \fi%
64 \exerciseifempty[#2]{\getexerciseconfig{termsheet} #1}{#2}}
65 \exercisedata{title=%
66 {\getexercisedata{course},
67 \ifsolutions\getexercisedata{solutions} \fi%
68 \getexercisedata{material}}}
69 \exercisedata{author=%
70 {\getexercisedata{instructor}, \getexercisedata{institution}}}
71 \fi
```

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

```
72 \ifdefined\metaset
73 \metaset{institution}{Katharinen-Volksschule}
74 \metaset{de}{course}{Mathematik}
75 \metaset{en}{course}{Mathematics}
76 \metaset{instructor}{J.\ G.\ B\"uttner}
77 \metaset{period}{ca.\ 1786}
78 \metaset{de}{material}{\"Ubungsaufgaben}
79 \metaset{en}{material}{Exercise Problems}
80 \else
81 \exercisedata{institution={Katharinen-Volksschule}}
82 \exercisedata{course={Mathematik}}
83 \exercisedata{instructor={J.\ G.\ B\"uttner}}
84 \exercisedata{period={ca.\ 1786}}
85 \exercisedata{material={\"Ubungsaufgaben}}
86 \fi
```

**Body.**

```
87 \begin{document}
```

Start sheet number 5:

```
88 \begin{sheet}[number=5,label={sheet5}]
```

Start a problem with a title:

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

Summary of points:

```
90 \exerciseloopstr{\getsubproblemlist{}}{c}%
91 \hfill\begin{tabular}{c|\exerciseloopret|c}
92 \exerciseloop{\getsubproblemlist{}}
93  {\&\ref{\getexerciseconfig{labelsubproblem}{#1}}}
94  &\ref{prob:\problemtag}\hline
95  \getexerciseconfig{termpoints}
96  \exerciseloop{\getsubproblemlist{}}{\&\extractpoints{\getsubproblempoints{#1}}}
97  \extractpoints{\getproblempoints{}}
98  \
99  extra
100 \exerciseloop{\getsubproblemlist{}}{\&\extractpoints*{\getsubproblempoints{#1}}}
101  \extractpoints*{\getproblempoints{}}
102 \end{tabular}
```

Some introduction to the problem:

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

A subproblem with a local label:

```
104 \begin{subproblem}[points=2,difficulty=simple,label={\problemtag-simplesum}]
105 Compute the sum
106 \showpoints
107 \begin{equation}
108 1+2+3.
109 \end{equation}
```

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

```
110 \begin{solution}
111 The result is
112 \begin{equation}
113 1+2+3=6.
114 \end{equation}
115 \end{solution}
```

End subproblem:

```
116 \end{subproblem}
```

Another subproblem:

```
117 \begin{subproblem}[points=97+0.5,difficulty=lengthy]
118 Compute the sum
119 \begin{equation}
120 1+2+3+\ldots+98+99+100.
121 \end{equation}
122 Keep calm and calculate!
123 %%That ought to keep him occupied for a while
124 \end{subproblem}
```

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

```
125 \begin{solution}[author={C.\ F.\ Gau\ss}]
126 We use the result  $1+2+3=6$  from part \ref{\problemtag-simplesum})
127 to jumpstart the calculation. The remaining sums yield
```

```

128 \awardpoints*[1 for each remaining sum]{97}
129 \begin{equation}
130 6+4+5+\ldots+99+100=5050.
131 \end{equation}
132 Alternatively the summands can be grouped into pairs as follows:
133 \begin{align}
134 1+100&=101, \\
135 2+99&=101, \\
136 3+98&=101, \\
137 \dots &\nonumber \\
138 50+51&=101.
139 \end{align}
140 These amount to 50 times the same number 101.
141 Therefore the sum equals
142 \begin{equation}
143 1+2+\ldots+99+100=50\cdot 101=5050.
144 \end{equation}
145 \textit{Liggett se!} \awardpoints{97+0.5}
146 \end{solution}

```

Some text between subproblems:

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

Final subproblem; this one is optional:

```

148 \begin{subproblem}[optional={optional},
149   difficulty={requires inspiration},points={+3.5}]
150 Compute the series
151 \showpoints
152 \begin{equation}
153 1+2+3+\ldots
154 \end{equation}

```

Provide a solution:

```

155 \begin{solution}
156 The series is divergent, so the result is  $\infty$  \awardpoints{+1}.
157 \par
158 However, after subtracting the divergent part,
159 the result clearly is
160 \begin{equation}
161 \zeta(-1)=-\frac{1}{12},
162 \end{equation}
163 where the zeta-function  $\zeta(s)$  is defined by
164 \begin{equation}
165 \zeta(s):=\sum_{k=1}^{\infty} \frac{1}{k^s},
166 \end{equation}
167 This definition holds only for  $s>1$  where the sum is convergent,
168 but one can continue the complex analytic function to  $s<0$ 
169 \awardpoints{+1.5}.
170 \par
171 Another way of understanding the result
172 is to use the indefinite summation formula
173 for arbitrary exponent  $s$  in the summand
174 (which also follows from the Euler--MacLaurin formula)
175 \begin{equation}
176 \sum_n n^s
177 = \frac{n^{s+1}}{s+1}
178 - \sum_{j=0}^{\infty} \frac{\zeta(j-s)}{(s-j)!} n^j
179 = \dots - \zeta(-s) n^0.

```

```

180 \end{equation}
181 Curiously, the constant term with  $j=0$  is just the desired result
182 but with the wrong sign
183 (in fact, the constant term of an indefinite sum is ambiguous;
184 for the claim we merely set  $j=0$ 
185 in the expression which holds for others values of  $j$ )
186 \awardpoints{+0.5}.
187 In order to understand the sign,
188 we propose that the above formula describes the regularised result
189 for the sum with limits  $+\infty$  and  $n$ 
190 \begin{equation}
191 \sum_{k=-\infty}^n k^s
192 \simeq \frac{n^{s+1}}{s+1}
193 -\sum_{j=0}^s \frac{\zeta(j-s)\,s!}{(s-j)!\,j!}\,n^j.
194 \end{equation}
195 Then we flip the summation limits of the desired sum
196 to bring it into the above form
197 \awardpoints{+0.5}
198 \begin{equation}
199 \sum_{k=1}^{\infty} k^s
200 = -\sum_{k=-\infty}^0 k^s
201 \simeq \zeta(-s).
202 \end{equation}
203 \end{solution}

```

End subproblem:

```
204 \end{subproblem}
```

End problem:

```
205 \end{problem}
```

Another problem; this one is untitled:

```

206 \begin{problem}[points=1, difficulty=insane]
207 Show that the equation
208 \begin{equation}
209 a^3+b^3=c^3
210 \end{equation}
211 has no positive integer solutions.
212 \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):

```

213 \begin{solution}
214 \normalmarginpar
215 This is beyond the scope of this example.
216 \marginpar{\footnotesize\raggedright does not fit here.\par}
217 \end{solution}

```

Summary of points per problem and per subproblem for grading:

```

218 \ifsolutions\else
219 \textbf{Grading:}\par
220 \exerciseloopstr{\getproblemlist}{|c}
221 \begin{tabular}{|c|\exerciseloopret||c|}\hline
222 \getexerciseconfig{termsheet} \ref{sheet5}
223 \exerciseloop{\getproblemlist{*}}
224 {\&\ref{\getexerciseconfig{labelproblem}{#1}}}
225 &total

```

```

226 \\hline
227 value
228 \exerciseloop{\getproblemlist{*}}
229  {\&\extractpoints{\getproblempoints{#1}}}%
230 &\extractpoints{\getsheetpoints{}}
231 \\hline
232 \exerciseloop{\getproblemlist{*}}{&}
233 &\\hline
234 \end{tabular}\quad
235 \exerciseloop{\getproblemlist{*}}{
236  \exerciseloopstr{\getsubproblemlist{#1}}{|c|}
237  \ifnum\value{exerciseloop}>0\relax
238  \begin{tabular}{|c|\exerciseloopret|c|}\hline
239  \getexerciseconfig{termproblem} \ref{\getexerciseconfig{labelproblem}{#1}}
240  \exerciseloop{\getsubproblemlist{#1}}
241  {\&\ref{\getexerciseconfig{labelsubproblem}{##1}}}
242  &total
243  \\hline
244  value
245  \exerciseloop{\getsubproblemlist{#1}}
246  {\&\extractpoints{\getsubproblempoints{##1}}}%
247  &\extractpoints{\getproblempoints{#1}}
248  \\hline
249  \exerciseloop{\getsubproblemlist{#1}}{&}
250  &\\hline
251  \end{tabular}\quad
252  \fi
253 }
254 \fi

```

End sheet:

```
255 \end{sheet}
```

End of document body:

```
256 \end{document}
```

## 2 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.

### 2.1 Main File

The main source file is called `exframe-ser.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:

```

257 \input{childdoc.def}
258 \childdocmain{}

```

**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:

```
259 \ifchilddoc
260 \providecommand{\printsol}{y}
261 \else
262 \providecommand{\printsol}{n}
263 \fi
264 \providecommand{\draftver}{y}
265 \newif\ifdraft\if\draftver y\drafttrue\else\draftfalse\fi
```

**Preamble.** Standard document class:

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

267 % \ctanpkg{graphicx} package to display license logo:
268 \RequirePackage{graphicx}
```

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

```
269 \PassOptionsToPackage{bookmarks=\ifchilddoc false\else true\fi}{hyperref}
270 \PassOptionsToPackage{bookmarksopen=true}{hyperref}
271 \RequirePackage{hyperref}
```

**metastr Package.** Use the metastr package to handle metadata and copyright information; disable usage of hyperxmp package if not installed; write auxiliary pdf metadata and rights information:

```
272 \ifdraft
273 \PassOptionsToPackage{draft}{metastr}
274 \fi
275 \IfFileExists{hyperxmp.sty}{\PassOptionsToPackage{hyperxmp=false}{metastr}}
276 \RequirePackage{course}{metastr}
277 \metaset[aux]{writepdf}{}
278 \metaset[rights]{writepdf}{}
```

**exframe Package.** Invoke exframe with extended data and styles:

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

Set solutions switch and declare two-sided layout only if no solutions are printed; do not use solution buffer in draft mode for easier identification of potential compile errors:

```
280 \if\printsol n
281 \exercisestop{solutions=false}
282 \exercisestop{twoside=true}
283 \else
284 \exercisestop{solutions=true}
285 \exercisestop{twoside=false}
286 \ifdraft
287 \exercisestop{solutionbuf=false}
288 \fi
289 \fi
```

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

```
290 %\if\printsol n\else\showproblem\info{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:

```
291 \exercisestyle{autolabelproblem}
292 \exercisestyle{contents,solutionsep}
293 \ifchilddocmanual\else
294 \exercisestyle{pagebysheet,problemby sheet,equationbysheet,sheetequation}
295 \fi
296 \exercisestyle{solutionbelow={problem}}
297 \ifchilddoc\ifchilddocmanual\else\exercisestyle{pdfdata=sheet}\fi\fi
```

**Language.** Redefine some terms:

```
298 \metasetlang{en-GB}
299
300 \metasetterm[en]{sheet}{sheet}
301 \metasetterm[en]{sheets}{sample sheets}
302 \metasetterm[en]{solution}{Solution}
303 \metasetterm[en]{solutions}{solutions}
```

**Layout Definitions.** Set page dimensions and layout:

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

Remove paragraph indentation:

```
306 \setlength\parindent{0pt}
307 \setlength\parskip{\smallskipamount}
```

Show overfull lines:

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

Define turn page over mark; hide when printing solutions:

```
309 \newcommand{\turnover}{\ifsolutions\else\vfill%
310 \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:

```
311 \exercisestyle{plainheader}
312 \exerciseconfig{composeheaderbelowright}
313 {\sheetdataempty{editdate}}{\version: \getsheetdata{editdate}}
314 \ifchilddoc\ifsolutions\else
315 \exerciseconfig{composeheaderbelowright}
316 {\sheetdataempty{due}}{\due: \getsheetdata{due}}
317 \fi\fi
```

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

```
318 \def\lorem{Lorem ipsum dolor sit amet, consectetur adipiscing elit,
```

```

319 sed eiusmod tempor incididunt ut labore et dolore magna aliqua.
320 Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
321 nisi ut aliquid ex ea commodi consequat.
322 Quis aute iure reprehenderit in voluptate velit esse
323 cillum dolore eu fugiat nulla pariatur.
324 Excepteur sint obcaecat cupiditat non proident,
325 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:

```

326 \IfFileExists{mpostinl.sty}{\RequirePackage{mpostinl}}{}
327 \ifdefined\mpostsetup

```

Setup `mpostinl`. Use checksums to invoke metapost only when figures change. Process all figures individually and immediately when in draft mode for child documents (avoids a second compilation pass). Number figures within sheet to provide a stable numbering upon insertion/deletion of new figures or partial compilation:

```

328 \mpostsetup{checksum}
329 \ifchilddoc\ifdraft\mpostsetup{now,nowall}\fi\fi
330 \ifchilddocmanual\else\mpostsetup{numberwithin={sheet}}\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), `pensize` changes the size of the pen, `fillshape` fills and outlines a shape:

```

331 \mpostsetup{globaldef=true}
332 \begin{mposttex}
333 \def\figure{figure}
334 \end{mposttex}
335 \begin{mpostdef}
336 path paths[];
337 newinternal numeric xu;
338 xu:=1cm;
339 def pensize(expr s)=withpen pencircle scaled s enddef;
340 def fillshape(expr p,c)=
341   fill p withcolor c;
342   draw p
343 enddef;
344 \end{mpostdef}
345 \mpostsetup{globaldef=false}

```

Close optional `mpostinl` processing:

```

346 \fi

```

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

```

347 \metaset{course}{exframe package samples}
348 \metaset{instructor}{N.\ Beisert}
349 \metaset{author}{Niklas Beisert, \metapick[#1]{institution}}
350 \metaset{institution}{exframe academy}
351 \metaset{period}{spring 2019}
352 \metaset{copyrightdate}{2019--2026}

```

For child documents declare part of:

```

353 \ifchilddoc
354 \metaset{partof}{\metatranslate[#1]{sheets} \metapick[#1]{course}}
355 \fi

```

Assemble some entries from given data:

```

356 \metaset{sep}{draft}{ -- }
357 \metaset{sep}{subtitle}{, }
358 \metaset{material}{\ifsolutions\metatranslate[#1]{solutions} \fi%
359 \metatranslate[#1]{sheets}}
360 \exerciseconfig{composetitlesheet}[2]{\exerciseifempty{#2}%
361 {\ifsolutions\metaterm{solutions}\else\metaterm{sheet}\fi \ #1}%
362 {\ifsolutions\metaterm{solutions} \fi #2}}
363 \metaset{sheettitle}{\ifsolutions\metatranslate[#1]{solutions} \fi%
364 \exerciseifempty{\getsheetdata{rawtitle}}%
365 {\metatranslate[#1]{sheet} \thesheet}%
366 {\getsheetdata{rawtitle}}}
367 \metaset{subject}{Lecture Series,
368 \metapick[#1]{institution}, \metapick[#1]{period}}

```

**License.** It is good practice to specify a copyright line and a license for the document:

```

369 \metaset{copyrightowner}{\metapick[#1]{author}}
370 \ifchilddoc
371 \metacopyright{doc}
372 \else
373 \metacopyright{doc-parts}
374 \fi
375 \metaset{licenseprovider}{of \metapick[#1]{institution}}
376 \metalicense{consent}

```

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

```

377 \ifsolutions\else\ifdraft\else
378 \metalicensecc{by-sa}
379 \fi\fi

```

**Body.** Start document body:

```

380 \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}`:

```

381 \def\tmp{}
382 \ifchilddocmanual
383 \def\tmp{\end{document}}
384 \input{\childdocname}
385 \fi\tmp

```

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

```

386 \setcounter{section}{-1}
387 \begingroup\ifchilddoc\else
388 \renewcommand{\thepage}{\arabic{page}}

```

Prepare a title page to display some relevant data:

```
389 \pdfbookmark[1]{\metaterm{title}}{title}
390 \thispagestyle{empty}
391 \vspace*{\fill}
392 \begin{center}
393 \metapick[course]{titletext}
394 \end{center}
395 \vspace*{\fill}\vspace*{\fill}
396 \newpage
```

Prepare a copyright and license page using data specified above:

```
397 \phantomsection\pdfbookmark[1]{\metaterm{copyright}}{copyright}
398 \thispagestyle{empty}
399 \vspace*{\fill}\vspace*{\fill}
400 \begin{center}\begin{minipage}{11cm}\raggedright
401 \metapick[print]{rightstext}
402 \end{minipage}\end{center}
403 \vspace*{\fill}\vspace*{\fill}\vspace*{\fill}
404 \newpage
```

Print table of contents:

```
405 \makeatletter\renewcommand\@pnumwidth{2.4em}\makeatother
406 \setcounter{tocdepth}{2}
407 \phantomsection\pdfbookmark[1]{\metaterm{contents}}{contents}
408 {\parskip0pt\tableofcontents}
409 \exercisecleardoublepage\setcounter{page}{1}
```

End of frontmatter:

```
410 \fi\endgroup
```

**Include Sheets.** Include problem sheets:

```
411 \include{exframe-ser-01}
412 \include{exframe-ser-02}
413 \include{exframe-ser-03}
```

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

```
414 \def\jobnameunused{exframe-ser-aa}
415 \ifx\childdocname\jobnameunused\include{\jobnameunused}\fi
```

**End.** End of document body:

```
416 \end{document}
```

## 2.2 Sheet File

Provide some source files `exframe-ser-01.tex`, `exframe-ser-02.tex`, `exframe-ser-03.tex`, `exframe-ser-aa.tex` for problem sheets.

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

```
417 %\providecommand{\printsol}{n}
418 \input{childdoc.def}
419 \childdocof{exframe-ser}
```

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

**Sheet Environment.** Declare a sheet with intended due date:

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

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

**Problems.** Start a problem:

```
421 \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).  
Also declare a figure `tag-solfig` for use within the solution:

```
422 \ifdefined\mpostuse
423 \begin{mpostfig}[label={\problemtag-fig}]
424 interim xu:=1.5cm;
425 paths[1]:=fullcircle scaled 1xu;
426 fillshape(paths[1], 0.7white) pensize(1pt);
427 label(btex \figure etex, center(paths[1]));
428 \end{mpostfig}
429 \begin{onlysolutions}
430 \begin{mpostfig}[label={\problemtag-solfig}]
431 interim xu:=1.5cm;
432 paths[1]:=fullcircle scaled 1xu;
433 paths[2]:=((subpath (-2,2) of paths[1])--cycle) shifted (+0.5xu,0);
434 paths[3]:=((subpath (2,6) of paths[1])--cycle) shifted (-0.5xu,0);
435 fillshape(paths[2], 0.7white) pensize(1pt);
436 fillshape(paths[3], 0.7white) pensize(1pt);
437 \end{mpostfig}
438 \end{onlysolutions}
439 \fi
```

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

```
440 \lorem
441
442 \begin{subproblem}
443 \lorem
444 \begin{center}
445 \ifdefined\mpostuse\mpostuse{\problemtag-fig}\else figure\fi
446 \end{center}
447 \lorem
448 \end{subproblem}
449
450 \begin{solution}
451 \lorem
452 \begin{center}
453 \ifdefined\mpostuse\mpostuse{\problemtag-solfig}\else figure\fi
454 \end{center}
455 \lorem
456 \end{solution}
457
458 \begin{subproblem}
```

```
459 \lorem
460 \end{subproblem}
461
462 \begin{solution}
463 \lorem
464 \end{solution}
465
466 \lorem
467
468 \begin{subproblem}
469 \lorem
470 \end{subproblem}
471
472 \begin{solution}
473 \lorem
474 \end{solution}
```

End the problem:

```
475 \end{problem}
```

Start new page:

```
476 \turnover
```

Write a second problem to accompany the first one:

```
477 \begin{problem}
```

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

```
478 \lorem
479
480 \begin{subproblem}
481 \lorem
482 \begin{solution}
483 \lorem
484 \end{solution}
485 \end{subproblem}
486
487 \begin{subproblem}
488 \lorem
489 \begin{solution}
490 \lorem
491 \end{solution}
492 \end{subproblem}
```

End the problem:

```
493 \end{problem}
```

**End Sheet.** End the sheet:

```
494 \end{sheet}
```

## 2.3 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 `exframe-ser-pnn.tex` containing the problem environment. This file should start with:

```
495 %\providecommand{\printsol}{n}
496 \input{childdoc.def}
497 \childdocby{exframe-ser}
```

and end with `\endinput`. Then compose the problem sheet by including the appropriate set of problem files via `\input{exframe-ser-pnn}`.

In the example, a sheet file `exframe-ser-03.tex` includes two problem files `exframe-ser-pe.tex` and `exframe-ser-pf.tex`.

## 2.4 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 `exframe-ser.sh` compiles one part of the collection of problem sheets in a given mode.

Shebang for `bash` script:

```
498 #!/bin/bash
```

Configure and declare variables with default values. `srcmain` defines the name of the main source file; `srcsecnm` 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 `nm`:

```
499 srcmain="exframe-ser"
500 srcsec="exframe-ser-"
501 trglist=(Problems Solutions)
502 trgsol=(n y)
503 secnum="01 02 03 aa"
```

Display usage:

```
504 if [ -z $1 ]
505 then
506   echo "Usage:
507   $0 number [version]
508   number: number of sheet, 0 for combined document
509   version: 0 for problems, 1 for solutions
510   $0 filename
511   filename: target file to be compiled"
512   exit 1
513 fi
```

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

```
514 num="$1"
515 ver="$2"
516 nl=$'\n'
517 secokay=""
518 make=".pdf"
```

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

```
519 for v in "${trglist[@]}"
520 do
521   if [[ $num =~ ^$v ]]
522   then
523     ver=$v
524     num=${num#$v}
525     if [[ $num =~ ^.*\.tex$ ]]; then make=".tex"; fi
526     num=${num%.*}
527   fi
528 done
```

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

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

Check whether num is acceptable:

```
531 if [[ -z $num ]]; then secokay="okay"; fi
532 for v in $secnum
533 do
534   if [[ "$num" == "$v" ]]; then secokay="okay"; fi
535 done
```

Otherwise display error message and exit:

```
536 if [[ -z $secokay ]]
537 then
538   echo "error: unknown sheet"
539   exit 1
540 fi
```

Disable newline character for command line tex code:

```
541 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:

```
542 function docompile
543 {
544   if [[ -z $num ]]
545   then
546     job="$srcmain"
547     fwd="\childdocforward{$srcmain}"
548   else
549     job="$srcsec$num"
550     fwd="\childdocforward[$srcmain]{$srcsec$num}"
551   fi
552   body="\def\jobname{$job}$optdef\input{childdoc.def}$fwd"
553   for pass in first main extra
554   do
555     par="";
556     if [[ "$pass" == "first" ]]; then par="-draftmode"; fi
557     drop="This is|entering extended mode|\write18"
558     pdflatex -shell-escape -interaction=batchmode $par \
559       -jobname "$trg" "$body" | grep -vE "$drop"
560     if [[ "$pass" != "main" ]]; then continue; fi
561   fi
```

```

562     if ! (grep -E -q "may have changed|rerunfilecheck Warning" "$trg.log")
563         then break; fi
564 done
565 grep -E "^! |Warning|Error|Undefined|Overfull|Underfull" "$trg.log"
566 }

```

Function to generate a chiddoc compile file with specific options:

```

567 function writesource
568 {
569     if [[ -z $num ]]
570     then
571         fwd="\chiddocforward{$srcmain}"
572     else
573         fwd="\chiddocforwardprefix[$srcmain]{$target}{$srcsec}"
574     fi
575     body="$optdef\input{chiddoc.def}$nl$fwd"
576     echo "$body" > $trg.tex
577 }

```

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

```

578 for i in "${!trglist[@]}"
579 do
580     if [[ -z $ver || "$ver" == "${trglist[$i]}" || $ver = $i ]]
581     then
582         target="${trglist[$i]}"
583         sol="${trgsol[$i]}"
584         trg="$target$num"
585         optdef="\def\draftver{n}$nl\def\printsol{$sol}$nl"
586         if [[ "$make" == ".pdf" ]]; then docompile; else writesource; fi
587     fi
588 done

```

Finish with blank line:

```

589 echo

```

**Makefile.** Define a make file `exframe-ser.mak` for the project. The compilation is then started by `make -f exframe-ser.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:

```

590 SRCMAIN = exframe-ser
591 SRCSEC = exframe-ser-
592 SRCPRB = exframe-ser-p
593 SCRIPT = exframe-ser.sh
594 MAKEFILE = exframe-ser.mak
595 TRGLIST = Problems Solutions
596 SECNUM = 01 02 03 aa
597 PREREQS = $(SRCMAIN).tex
598
599 SRCSECFILES = $(SECNUM:%=$(SRCSEC)%.tex)
600 TRGMAINFILES = $(foreach trg,$(TRGLIST),$(trg).pdf)
601 TRGSECFILES = $(foreach trg,$(TRGLIST),$(trg).pdf $(SECNUM:%=$(trg)%.pdf))
602 GENFILES = $(foreach trg,$(TRGLIST),$(trg).tex $(SECNUM:%=$(trg)%.tex))
603 BAKFILES = $(PREREQS) $(SRCSECFILES) $(GENFILES)\

```

```
604 $(MAKEFILE) $(SCRIPT) $(SRCPRB)*
```

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

```
605 default: sheets ;
606 main: $(TRGMAINFILES) ;
607 sheets: $(TRGSECFILES) ;
608 sheet%: $(foreach trg,$(TRGLIST),$(trg).pdf) ;
609 all: main sheets ;
610 sources: $(GENFILES) ;
```

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

```
611 $(TRGMAINFILES): $(SRCSECFILES) $(PREREQS)
612  bash ./$(SCRIPT) $@
613 $(word 1,$(TRGLIST)).pdf: $(SRCSEC).tex $(PREREQS)
614  bash ./$(SCRIPT) $@
615 $(word 2,$(TRGLIST)).pdf: $(SRCSEC).tex $(PREREQS)
616  bash ./$(SCRIPT) $@
617 $(GENFILES):
618  bash ./$(SCRIPT) $@
```

Touch main file for recompile:

```
619 touch:
620  touch $(SRCMAIN).tex
```

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

```
621 clean:
622  rm -f $(foreach ext,.aux .log,$(SECNUM:%=$(SRCSEC)%$(ext)))
623  rm -f $(foreach trg,$(TRGLIST),$(SECNUM:%=$(trg).log) $(trg).log)
624  rm -f $(foreach ext,.aux .log .out .toc,$(SRCMAIN)$$(ext))
625  rm -f $(foreach ext,.mp .mpx -*.mps,$(SRCMAIN)$$(ext))
626  rm -f $(foreach ext,-tmp.log -tmp.mp -tmp.mpx,$(SRCMAIN)$$(ext))
627  rm -f mpxerr.tex mpxerr.log mpxerr.dvi texput.log
628  rm -f $(patsubst %,$(SRCPRB)*%,.aux .log .mp .mpx -*.mps)
629  rm -f $(patsubst %,$(SRCPRB)*%,-tmp.log -tmp.mp -tmp.mpx)
```

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

```
630 clean-bak:
631  rm -f $(BAKFILES:%=%~) $(BAKFILES:%=%.bak)
```

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

```
632 clean-all: clean
633  rm -f $(TRGSECFILES) $(TRGMAINFILES) $(GENFILES)
634  rm -f $(SECNUM:%=$(SRCSEC)%).pdf $(SRCMAIN).pdf
635  rm -f $(SRCPRB)*.pdf
```

## 3 Implementation

This section describes the package `exframe.sty`.

### 3.1 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:

```
636 \RequirePackage{verbatim}
637 \RequirePackage{xkeyval}
```

### General Definitions.

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

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

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

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

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

```
640 \long\def\exf@expswitch#1#2{#2{#1}}
641 \long\def\exf@exparg#1#2{\exf@exptwo\exf@expswitch{#2}{#1}}
```

`\exf@csdo` Some macros to conveniently expand `\csname` arguments before expanding the macro:  
`\exf@csdotwo`

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

`\exf@csor` A macro to return `\csname` or default if not exists:

```
644 \def\exf@csor#1#2{\ifcsname#1\endcsname\csname#1\endcsname\else#2\fi}
```

`\exf@append@def` Add definitions to macros (after or before original content):

`\exf@prepend@def`

```
645 \long\def\exf@append@def#1#2{\exf@exptwo\def#1\expandafter{#1#2}}
646 \long\def\exf@prepend@switch#1#2#3{#2{#3#1}}
647 \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:

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

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

```
650 \newcommand{\exf@ifis}[3]{%
651 {\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):

```
652 \newcommand{\exf@href}[2]{%
653 \ifdefined#1\ifx#1\exf@empty#2\else%
654 \ifdefined\hyperlink\protect\hyperlink{#1}{#2}\else#2\fi\fi\else#2\fi}
```

`\exf@text` 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:

```
655 \newcommand{\exf@text}[1]{\ifdefined\text\text{#1}\else#1\fi}
656 \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:

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

### Auxiliary File Data Storage.

`\exf@notedata` Process stored data by handler:

```
659 \newcommand{\exf@notedata}[3]{\csname exf@notedata@#1\endcsname{#2}{#3}}
```

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

```
660 \AtBeginDocument{\immediate\write\@auxout{%
661   \string\providecommand{\string\exf@notedata}[3]{}}}
```

`\exf@writedata` Write data to the .aux file:

```
662 \newcommand{\exf@writedata}[3]{%
663   {\immediate\write\@auxout{\string\exf@notedata{#1}{#2}{#3}}}
```

## 3.2 Package Setup

### Initialisation Options.

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

Configure names of main environments and counters:

```
664 \def\exf@problemname{problem}
665 \def\exf@subproblemname{sub\exf@problemname}
666 \def\exf@solutionname{solution}
667 \def\exf@sheetname{sheet}
668 \def\exf@problemcounter{problem}
669 \def\exf@subproblemcounter{sub\exf@problemcounter}
670 \def\exf@solutioncounter{solution}
671 \def\exf@sheetcounter{sheet}
672 \define@key{exframe.sty}{problemenv}{\def\exf@problemname{#1}}
673 \define@key{exframe.sty}{subproblemenv}{\def\exf@subproblemname{#1}}
674 \define@key{exframe.sty}{solutionenv}{\def\exf@solutionname{#1}}
675 \define@key{exframe.sty}{sheetenv}{\def\exf@sheetname{#1}}
676 \define@key{exframe.sty}{problemcounter}{\def\exf@problemcounter{#1}}
677 \define@key{exframe.sty}{subproblemcounter}{\def\exf@subproblemcounter{#1}}
678 \define@key{exframe.sty}{solutioncounter}{\def\exf@solutioncounter{#1}}
679 \define@key{exframe.sty}{sheetcounter}{\def\exf@sheetcounter{#1}}
```

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

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

Whether to load package metastr (available while loading only):

```
682 \define@boolkey{exframe.sty}[exf@]{metastr}[true]{}
```

### Setup Options.

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

Main switch for solutions:

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

Switch for writing pdf metadata:

```
684 \define@choicekey{exf@setup}{pdfdata}%  
685   {auto,manual,sheet,off}[auto]{\def\exf@metadata{#1}}  
686 \def\exf@metadata{auto}
```

Write line number indicators to output file:

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

Prepare two-sided sheets:

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

Generate hyperreferences from solutions to corresponding problems:

```
689 \define@boolkey{exf@setup}[exf@]{solutionhref}[true]{}  
690 \exf@solutionhreftrue
```

Automatically generate labels for sheets and problems:

```
691 \define@boolkey{exf@setup}[exf@]{autolabelsheet}[true]{}  
692 \define@boolkey{exf@setup}[exf@]{autolabelproblem}[true]{}
```

Write warning message to document for better detection of inconsistencies:

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

Activate buffering of solutions:

```
694 \define@boolkey{exf@setup}[exf@]{solutionbuf}[true]{}  
695 \exf@solutionbuftrue
```

Activate buffering of problems:

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

Activate buffering of subproblems:

```
697 \define@boolkey{exf@setup}[exf@]{subproblembuf}[true]{}
```

`\exf@emptytestchar` Redefine character for testing emptiness:

```
698 \def\exf@emptytestchar{&}  
699 \define@key{exf@setup}{emptytestchar}{\def\exf@emptytestchar{#1}}
```

**Processing.** Process global options while loading package:

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

`\exercisestop` Configure package when package is already loaded:

```
701 \newcommand{\exercisestop}[1]{\exf@expsetkeys{exf@setup}{#1}}
```

**Additional Packages.** Load metastr package if desired:

```
702 \ifexf@metastr  
703 \PassOptionsToPackage{course=true}{metastr}  
704 \RequirePackage{metastr}  
705 \fi
```

## Solutions Only Processing.

`onlysolutions` Process block only in solutions mode:

```
706 \newenvironment{onlysolutions}%
707   {\ifsolutions\else%
708     \let\endonlysolutions\endcomment%
709     \expandafter\comment\fi}%
710   {}
```

## 3.3 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:

```
711 \newcommand{\exerciseconfig}[1]{%
712   \@ifnextchar[{\exf@configopt{#1}}{\exf@confignoopt{#1}}]
713   \long\def\exf@configopt#1[#2]#3{%
714     \exf@csdo\def{exf@config@#1}{#2}{#3}%
715     \exf@csdo\renewcommand{exf@config@#1}[#2]{#3}%
716   \long\def\exf@confignoopt#1#2{\exf@csdo\def{exf@config@#1}{#2}}
```

`\exerciseconfigappend` Append to a (parameterless) configuration macro:

`\exerciseconfigprepend`

```
717 \newcommand{\exerciseconfigappend}[2]{%
718   \exf@csdo\exf@append@def{exf@config@#1}{#2}}
719 \newcommand{\exerciseconfigprepend}[2]{%
720   \exf@csdo\exf@prepend@def{exf@config@#1}{#2}}
```

`\getexerciseconfig` Get configuration macro:

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

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

```
722 \newcommand{\exerciseconfigempty}[3]{\exf@csdo\ifx{exf@config@#1}\exf@empty%
723   #2\else#3\fi}
```

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

`\exerciseifnotempty`

```
724 \long\def\exerciseifempty#1#2#3{%
725   \if\exf@emptytestchar#1\exf@emptytestchar#2\else#3\fi}
726 \long\def\exerciseifnotempty#1#2{%
727   \if\exf@emptytestchar#1\exf@emptytestchar\else#2\fi}
```

### Terms.

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

```
728 \exerciseconfig{termsheet}{Sheet}
729 \exerciseconfig{termsheets}{Sheets}
```

```

730 \exerciseconfig{termpblem}{Problem}
731 \exerciseconfig{termpblems}{Problems}
732 \exerciseconfig{termsolution}{Solution}
733 \exerciseconfig{termsolutions}{Solutions}
734 \exerciseconfig{termpoint}{point}
735 \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:

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

Style for problems:

```
737 \exerciseconfig{styletextproblem}{}

```

Style for solutions:

```
738 \exerciseconfig{styletextsolution}{\footnotesize}

```

Basic style for titles:

```
739 \exerciseconfig{styletitle}{\bfseries}

```

Style for problem titles:

```
740 \exerciseconfig{styletitleproblem}{\large}

```

Style for subproblem titles:

```
741 \exerciseconfig{styletitlesubproblem}{}

```

Style for solution titles:

```
742 \exerciseconfig{styletitlesolution}{}

```

Style for problem section title in solution block:

```
743 \exerciseconfig{styletitlesolutionsproblem}{\small}

```

Style for solution block title:

```
744 \exerciseconfig{styletitlesolutions}{\normalsize}

```

Style for problem block title:

```
745 \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:

```
746 \exerciseconfig{skipproblemabove}{3.25ex plus 1ex minus 1.5ex}

```

Space below problem environment:

```
747 \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):

```
748 \exerciseconfig{skipproblemtitle}{3pt plus 1pt minus 1pt}
```

Horizontal space between items in the problem opening line:

```
749 \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:

```
750 \exerciseconfig{skipproblemitem}{0pt}
```

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

```
751 \exerciseconfig{skipsubproblemabove}{1.5ex plus 0.5ex minus 1ex}
```

```
752 \exerciseconfig{skipsubproblembelow}{1.5ex plus 0.5ex minus 1ex}
```

```
753 \exerciseconfig{skipsubproblemtitle}{-1em}
```

```
754 \exerciseconfig{skipsubprobleminfo}{0.25em}
```

```
755 \exerciseconfig{skipsubproblemitem}{-1pt}
```

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

```
756 \exerciseconfig{skipsolutionabove}{0ex}
```

```
757 \exerciseconfig{skipsolutionbelow}{1.5ex plus 0.5ex minus 1ex}
```

```
758 \exerciseconfig{skipsolutiontitle}{-0.5em}
```

```
759 \exerciseconfig{skipsolutioninfo}{0.25em}
```

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

```
760 \exerciseconfig{skipsolutionitem}{0pt}
```

```
761 \exerciseconfig{skipsolutionitemsub}{0pt}
```

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

```
762 \exerciseconfig{skipsolutionsabove}{1.5ex plus 0.5ex minus 1ex}
```

```
763 \exerciseconfig{skipsolutionsbelow}{1.5ex plus 0.5ex minus 1ex}
```

Space above problem titles in a solution block:

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

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

```
765 \exerciseconfig{skipsolutionsproblemtitle}{1.0ex plus 0ex minus 0.5ex}
```

```
766 \exerciseconfig{skipsolutionsproblem}{\exf@config@skipsolutionsproblemtitle}
```

Space following title of a solution block:

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

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

```
768 \exerciseconfig{skipproblemsabove}{1.5ex plus 0.5ex minus 1ex}
```

```
769 \exerciseconfig{skipproblemsbelow}{1.5ex plus 0.5ex minus 1ex}
```

Space following title of a problem block:

```
770 \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:

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

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

```
772 \exerciseconfig{insertsheetclearpage}{\exercisecleardoublepage}
```

Code to insert before a sheet is displayed:

```
773 \exerciseconfig{insertsheetbefore}{}
```

Code to insert after a sheet is displayed:

```
774 \exerciseconfig{insertsheetafter}{}
```

Code to insert before a solution block is displayed:

```
775 \exerciseconfig{insertsolutionsbefore}{}
```

Code to insert after a solution block is displayed:

```
776 \exerciseconfig{insertsolutionsafter}{}
```

Code to insert before a problem block is displayed:

```
777 \exerciseconfig{insertproblemsbefore}{}
```

Code to insert after a problem block is displayed:

```
778 \exerciseconfig{insertproblemsafter}{}
```

Code to insert before a problem is displayed:

```
779 \exerciseconfig{insertproblembefore}{}
```

Code to insert after a problem is displayed:

```
780 \exerciseconfig{insertproblemafter}{}
```

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

```
781 \exerciseconfig{insertproblemsolution}{}
```

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

```
782 \exerciseconfig{insertprobleminfo}{}
```

Code to determine the display of problems:

```
783 \exerciseconfig{insertproblemsselect}[1]{}
```

Analogous definitions for subproblems:

```
784 \exerciseconfig{insertsubproblembefore}{}
```

```
785 \exerciseconfig{insertsubproblemafter}{}
```

```
786 \exerciseconfig{insertsubprobleminfo}{}
```

```
787 \exerciseconfig{insertsubproblemsolution}{}
```

```
788 \exerciseconfig{insertsubproblemselect}[1]{}
```

Analogous definitions for solutions:

```
789 \exerciseconfig{insertsolutionbefore}{}
```

```
790 \exerciseconfig{insertsolutionafter}{}
```

```
791 \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:

```
792 \exerciseconfig{composeitemsep}{\ }
```

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

```
793 \exerciseconfig{composetitlesheet}[2]%  
794 {\exerciseifempty{#2}{\getexerciseconfig{termsheet} #1}{#2}}
```

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

```
795 \exerciseconfig{composemetasheet}[2]%  
796 {\getexerciseconfig{composetitlesheet}{#1}{#2}}
```

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

```
797 \exerciseconfig{composetocsheet}[2]%  
798 {\exerciseifempty{#2}{\getexerciseconfig{termsheet} #1}{#2}}
```

Problem item separator, delimiter:

```
799 \exerciseconfig{composeitemproblemsep}{\getexerciseconfig{composeitemsep}}  
800 \exerciseconfig{composeitemproblemdelim}{.}
```

Compose problem item; argument is problem number:

```
801 \exerciseconfig{composeitemproblem}[1]%  
802 {#1\getexerciseconfig{composeitemproblemdelim}}
```

Unnamed problem separator, delimiter:

```
803 \exerciseconfig{composebareproblemsep}{ }  
804 \exerciseconfig{composebareproblemdelim}{}
```

Compose unnamed problem title; arguments are problem number and name; includes exception for legacy behaviour:

```
805 \exerciseconfig{composebareproblem}[1]%  
806 {\getexerciseconfig{termproblem}\getexerciseconfig{composebareproblemsep}%  
807 \ifx\exf@config@composebareproblemdelim\exf@config@composeitemproblemdelim%  
808 \getexerciseconfig{composeitemproblem}{#1}\else%  
809 #1\getexerciseconfig{composebareproblemdelim}\fi}
```

Named problem separator, delimiter:

```
810 \exerciseconfig{composenamedproblemsep}{ }  
811 \exerciseconfig{composenamedproblemdelim}{.}
```

Compose named problem title; arguments are problem number and name; includes exception for legacy behaviour:

```
812 \exerciseconfig{composenamedproblem}[2]%  
813 {\ifx\exf@config@composenamedproblemdelim\exf@config@composeitemproblemdelim%  
814 \getexerciseconfig{composeitemproblem}{#1}\else%  
815 #1\getexerciseconfig{composenamedproblemdelim}\fi%  
816 \getexerciseconfig{composenamedproblemsep}#2}
```

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”:

```
817 \exerciseconfig{composetitleproblem}[2]{\exerciseifempty{#1}%
818   {\exerciseifempty{#2}{}}{#2}}%
819   {\exerciseifempty{#2}{\getexerciseconfig{composebareproblem}{#1}}}%
820   {\getexerciseconfig{composenamedproblem}{#1}{#2}}}
```

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

```
821 \exerciseconfig{composetocproblem}[2]%
822   {\exerciseifempty{#2}{\getexerciseconfig{termproblem} #1}%
823   {#1\getexerciseconfig{composenamedproblemdelim} #2}}
```

Subproblem item separator, delimiter:

```
824 \exerciseconfig{composeitemsubproblemsep}{\getexerciseconfig{composeitemsep}}
825 \exerciseconfig{composeitemsubproblemdelim}{}
```

Compose subproblem item; argument is subproblem number:

```
826 \exerciseconfig{composeitemsubproblem}[1]%
827   {#1\getexerciseconfig{composeitemsubproblemdelim}}
```

Compose subproblem title; argument is subproblem number:

```
828 \exerciseconfig{composetitlesubproblem}[1]%
829   {\getexerciseconfig{composeitemsubproblem}{#1}}
```

Solution item separator, delimiter:

```
830 \exerciseconfig{composeitemsolutionsep}{\getexerciseconfig{composeitemsep}}
831 \exerciseconfig{composeitemsolutiondelim}{:}
```

Compose solution item; arguments are problem and subproblem number:

```
832 \exerciseconfig{composeitemsolution}[2]{}
833 \exerciseconfig{composeitemsolutionsub}[2]%
834   {\getexerciseconfig{composeitemsubproblem}{#2}}
```

Compose solution item for first solution in a problem section; arguments are problem and subproblem number:

```
835 \exerciseconfig{composeitemsolutionfirst}[2]%
836   {\getexerciseconfig{composeitemsolution}{#1}{#2}}
837 \exerciseconfig{composeitemsolutionfirstsub}[2]%
838   {\getexerciseconfig{composeitemsolutionsub}{#1}{#2}}
```

Compose title for single solution (solution intro if `solutionbelow` is here or subproblem); arguments are corresponding problem and subproblem number:

```
839 \exerciseconfig{composetitlesolutionsingle}[2]%
840   {\getexerciseconfig{termsolution}%
841   \getexerciseconfig{composeitemsolutiondelim}}
```

Compose title for one out of several solutions (solution intro if `solutionbelow` is problem, sheet or manual); arguments are corresponding problem and subproblem number:

```
842 \exerciseconfig{composetitlesolutionproblem}[1]{}
843 \exerciseconfig{composetitlesolutionsubproblem}[2]%
844   {\getexerciseconfig{composeitemsubproblem}{#2}}
845 \exerciseconfig{composetitlesolutionmulti}[2]{\exerciseifempty{#2}%
846   {\getexerciseconfig{composetitlesolutionproblem}{#1}}}%
847   {\getexerciseconfig{composeitemsubproblem}{#1}{#2}}}
```

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

```
848 \exerciseconfig{composetocsolution}[2]%
849 {\getexerciseconfig{composetocproblem}{#1}{#2}}
```

Compose title for solution block:

```
850 \exerciseconfig{composetitlesolutions}%
851 {\getexerciseconfig{termsolutions}}
```

Compose title for problem block:

```
852 \exerciseconfig{composetitleproblems}%
853 {\getexerciseconfig{termproblems}}
```

Compose table of contents line for solution block:

```
854 \exerciseconfig{composetocsolutions}%
855 {\getexerciseconfig{composetitlesolutions}}
```

Compose table of contents line for problem block:

```
856 \exerciseconfig{composetocproblems}%
857 {\getexerciseconfig{composetitleproblems}}
```

Compose sectional title for solution following a single problem (problem section intro if `solutionbelow` is `problem`); arguments are problem number and raw title:

```
858 \exerciseconfig{composetitlesolutionsproblemsingle}[2]%
859 {\getexerciseconfig{termsolution}}
```

Compose sectional title for solution of one problem within a solution block (problem section intro if `solutionbelow` is `sheet` or `manual`); arguments are problem number and raw title:

```
860 \exerciseconfig{composetitlesolutionsproblemmulti}[2]%
861 {\exerciseifempty{#2}{\getexerciseconfig{composebareproblem}{#1}}%
862 {\getexerciseconfig{composenamedproblem}{#1}{#2}}}
```

Compose label:

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

**Points.** Compose number of points:

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

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

```
865 \exerciseconfig{composepoints}[1]{\getexerciseconfig{composepointsnum}{#1}~%
866 \ifdim #1pt=1pt\getexerciseconfig{termpoint}%
867 \else\getexerciseconfig{termpoints}\fi}
```

Compose points declaration for use in opening line:

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

Compose points declaration for use in margin:

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

Compose points declaration for use in text:

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

Compose points declaration for use in sheet data:

```
871 \exerciseconfig{composepointssheet}[1]{%
872  \exerciseifnotempty{#1}{\getexerciseconfig{composepoints}{#1}}}
```

Compose points declaration for solution with comment:

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

Compose alternative points for solution declaration with comment:

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

Compose pairs of points; omit 0 components:

```
877 \exerciseconfig{composepointspair}[2]{%
878  \ifdim#2pt=0pt%
879  \getexerciseconfig{composepoints}{#1}%
880  \else\ifdim#1pt=0pt%
881  +\getexerciseconfig{composepoints}{#2}%
882  \else%
883  \getexerciseconfig{composepointsnum}{#1}+%
884  \getexerciseconfig{composepointsnum}{#2}~%
885  \getexerciseconfig{termpoints}%
886  \fi\fi}
```

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

```
887 \exerciseconfig{composepointspairbody}[2]{%
888  \ifdim#2pt=0pt\getexerciseconfig{composepointspairbody}{#1}\else%
889  (\getexerciseconfig{composepointspair}{#1}{#2})\fi}
890 \exerciseconfig{composepointspairstart}[2]{%
891  \ifdim#2pt=0pt\getexerciseconfig{composepointspairstart}{#1}\else%
892  (\getexerciseconfig{composepointspair}{#1}{#2})\fi}
893 \exerciseconfig{composepointspairmargin}[2]{%
894  \ifdim#2pt=0pt\getexerciseconfig{composepointspairmargin}{#1}\else%
895  \getexerciseconfig{composepointspair}{#1}{#2}\fi}
896 \exerciseconfig{composepointspairsheet}[2]{%
897  \ifdim#2pt=0pt\getexerciseconfig{composepointssheet}{#1}\else%
898  \getexerciseconfig{composepointspair}{#1}{#2}\fi}
899 \exerciseconfig{composepointspairaward}[3]{%
900  \ifdim#2pt=0pt\getexerciseconfig{composepointsaward}{#1}{#3}\else%
901  (\getexerciseconfig{composepointspair}{#1}{#2})%
902  \exerciseifnotempty{#3}{; #3})\fi}
903 \exerciseconfig{composepointspairawardalt}[3]{%
904  \ifdim#2pt=0pt\getexerciseconfig{composepointspairawardalt}{#1}{#3}\else%
905  (\getexerciseconfig{composepointspair}{#1}{#2}*%
906  \exerciseifnotempty{#3}{; #3})\fi}
```

Compose pairs of points for designated situations:

```
907 \exerciseconfig{composepointspairbodyproblem}[2]{%
908  \getexerciseconfig{composepointspairbody}{#1}{#2}}
909 \exerciseconfig{composepointspairbodysubproblem}[2]{%
910  \getexerciseconfig{composepointspairbody}{#1}{#2}}
911 \exerciseconfig{composepointspairbodysolution}[2]{%
912  \getexerciseconfig{composepointspairbody}{#1}{#2}}
913 \exerciseconfig{composepointspairstartproblem}[2]{%
914  \getexerciseconfig{composepointspairstart}{#1}{#2}}
```

```

915 \exercisecfg{composepointspairstartsubproblem}[2]{%
916   \getexercisecfg{composepointspairstart}{#1}{#2}}
917 \exercisecfg{composepointspairstartsolution}[2]{%
918   \getexercisecfg{composepointspairstart}{#1}{#2}}
919 \exercisecfg{composepointspairmarginproblem}[2]{%
920   \getexercisecfg{composepointspairmargin}{#1}{#2}}
921 \exercisecfg{composepointspairmarginsubproblem}[2]{%
922   \getexercisecfg{composepointspairmargin}{#1}{#2}}
923 \exercisecfg{composepointspairmarginresolution}[2]{%
924   \getexercisecfg{composepointspairmargin}{#1}{#2}}

```

Display points in the margin:

```

925 \exercisecfg{insertpointsmargin}[1]{\marginpar{\footnotesize #1}}
926 \exercisecfg{insertpointsmarginproblem}[1]{%
927   \getexercisecfg{insertpointsmargin}{#1}}
928 \exercisecfg{insertpointsmarginsubproblem}[1]{%
929   \getexercisecfg{insertpointsmargin}{#1}}
930 \exercisecfg{insertpointsmarginresolution}[1]{%
931   \getexercisecfg{insertpointsmargin}{#1}}

```

Display warning about points mismatch:

```

932 \exercisecfg{insertwarnpoints}[3]{%
933   {\textbf{points mismatch for #1 (#2 determined vs.\ #3 given)}}}

```

Display warning about points changed:

```

934 \exercisecfg{insertwarnpointsrerun}[1]{%
935   {\textbf{points changed for #1 (please recompile)}}}

```

**Counters.** Define counter display via configuration interface:

```

936 \exercisecfg{countersheet}{\arabic{\exf@sheetcounter}}
937 \exercisecfg{counterproblem}{\arabic{\exf@problemcounter}}
938 \exercisecfg{counterproblemmax}{10}
939 \exercisecfg{countersubproblem}{\alph{\exf@subproblemcounter}}
940 \exercisecfg{countersubproblemmax}{m}
941 \exercisecfg{countersheetequation}{\arabic{equation}}
942 \exercisecfg{counterproblemequation}{P\arabic{equation}}
943 \exercisecfg{countersolutionequation}{S\arabic{equation}}

```

**Further Definitions.**

`tagsheet` Templates for automatic generation of tags:  
`tagproblem`  
`tagsubproblem`

```

944 \exercisecfg{tagsheet}{\arabic{\exf@sheetcounter}}
945 \exercisecfg{tagproblem}{\csname the\exf@problemcounter\endcsname}
946 \exercisecfg{tagsubproblem}{\problemtag-\arabic{\exf@subproblemcounter}}

```

`labelsheet` Templates for automatic generation of labels from tags:  
`labelproblem`  
`labelsubproblem`

```

947 \exercisecfg{labelsheet}[1]{sheet:#1}
948 \exercisecfg{labelproblem}[1]{prob:#1}
949 \exercisecfg{labelsubproblem}[1]{\getexercisecfg{labelproblem}{#1}}

```

`toclevel...` Table of contents levels for sheets, problems, solutions of problems and solution blocks; empty means no writing to table of contents:

```

950 \exerciseconfig{toclevelsheet}{}
951 \exerciseconfig{toclevelproblem}{}
952 \exerciseconfig{toclevelproblems}{}
953 \exerciseconfig{toclevelsolution}{}
954 \exerciseconfig{toclevelsolutions}{}

```

`extsolutions` Filename extension for solution and problem blocks:

```

extproblems 955 \exerciseconfig{extsolutions}{.sol}
956 \exerciseconfig{extproblems}{.prb}

```

### 3.4 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:

```

957 \newcommand{\defexercisestylearg}[3] []{%
958   \def\exf@tmp{#1}\ifx\exf@tmp\exf@empty%
959     \define@key{exf@style}{#2}{#3}\else%
960     \define@key{exf@style}{#2}[#1]{#3}\fi}

```

`\defexercisestyle` Define a style with a boolean argument; execute code only if true:

```

961 \newcommand{\defexercisestyle}[2]{%
962   \exf@csdotwo\long\def{exf@style@code@#1}{#2}%
963   \exf@exparg{\define@boolkey{exf@style}[exf@style@]{#1}[true]}%
964   {\csname ifexf@style@#1\endcsname\csname exf@style@code@#1\endcsname\fi}}

```

`\exercisestyle` Process styles:

```

965 \newcommand{\exercisestyle}[1]{\exf@expsetkeys{exf@style}{#1}}

```

#### Default Exercise Styles.

`problemmanual` Delay display of problems:

```

966 \define@boolkey{exf@style}[exf@]{problemmanual}[true]{}
967 \exf@problemmanualfalse

```

`solutionbelow` Choose location for solutions:

```

968 \def\exf@solutionbelow{subproblem}
969 \define@choicekey{exf@style}{solutionbelow}%
970   {here,subproblem,subproblem*,problem,problem*,sheet>manual}%
971   {\ifexf@solfile@open\else\gdef\exf@solutionbelow{#1}\fi}

```

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

```

problemequation 972 \defexercisestyle{sheetequation}{}
solutionequation 973 \defexercisestyle{problemequation}{}
974 \defexercisestyle{solutionequation}{}
975 \exf@style@solutionequationtrue

```

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

```

subproblempointsat
solutionpointsat 976 \def\exf@pointsat{start}
977 \define@choicekey{exf@style}{problempointsat}%
978   {start,start*,margin,end,manual,off}{\def\exf@pointsat{#1}}
979 \define@choicekey{exf@style}{pointsat}%
980   {start,start*,margin,end,manual,off}{\def\exf@pointsat{#1}}
981 \def\exf@subpointsat{end}
982 \define@choicekey{exf@style}{subproblempointsat}%
983   {start,start*,margin,end,manual,off}{\def\exf@subpointsat{#1}}
984 \define@choicekey{exf@style}{subpointsat}%
985   {start,start*,margin,end,manual,off}{\def\exf@subpointsat{#1}}
986 \def\exf@solpointsat{off}
987 \define@choicekey{exf@style}{solutionpointsat}%
988   {start,start*,margin,end,manual,off}{\def\exf@solpointsat{#1}}
989 \define@choicekey{exf@style}{solpointsat}%
990   {start,start*,margin,end,manual,off}{\def\exf@solpointsat{#1}}

```

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

```

equationby 991 \defexercisestylearg{problemby}{\exf@numberproblemwithin{#1}}
992 \defexercisestylearg{equationby}{\exf@numberequationwithin{#1}}

```

**pagebysheet** Number pages, problems or equations by sheet:

```

problembysheet
equationbysheet 993 \defexercisestyle{pagebysheet}{%
994   \def\thepage{\csname the\exf@sheetcounter\endcsname.\arabic{page}}%
995   \def\theHpage{\csname theH\exf@sheetcounter\endcsname.\arabic{page}}%
996   \exerciseconfigappend{insertsheetbefore}{\setcounter{page}{1}}
997 \defexercisestyle{problembysheet}%
998   {\exf@numberproblemwithin{\exf@sheetcounter}}
999 \defexercisestyle{equationbysheet}%
1000   {\exf@numberequationwithin{\exf@sheetcounter}}

```

**fracpoints** Use vulgar fractions to display binary fractional points:

```

1001 \defexercisestyle{fracpoints}%
1002   {\exerciseconfig{composepointsnum}[1]{\protect\showfracpoints{##1}}}

```

**twoside** Use two-sided layout for sheets:

```

1003 \defexercisestylearg{true}{twoside}{\exercisestop{twoside={#1}}}

```

**subproblemdelimitem** to append delimiter ‘)’ to subproblem reference labels Use two-sided layout for sheets:

```

1004 \defexercisestyle{subproblemdelimitem}%
1005   {\exerciseconfig{composeitemsubproblemdelim}{}}%
1006   \exerciseconfig{countersubproblem}{\alph{\exf@subproblemcounter}}%
1007   \exerciseconfig{countersubproblemmax}{m}

```

**Extended Exercise Styles.** Declare more specific styles:

```

1008 \ifexf@extstyle

```

**contents** Add sheets and problems to table of contents:

```

1009 \defexercisestyle{contents}{%
1010   \exerciseconfig{toclevelsheet}{section}%
1011   \exerciseconfig{toclevelproblem}{subsection}}

```

`solutionsf` Use sans serif font for solutions:

```
1012 \defexercisestyle{solutionsf}{%
1013   \exerciseconfigappend{styletextsolution}{\sffamily\let\itshape\slshape}}
```

`solutiondimproblem` Dim problem text if solutions are displayed:

```
1014 \defexercisestyle{solutiondimproblem}{%
1015   \RequirePackage{color}%
1016   \exerciseconfigappend{styletextsolution}{\color[gray]{0}}%
1017   \exerciseconfigappend{styletextproblem}{\color[gray]{0.2}}}
```

`solutionsep` Separate solutions by horizontal lines:

```
1018 \defexercisestyle{solutionsep}{%
1019   \exerciseconfig{insertsolutionsbefore}{\hrule\nopagebreak[3]\vspace{0.5ex}}%
1020   \exerciseconfig{insertsolutionsafter}{%
1021     {\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:

```
1022 \defexercisestyle{plainheader}{%
1023   \exerciseconfig{styleheadertitle}{\Large\bfseries}%
1024   \exerciseconfig{styleheadercourse}{\sffamily}%
1025   \exerciseconfig{styleheaderbelow}{\footnotesize}%
1026   \exerciseconfig{skipheaderbelow}{3ex}%
1027   \exerciseconfig{composeheaderbelowleft}{}%
1028   \exerciseconfig{composeheaderbelowright}{}%
1029   \exerciseconfig{composeheaderbelowcenter}{}%
1030   \exerciseconfig{insertsheettitle}{\noindent%
1031     \begin{minipage}{\textwidth}%
1032       {\getexerciseconfig{styleheadertitle}%
1033         \makebox[0pt][l]{\getexercisedata{course}}%
1034         \hfill\makebox[0pt][r]{\getsheetdata{title}}\par}%
1035       {\getexerciseconfig{styleheadercourse}%
1036         \makebox[0pt][l]{\getexercisedata{institution}%
1037           \exercisedataempty{period}{}{\getexercisedata{period}}}%
1038         \hfill\makebox[0pt][r]{\getexercisedata{instructor}}}%
1039       \vphantom{g}\par}%
1040   \hrule%
1041   {\def\tmp{}%
1042     \exerciseconfigempty{composeheaderbelowleft}{}{\def\tmp{.}}%
1043     \exerciseconfigempty{composeheaderbelowcenter}{}{\def\tmp{.}}%
1044     \exerciseconfigempty{composeheaderbelowright}{}{\def\tmp{.}}%
1045     \exerciseifnotempty{\tmp}%
1046     {\getexerciseconfig{styleheaderbelow}\vphantom{\^A}%
1047       \makebox[0pt][l]{\getexerciseconfig{composeheaderbelowleft}}%
1048       \hfill\makebox[0pt][c]{\getexerciseconfig{composeheaderbelowcenter}}%
1049       \hfill\makebox[0pt][r]{\getexerciseconfig{composeheaderbelowright}}%
1050       \vspace*{-\baselineskip}\vspace*{-\parskip}\par}%
1051   \end{minipage}%
1052   \par\addvspace{\getexerciseconfig{skipheaderbelow}}}
```

Done with extended styles:

```
1053 \fi
```

## 3.5 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`:

```
1054 \newcommand{\defexercisedata}[1]{%
1055   \exf@csdo\def{exf@data@#1}{}%
1056   \define@key{exf@data}{#1}%
1057   {\exf@csdo\gdef{exf@data@#1}{##1}}}
```

`\exercisedata` Process key-value pairs:

```
1058 \newcommand{\exercisedata}[1]{\setkeys{exf@data}{#1}}
```

`\getexercisedata` Read global metadata:

```
1059 \newcommand{\getexercisedata}[1]{\csname exf@data@#1\endcsname}
```

`\exercisedataempty` Check whether the field is empty:

```
1060 \newcommand{\exercisedataempty}[3]{\exf@csdo\ifx{exf@data@#1}\exf@empty%
1061   #2\else#3\fi}
```

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

```
1062 \defexercisedata{author}
1063 \defexercisedata{title}
1064 \defexercisedata{subject}
1065 \defexercisedata{keyword}
```

Declare additional general purpose fields:

```
1066 \defexercisedata{date}
```

Declare metadata related to courses:

```
1067 \defexercisedata{instructor}
1068 \defexercisedata{course}
1069 \defexercisedata{institution}
1070 \defexercisedata{period}
1071 \defexercisedata{material}
```

Overwrite standard definitions for `author`, `title`, `date` to also fill ordinary L<sup>A</sup>T<sub>E</sub>X structures:

```
1072 \define@key{exf@data}{author}{\gdef\exf@data@author{#1}\author{#1}}
1073 \define@key{exf@data}{title}{\gdef\exf@data@title{#1}\title{#1}}
1074 \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`:

```
1075 \newcommand{\defsheetdata}[1]{%
1076   \exf@csdo\def{exf@data@sheet@#1}{}%
1077   \define@key{exf@sheet}{#1}%
1078   {\exf@csdo\def{exf@data@sheet@#1}{##1}}}
```

`\setsheetdata` Set sheet metadata:

```
1079 \newcommand{\setsheetdata}[1]{\setkeys{exf@sheet}{#1}}
```

`\getsheetdata` Read sheet metadata:

```
1080 \newcommand{\getsheetdata}[1]{\csname exf@data@sheet@#1\endcsname}
```

`\sheetdataempty` Check whether the field is empty:

```
1081 \newcommand{\sheetdataempty}[3]{\exf@csdo\ifx{exf@data@sheet@#1}\exf@empty%
1082 #2\else#3\fi}
```

Declare general purpose fields:

```
1083 \defsheetdata{due}
1084 \defsheetdata{handout}
1085 \defsheetdata{editdate}
1086 \defsheetdata{author}
1087 \defsheetdata{editor}
```

Special title processing:

```
1088 \def\exf@data@sheet@rawtitle{}
1089 \define@key{exf@sheet}{title}{\def\exf@data@sheet@rawtitle{#1}}
1090 \def\exf@data@sheet@title{\exf@config@composetitlesheet%
1091 {\csname the\exf@sheetcounter\endcsname}{\exf@data@sheet@rawtitle}}%
```

Special points processing:

```
1092 \def\exf@data@sheet@points{\ifdefined\exf@sheet@points%
1093 \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`:

```
1094 \newcommand{\defproblemdata}[1]{%
1095 \exf@csdo\def{exf@data@problem@#1}{}%
1096 \define@key{exf@problem}{#1}%
1097 {\exf@csdo\def{exf@data@problem@#1}{##1}}}
```

`\setproblemdata` Set problem metadata:

```
1098 \newcommand{\setproblemdata}[1]{\setkeys{exf@problem,exf@scanproblem}{#1}}
```

`\setsubproblemdata` Set subproblem metadata:

```
1099 \newcommand{\setsubproblemdata}[1]{%
1100 \setkeys{exf@subproblem,exf@scansubproblem}{#1}}
```

`\getproblemdata` Read problem metadata:

```
1101 \newcommand{\getproblemdata}[1]{\csname exf@data@problem@#1\endcsname}
```

`\problemdataempty` Check whether the field is empty:

```
1102 \newcommand{\problemdataempty}[3]{\exf@csdo\ifx{exf@data@problem@#1}\exf@empty%
1103 #2\else#3\fi}
```

Special title processing:

```
1104 \def\exf@data@problem@rawtitle{}
1105 \define@key{exf@problem}{title}{\def\exf@data@problem@rawtitle{#1}}
1106 \def\exf@data@problem@title{\exf@config@composetitleproblem{%
1107 \cscname the\exf@problemcounter\endcscname}{\exf@data@problem@rawtitle}}%
```

## Problem Environment Code.

`\exf@addmargin` Define a length for environment margin:

```
1108 \newskip\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):

```
1109 \newcommand{\exf@section}[2]{%
1110 \ifdim\glueexpr#1\relax<0pt%
1111 \protected@edef\exf@tmp{#2}%
1112 \else%
1113 \def\exf@tmp{#2}%
1114 \fi%
1115 \exf@exparg{\@startsection{}{}{0pt}{0pt}{#1}{*}}{\exf@tmp}}
```

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

```
1116 \newcommand{\exf@init@block}[1]{%
1117 \def\exf@intro{} \def\exf@intro@skip{#1}%
1118 \exf@addmargin0pt\def\exf@introitem{}}
```

`\exf@append@intro` Append to info buffer:

```
1119 \newcommand{\exf@append@intro}[1]{%
1120 {\exf@append@def\exf@intro{#1\hspace{\exf@intro@skip}}}}
```

`\exf@prepend@intro` Prepend to info buffer:

```
1121 \newcommand{\exf@prepend@intro}[1]{%
1122 {\exf@prepend@def\exf@intro{#1\hspace{\exf@intro@skip}}}}
```

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

```
1123 \newcommand{\exf@open@block}[1]{%
1124 \advance\leftskip\exf@addmargin%
1125 \advance\linewidth-\exf@addmargin%
1126 \advance\@totalleftmargin\exf@addmargin%
1127 \ifx\exf@intro\exf@empty%
1128 \exf@section{0pt}{\exf@introitem}%
1129 \else%
1130 \exf@section{#1}{\exf@introitem\exf@intro\unskip}%
1131 \fi}%
```

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

```
1132 \newcommand{\exf@close@block}{%
1133 \advance\leftskip-\exf@addmargin%
1134 \advance\linewidth\exf@addmargin%
1135 \advance\@totalleftmargin-\exf@addmargin%}
```

`\addprobleminfo` Interface to append or prepend to info buffer:

```
1136 \newcommand{\addprobleminfo}{\@ifstar\exf@prepend@intro\exf@append@intro}
```

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

```
1137 \newcommand{\exf@addinfoswitch}[1]%  
1138   {\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:

```
1139 \newcommand{\defprobleminfo}[2]{%  
1140   \exf@addinfoswitch{#1}%  
1141   \exerciseconfig{compose@probleminfo@#1}[1]{#2}%  
1142   \exf@exparg{\define@key{exf@probleminfo}{#1}}%  
1143   {\csname ifexf@showdata@#1\endcsname\exf@append@intro{%  
1144     \csname exf@config@compose@probleminfo@#1\endcsname{##1}}\fi}}
```

`\showprobleminfo` Process info switches, expand argument first:

```
1145 \newcommand{\showprobleminfo}[1]{\exf@expsetkeys{exf@infoswitch}{#1}}
```

**Problem Info Declarations.** Declare general purpose fields:

```
1146 \defprobleminfo{optional}{\emph{#1:}}  
1147 \showprobleminfo{optional}  
1148 \defprobleminfo{difficulty}{( #1 )}
```

Declare fields for internal information (mostly):

```
1149 \defprobleminfo{comment}{#1}  
1150 \defprobleminfo{author}{\langle$#1$\rangle$}  
1151 \defprobleminfo{editor}{\{ $#1 \}}  
1152 \defprobleminfo{source}{[ #1 ]}  
1153 \defprobleminfo{keyword}{\#( #1 )}
```

Declare more specific fields:

```
1154 \ifexf@extdata  
1155 \defprobleminfo{review}{#1}  
1156 \defprobleminfo{recycle}{[[ #1 ]]}  
1157 \defprobleminfo{timesolve}{\{ #1 \}}  
1158 \defprobleminfo{timepresent}{\{ ! \{ #1 \} ! \}}  
1159 \fi
```

**Write Metadata to PDF Files.**

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

```
1160 \newcommand{\exf@writemetadata}{%  
1161   \ifdefined\hypersetup%
```

Write author, title, subject and keywords:

```
1162   \ifx\exf@data@author\exf@empty\else%  
1163     \hypersetup{pdfauthor={\exf@data@author}}\fi%  
1164   \ifx\exf@data@title\exf@empty\else%  
1165     \hypersetup{pdftitle={\exf@data@title}}\fi%  
1166   \ifx\exf@data@subject\exf@empty\else%
```

```

1167 \hypersetup{pdfsubject={\exf@data@subject}}\fi%
1168 \ifx\exf@data@keyword\exf@empty\else%
1169 \hypersetup{pdfkeywords={\exf@data@keyword}}\fi%
1170 \fi}

```

Automatic writing at `\begin{document}`:

```

1171 \AtBeginDocument{\exf@ifis\exf@metadata{auto}}%
1172 {\exf@writemetadata\gdef\exf@metadata{off}}

```

`\writeexercisedata` Write metadata manually:

```

1173 \newcommand{\writeexercisedata}{\exf@ifis\exf@metadata{manual}}%
1174 {\exf@writemetadata\gdef\exf@metadata{off}}

```

### 3.6 Counters

`sheet` Define main counters (with customised names if necessary) and equation counters:

```

problem 1175 \newcounter{\exf@sheetcounter}
subproblem 1176 \newcounter{\exf@problemcounter}
solution 1177 \newcounter{\exf@subproblemcounter}[\exf@problemcounter]
1178 \newcounter{\exf@solutioncounter}[\exf@problemcounter]
1179 \newcount\exf@eqsav
1180 \newcounter{\exf@sheetequation}
1181 \newcounter{\exf@problemequation}
1182 \newcounter{\exf@solutionequation}

```

Implement counter display; take care of corresponding hyperref labels:

```

1183 \exf@csdo\def{the\exf@sheetcounter}{\exf@config@countersheet}
1184 \exf@csdo\def{the\exf@problemcounter}{\exf@config@counterproblem}
1185 \exf@csdo\def{the\exf@subproblemcounter}{\exf@config@countersubproblem}
1186 \def\theexf@sheetequation{\exf@config@countersheetequation}
1187 \def\theHexf@sheetequation{sheet.\arabic{equation}}
1188 \def\theexf@problemequation{\exf@config@counterproblemequation}
1189 \def\theHexf@problemequation{prob.\arabic{equation}}
1190 \def\theexf@solutionequation{\exf@config@countersolutionequation}
1191 \def\theHexf@solutionequation{sol.\arabic{equation}}

```

`@numberproblemwithin` Declare problem counter as subcounter of #1:

```

1192 \newcommand{\exf@numberproblemwithin}[1]{%
1193 \@addtoreset{\exf@problemcounter}{#1}}%
1194 \exf@csdo\def{the\exf@problemcounter}{%
1195 {\csname the#1\endcsname.\exf@config@counterproblem}}%
1196 \edef\exf@tmp{#1}}%
1197 \ifx\exf@tmp\exf@sheetcounter%
1198 \exerciseconfig{tagproblem}{\ifdefined\sheettag\sheettag-\fi%
1199 \arabic{\exf@problemcounter}}}%
1200 \else%
1201 \exerciseconfig{tagproblem}{\csname the#1\endcsname-%
1202 \arabic{\exf@problemcounter}}}%
1203 \fi}

```

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

```

1204 \newcommand{\exf@numberequationwithin}[1]{%

```

```

1205 \@addtoreset{exf@sheetequation}{#1}%
1206 \def\theexf@sheetequation%
1207   {\csname the#1\endcsname.\exf@config@countersheetequation}%
1208 \def\theHexf@sheetequation%
1209   {\csname theH#1\endcsname.sheet.\arabic{equation}}}%
1210 \@addtoreset{exf@problemequation}{#1}%
1211 \def\theexf@problemequation%
1212   {\csname the#1\endcsname.\exf@config@counterproblemequation}%
1213 \def\theHexf@problemequation%
1214   {\csname theH#1\endcsname.prob.\arabic{equation}}}%
1215 \@addtoreset{exf@solutionequation}{#1}%
1216 \def\theexf@solutionequation%
1217   {\csname the#1\endcsname.\exf@config@countersolutionequation}%
1218 \def\theHexf@solutionequation%
1219   {\csname theH#1\endcsname.sol.\arabic{equation}}}%

```

### 3.7 Buffers

#### File Output.

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

```

1220 \newif\ifexf@solfile@open\exf@solfile@openfalse
1221 \newif\ifexf@probfile@open\exf@probfile@openfalse

```

`\exf@solfile` Reserve file handles:

```

1222 \newwrite\exf@solfile
1223 \newwrite\exf@probfile

```

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

```

1224 \newcommand{\exf@writeline}[2]{\immediate\write#1{#2}}

```

`\exf@linesep` Return a separator line:

```

1225 \newcommand{\exf@linesep}%
1226   {\@percentchar-----}

```

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

```

1227 \newcommand{\exf@lineno}{\@percentchar%
1228   \ifdefined\currfilename\currfilename\space\fi%
1229   1.\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:

```

1230 \newcommand{\exf@start@solfile}[1]{%
1231   \ifexf@solfile@open\else%
1232     \exercisestyle{solutionbelow=manual}%
1233     \global\exf@solfile@opentrue%
1234     \edef\exf@tmp{#1}%
1235     \immediate\openout\exf@solfile\exf@tmp\exf@config@extsolutions\relax%
1236     \exf@writeline\exf@solfile{\@percentchar%
1237       generated from file '\jobname' by exframe.sty}%
1238     \ifexf@lineno\exf@writeline\exf@solfile{\exf@lineno}\fi%

```

```

1239 \exf@writeline\exf@solfile{ }%
1240 \fi}

```

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

```

1241 \newcommand{\exf@close@solfile}{%
1242 \ifexf@solfile@open%
1243 \ifexf@lineno\exf@writeline\exf@solfile{\exf@linesep}%
1244 \exf@writeline\exf@solfile{\exf@lineno}\fi%
1245 \exf@writeline\exf@solfile{\@backslashchar endinput}%
1246 \immediate\closeout\exf@solfile%
1247 \global\exf@solfile@openfalse%
1248 \fi}

```

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

```

1249 \newcommand{\exf@start@probfile}[1]{%
1250 \ifexf@probfile@open\else%
1251 \global\exf@probfile@opentruetrue%
1252 \edef\exf@tmp{#1}%
1253 \immediate\openout\exf@probfile\exf@tmp\exf@config@extproblems\relax%
1254 \exf@writeline\exf@probfile{\@percentchar%
1255 generated from file '\jobname' by exframe.sty}%
1256 \ifexf@lineno\exf@writeline\exf@probfile{\exf@lineno}\fi%
1257 \exf@writeline\exf@probfile{ }%
1258 \fi}

```

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

```

1259 \newcommand{\exf@close@probfile}{%
1260 \ifexf@probfile@open%
1261 \ifexf@lineno\exf@writeline\exf@probfile{\exf@linesep}%
1262 \exf@writeline\exf@probfile{\exf@lineno}\fi%
1263 \exf@writeline\exf@probfile{\@backslashchar endinput}%
1264 \immediate\closeout\exf@probfile%
1265 \global\exf@probfile@openfalse%
1266 \fi}

```

Make sure to properly close files at the end:

```

1267 \AtEndDocument{\exf@close@solfile\exf@close@probfile}

```

## Buffers.

`\exf@solbuf` Declare token buffers for storing problems and solutions and conditionals indicating whether the buffers have been used:

```

\exf@subprobbuf
\ifexf@solbuf@clean 1268 \newtoks\exf@solbuf
\ifexf@probbuf@clean 1269 \newtoks\exf@probbuf
\ifexf@subprobbuf@clean 1270 \newtoks\exf@subprobbuf
1271 \newif\ifexf@solbuf@clean\exf@solbuf@cleantrue
1272 \newif\ifexf@probbuf@clean\exf@probbuf@cleantrue

```

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

```

\exf@clear@probbuf
\exf@clear@subprobbuf 1273 \def\exf@clear@solbuf{\global\exf@solbuf@cleantrue\global\exf@solbuf={}}
1274 \def\exf@clear@probbuf{\global\exf@probbuf@cleantrue\global\exf@probbuf={}}
1275 \def\exf@clear@subprobbuf{\global\exf@subprobbuf={}}

```

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

```
1276 \def\exf@append@buf#1#2{\global#1=\expandafter{\the#1#2}}
```

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

```
1277 \def\exf@addline#1#2{\protected@edef\exf@tmp{#2}%
1278 \exf@exparg{\exf@append@buf#1}{\exf@tmp^^J}}
```

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

```
1279 \def\exf@source@buf#1{\exf@exptwo\scantokens{\the#1}}
```

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

```
1280 \def\exf@write@buf#1#2{\exf@writeline#1{\the#2}}
```

### Verbatim Processing.

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

```
1281 \newcommand{\exf@verbatim}{%
1282 \begingroup%
1283 \@bsphack%
1284 \let\do\@makeother\dospecials%
1285 \catcode'\^^M\active%
1286 \def\verbatim@processline{\exf@exptwo\exf@verbatim@process%
1287 {\the\verbatim@line^^J}}%
1288 \verbatim@start}
```

`\exf@endverbatim` Stop reading the buffer:

```
1289 \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:

```
1290 \def\exf@scanblock#1{%
```

Check for empty first line:

```
1291 \@ifnextchar\par{\exf@scanblock@par{#1}}{\exf@scanblock@sel{#1}}
```

Handle empty first line, implies no optional argument:

```
1292 \long\def\exf@scanblock@par#1\par{\exf@scanblock@sel{#1}[]}
```

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

```
1293 \def\exf@scanblock@sel#1{\@ifnextchar[{\exf@scanblock@opt{#1}}%
1294 {\@ifnextchar\end{\exf@scanblock@end{#1}}{\exf@scanblock@noopt{#1}}}
```

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

```
1295 \def\exf@scanblock@end#1\end#2{%
1296 \def\exf@tmp{#2}\ifx\exf@tmp\@currentenv%
1297 \def\exf@verbatim{\def\exf@endverbatim}{%
1298 \fi%
1299 #1}{\scantokens{\end{#2}}}
```

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

```
1300 \def\exf@scanblock@noopt#1#2{#1}{\scantokens#2}}
1301 \def\exf@scanblock@opt#1[#2]{#1{#2}{}}
```

## 3.8 Points

### Points Arithmetic.

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

```
1302 \newcommand{\exf@addtopoints}[3]{%
1303   \ifdefined#1\else\def#1{{0}{0}}\fi%
1304   \dimen@dimexpr\expandafter\@firstoftwo#1pt\relax%
1305   \advance\dimen@\dimexpr#2pt\relax%
1306   \edef\exf@tmp{\strip@pt\dimen@}%
1307   \dimen@\dimexpr\expandafter\@secondoftwo#1pt\relax%
1308   \advance\dimen@\dimexpr#3pt%
1309   \xdef#1{\exf@tmp}{\strip@pt\dimen@}}}
```

`\exf@pointsmismatch` Execute #3 if points disagree:

```
1310 \newcommand{\exf@pointsmismatch}[3]{%
1311   \let\exf@tmp\undefined%
1312   \ifdim\expandafter\@firstoftwo#1pt=\expandafter\@firstoftwo#2pt\else%
1313     \def\exf@tmp{}\fi%
1314   \ifdim\expandafter\@secondoftwo#1pt=\expandafter\@secondoftwo#2pt\else%
1315     \def\exf@tmp{}\fi%
1316   \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:

```
1317 \def\exf@outpoints#1#2#3{\ifdefined#3%
1318   \exf@exptwo\exf@outpoints@switch{#3}{#1}{#2}%
1319   \global\let#3\undefined\fi}
1320 \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:

```
1321 \def\exf@scanpoints#1#2+#3+#4{%
1322   \edef#1{\if @#2@0\else#2\fi}%
1323   \edef#1{#1}{\if @#3@0\else#3\fi}}
```

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

```
1324 \def\exf@formatpoints#1#2{\ifdim#2pt=0pt#1\else%
1325   \ifdim#1pt=0pt+#2\else#1+#2\fi\fi}
```

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

```
1326 \newcommand{\extractpoints}{\@ifstar{\exf@extractpoints\@secondoftwo}%
1327   {\exf@extractpoints\@firstoftwo}}
1328 \newcommand{\exf@extractpoints}[2]{\edef\exf@tmp{#2}%
1329   \exf@exptwo\exf@scanpoints\exf@tmp\exf@tmp++@%
1330   \expandafter#1\exf@tmp}
```

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

```
1331 \newcommand{\switchpoints}[5]{\edef\exftmp{#5}%
1332 \exftwo\exfsanpoints\exftmp\exftmp++@%
1333 \expandafter\exfswitchpoints\exftmp{#1}{#2}{#3}{#4}}
1334 \newcommand{\exfswitchpoints}[6]{%
1335 \ifdim#2pt=0pt\ifdim#1pt=0pt\def\exftmp##1##2{#6}%
1336 \else\def\exftmp##1##2{#3}\fi%
1337 \else\ifdim#1pt=0pt\def\exftmp##1##2{#4}%
1338 \else\def\exftmp##1##2{#5}\fi\fi\exftmp{#1}{#2}}
```

## Tools.

`\pointsmarginproblem` Combination to typeset points in margin:

`\intsmarginsubproblem`

`\pointsmarginssolution`

```
1339 \newcommand{\exfmakepointsmarginproblem}[2]{%
1340 \exfconfig@insertpointsmarginproblem{%
1341 \exfconfig@composepointspairmarginproblem{#1}{#2}}}
1342 \newcommand{\exfmakepointsmarginssubproblem}[2]{%
1343 \exfconfig@insertpointsmarginssubproblem{%
1344 \exfconfig@composepointspairmarginssubproblem{#1}{#2}}}
1345 \newcommand{\exfmakepointsmarginssolution}[2]{%
1346 \exfconfig@insertpointsmarginssolution{%
1347 \exfconfig@composepointspairmarginssolution{#1}{#2}}}
```

`\exfwarnmismatch` If points #3 and #4 are defined and disagree, issue a warning message:

```
1348 \newcommand{\exfwarnmismatch}[4]{%
1349 \ifdefined#4\ifdefined#3\exfpointsmismatch#3#4{%
1350 \let\exftmp\PackageWarning%
1351 \ifx#1\exfsolutionname\let\exftmp\PackageWarningNoLine\fi%
1352 \exftmp{exframe}{points mismatch %
1353 (\expandafter\exfformatpoints#3 determined %
1354 vs. \expandafter\exfformatpoints#4 given) %
1355 for #1 \csname the#2\endcsname}%
1356 \ifexfwarntext\edef\exftmp{%
1357 {\expandafter\exfformatpoints#3}{\expandafter\exfformatpoints#4}}%
1358 \exftwo\exfconfig@insertwarnpoints#1\exftmp\fi}%
1359 \fi\fi}
```

`\exfwarnrerun` If points #3 and #4 are defined and disagree, issue advice to recompile:

```
1360 \newcommand{\exfwarnrerun}[4]{%
1361 \ifdefined#4\ifdefined#3\exfpointsmismatch#3#4{%
1362 \PackageWarning{exframe}{points changed %
1363 for #1 \csname the#2\endcsname; rerun to fix}%
1364 \ifexfwarntext\exfconfig@insertwarnpointsrerun#1\fi}%
1365 \fi\fi}
```

## Binary Rational Numbers.

`\exfsplitsign` Split a decimal float number into sign, integer and fractional part:

`\exfsplitdecimal`

```
1366 \def\exfsplitsign#1-#2-#3&{\def\exfplitnum{#1#2}\def\exfplitminus{#3}}
1367 \def\exfsplitdecimal#1.#2.#3&{\def\exfplitint{#1}\def\exfplitdec{#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:

```

1368 \newcommand{\showfracpoints}[1]{%
1369   \edef\exftmp{#1}%
1370   \expandafter\exfsplitsign\exftmp--&%
1371   \expandafter\exfsplitdecimal\exfsplitnum. .&%
1372   \if @\exfsplitint @\def\exfsplitint{0}\fi%
1373   \if @\exfsplitdec @\def\exfsplitdec{0}\fi%
1374   \def\exftmp{\exfsplitint.\exfsplitdec}%
1375   \ifnum\exfsplitdec=0\def\exftmp{\exfsplitint}\fi%
1376   \ifnum\exfsplitdec=5\def\exftmp{\exfconfigfrac{\exfsplitint}{1}{2}}\fi%
1377   \ifnum\exfsplitdec=25\def\exftmp{\exfconfigfrac{\exfsplitint}{1}{4}}\fi%
1378   \ifnum\exfsplitdec=75\def\exftmp{\exfconfigfrac{\exfsplitint}{3}{4}}\fi%
1379   \ifnum\exfsplitdec=125\def\exftmp{\exfconfigfrac{\exfsplitint}{1}{8}}\fi%
1380   \ifnum\exfsplitdec=375\def\exftmp{\exfconfigfrac{\exfsplitint}{3}{8}}\fi%
1381   \ifnum\exfsplitdec=625\def\exftmp{\exfconfigfrac{\exfsplitint}{5}{8}}\fi%
1382   \ifnum\exfsplitdec=875\def\exftmp{\exfconfigfrac{\exfsplitint}{7}{8}}\fi%
1383   \ifx\exfsplitminus\exfempty\else$\exfsplitminus$\fi\exftmp}

```

`\exfconfigfrac` Display a vulgar fraction such as  $12^3/4$ :

```

1384 \newcommand{\exfconfigfrac}[3]{%
1385   \ifnum#1=0\relax\else#1\fi%
1386   \ifnum#2=0\relax\else$%
1387   ~{\exftext{#2}}%
1388   \mskip-4mu/\mskip-2mu%
1389   _{\exftext{#3}}$\fi}

```

### Sheet Points Code.

`notedata@sheetpoints` Store a sheet point number in a macro:

```

1390 \newcommand{\exfnotedata@sheetpoints}[2]{%
1391   \exfcsdo\gdef{\exfsheetpoints@#1}{#2}}

```

`exfwritesheetpoints` Write sheet points to the .aux file:

```

1392 \newcommand{\exfwritesheetpoints}[2]{%
1393   {\exfwritedata{sheetpoints}{\sheettag}{\exfformatpoints{#1}{#2}}}

```

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

```

1394 \newcommand{\getsheetpoints}[1]{\if @#1%
1395   \ifdefined\exfsheet@points%
1396     \expandafter\exfformatpoints\exfsheet@points\else 0\fi%
1397   \else\exfcsor{\exfsheetpoints@#1}{0}\fi}

```

### Problem Points Code.

`tedata@problempoints` Store a problem point number in a macro:

```

1398 \newcommand{\exfnotedata@problempoints}[2]{%
1399   \exfcsdo\gdef{\exfproblempoints@#1}{#2}}

```

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

```

1400 \newcommand{\exfwriteproblempoints}[2]{%
1401   {\exfwritedata{problempoints}{\problemtag}{\exfformatpoints{#1}{#2}}}

```

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

```
1402 \newcommand{\getproblempoints}[1]{\if @#1%
1403   \ifdefined\exf@problem@points%
1404     \expandafter\exf@formatpoints\exf@problem@points\else 0\fi%
1405   \else\exf@csor{exf@problempoints@#1}{0}\fi}
```

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

```
1406 \newcommand{\showpoints}{%
1407   \ifdefined\exf@in@solution\exf@outpoints{\exf@ensuretext}%
1408   {\exf@config@composepointspairbodysolution}{\exf@solution@points@show}%
1409   \else\ifdefined\exf@in@subproblem\exf@outpoints{\exf@ensuretext}%
1410   {\exf@config@composepointspairbodysubproblem}{\exf@subproblem@points@show}%
1411   \else\ifdefined\exf@in@problem\exf@outpoints{\exf@ensuretext}%
1412   {\exf@config@composepointspairbodyproblem}{\exf@problem@points@show}%
1413   \fi\fi\fi}
```

### Subproblem Points Code.

`\notedata@subproblempoints` Store a subproblem point number in a macro:

```
1414 \newcommand{\exf@notedata@subproblempoints}[2]{%
1415   \exf@csdo\gdef{exf@subproblempoints@#1}{#2}}
```

`\writesubproblempoints` Write subproblem points to the .aux file:

```
1416 \newcommand{\exf@writesubproblempoints}[2]{%
1417   {\exf@writedata{subproblempoints}%
1418     {\subproblemtag}{\exf@formatpoints{#1}{#2}}}
```

`\getsubproblempoints` Read points for current subproblem:

```
1419 \newcommand{\getsubproblempoints}[1]{\if @#1%
1420   \ifdefined\exf@subproblem@points%
1421     \expandafter\exf@formatpoints\exf@subproblem@points\else 0\fi%
1422   \else\exf@csor{exf@subproblempoints@#1}{0}\fi}
```

### Solution Points Code.

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

```
1423 \newcommand{\exf@awardpointsalt}[2] [] {\exf@scanpoints\exf@tmp#2++@%
1424   \exf@exptwo\exf@ensuretext{%
1425     \expandafter\exf@config@composepointspairawardalt\exf@tmp{#1}}}
```

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

```
1426 \newcommand{\exf@awardpointsreg}[2] [] {\exf@scanpoints\exf@tmp#2++@%
1427   \exf@exptwo\exf@addtopoints\exf@solution@points@total\exf@tmp%
1428   \exf@scanpoints\exf@tmp#2++@%
1429   \exf@exptwo\exf@ensuretext{%
1430     \expandafter\exf@config@composepointspairaward\exf@tmp{#1}}}
```

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

```
1431 \newcommand{\awardpoints}{\@ifstar\exf@awardpointsalt\exf@awardpointsreg}
```

`\getsolutionpoints` Read points for current solution:

```
1432 \newcommand{\getsolutionpoints}[1]{\if @#1%
1433   \ifdefined\exf@solution@points%
1434     \expandafter\exf@formatpoints\exf@solution@points\else 0\fi%
1435   \else 0\fi}
```

### 3.9 Tag Lists

#### Loop Lists.

`exerciseloop` Counters for item number within list and depth of loop nesting:

`exf@loopdepth`

```
1436 \newcounter{exerciseloop}
1437 \newcounter{exf@loopdepth}
```

`exf@listwalk` Step through a list; call the callback function #1 with current item #2:

```
1438 \def\exf@listwalk#1#2{\if @#2@\def\exf@tmp{}\else%
1439   \def\exf@tmp{#1{#2}\exf@listwalk#1}\fi\exf@tmp}
```

`\exerciseloop` Loop through a list of items in braces #1 which is expanded first; store code #2 to be executed in a callback function (need to use different callback function for each loop depth, callback function must be global to work within table cells); initialise item counter:

```
1440 \newcommand{\exerciseloop}[2]{\addtocounter{exf@loopdepth}{1}%
1441   \setcounter{exerciseloop}{0}%
1442   \exf@csdo\gdef{exf@listcallback@\roman{exf@loopdepth}}##1%
1443     {\stepcounter{exerciseloop}#2}%
1444   \edef\exf@tmp{#1}%
1445   \exf@csdotwo\exf@exptwo\exf@listwalk%
1446     {exf@listcallback@\roman{exf@loopdepth}}\exf@tmp}%
1447   \addtocounter{exf@loopdepth}{-1}}
```

`\exerciseloopstr` Loop through list and save result to `\exerciseloopret` (or optional argument #1):

```
1448 \newcommand{\exerciseloopstr}[3][\exerciseloopret]{%
1449   \def#1{\exerciseloop{#2}{\protected@edef#1{#1#3}}}
```

#### Lists Management.

`\exf@notedata@sheet` Store a sheet tag:

```
1450 \def\exf@sheetlist{}
1451 \newcommand{\exf@notedata@sheet}[2]{%
1452   \xdef\exf@sheetlist{\exf@sheetlist{#1}}}
```

`exf@notedata@problem` Store a problem tag:

```
1453 \def\exf@problemlist{}
1454 \newcommand{\exf@notedata@problem}[2]{%
1455   \xdef\exf@problemlist{\exf@problemlist{#1}}%
1456   \if @#2@\else%
1457     \ifcsname exf@problemlist@#2\endcsname\else%
1458       \exf@csdo\gdef{exf@problemlist@#2}{}\fi%
1459     \exf@csdo\xdef{exf@problemlist@#2}%
1460     {\csname exf@problemlist@#2\endcsname{#1}}%
1461   \fi}
```

`@notedata@subproblem` Store a subproblem tag:

```
1462 \newcommand{\exf@notedata@subproblem}[2]{%
1463   \ifcsname exf@subproblemlist@#2\endcsname\else%
1464     \exf@csdo\gdef{exf@subproblemlist@#2}{}\fi%
1465   \exf@csdo\xdef{exf@subproblemlist@#2}%
1466   {\csname exf@subproblemlist@#2\endcsname{#1}}}
```

`\getsheetlist` Get list of sheet tags, problem tags (all, within current or particular sheet), subproblem

`\getproblemlist` tags (within current or particular problem):

`\getsubproblemlist`

```
1467 \newcommand{\getsheetlist}[1]{\exf@sheetlist}
1468 \newcommand{\getproblemlist}[1]{\if @#1%
1469   \ifdefined\sheettag\exf@csor{exf@problemlist@\sheettag}{}%
1470   \else\exf@problemlist\fi%
1471   \else%
1472   \if *#1\exf@problemlist\else\exf@csor{exf@problemlist@#1}{}\fi%
1473   \fi}
1474 \newcommand{\getsubproblemlist}[1]{\if @#1%
1475   \exf@csor{exf@subproblemlist@problemtag}{}\else%
1476   \exf@csor{exf@subproblemlist@#1}{}\fi}
```

### 3.10 Sheet Environment

`exf@sheet` Define options for sheet environment:

```
1477 \define@key{exf@sheet}{points}{\exf@scanpoints\exf@sheet@points#1++@}
1478 \define@key{exf@sheet}{number}{\setcounter{\exf@sheetcounter}{#1}%
1479   \addtocounter{\exf@sheetcounter}{-1}\refstepcounter{\exf@sheetcounter}}
1480 \define@key{exf@sheet}{label}{\def\exf@label{#1}}
1481 \define@key{exf@sheet}{tag}{\def\sheettag{#1}}
```

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

```
1482 \newenvironment{\exf@sheetname}[1] [] {%
```

Insert hook code to clear page, step counter:

```
1483   \exf@config@insertsheetclearpage%
1484   \refstepcounter{\exf@sheetcounter}%
```

Use equation counter for sheets:

```
1485   \ifexf@style@sheetequation%
1486     \exf@eqsav\value{equation}\relax%
1487     \setcounter{equation}{\value{exf@sheetequation}}%
1488     \let\theequation\theexf@sheetequation%
1489     \let\theHequation\theHexf@sheetequation%
1490   \fi%
```

Reset optional arguments, process arguments:

```
1491   \let\exf@sheet@points\@undefined%
1492   \def\sheettag{\getexerciseconfig{tagsheet}}%
1493   \let\exf@sheet@points@total\@undefined%
1494   \let\exf@label\@undefined%
1495   \setkeys{exf@sheet}{#1}%
```

Process automatic and manual labels:

```
1496   \ifexf@autolabelsheet\label{\exf@config@labelsheet{\sheettag}}\fi%
```

```

1497 \ifdefined\exf@label\label{\exf@label}\fi%
1498 \exf@writedata{sheet}{\sheettag}{}%

```

Set points from explicit input or from .aux storage:

```

1499 \ifdefined\exf@sheet@points%
1500 \let\exf@sheet@points@given\exf@empty%
1501 \else%
1502 \let\exf@sheet@points@given\@undefined%
1503 \ifcsname exf@sheetpoints@\sheettag\endcsname%
1504 \exf@csdotwo\let\exf@tmp{\exf@sheetpoints@\sheettag}%
1505 \exf@exptwo\exf@scanpoints\exf@sheet@points\exf@tmp++@%
1506 \fi\fi%

```

Process metadata:

```

1507 \exf@ifis\exf@metadata{sheet}{{%
1508 \ifx\exf@data@sheet@author\exf@empty\else%
1509 \let\exf@data@author\exf@data@sheet@author\fi%
1510 \def\exf@data@title{\exf@config@composemetasheet%
1511 {\csname the\exf@sheetcounter\endcsname}{\exf@data@sheet@rawtitle}}%
1512 \exf@writemetadata}\gdef\exf@metadata{off}}%

```

Insert hook code:

```

1513 \exf@config@insertsheetbefore%

```

Add table of contents line:

```

1514 \ifx\exf@config@toclevelsheet\exf@empty\else%
1515 \ifdefined\phantomsection\phantomsection\fi\fi%
1516 \exf@addcontentsline{\exf@config@toclevelsheet}%
1517 {\exf@config@composetocsheet{\csname the\exf@sheetcounter\endcsname}%
1518 {\exf@data@sheet@rawtitle}}%

```

Write sheet title:

```

1519 \exf@config@insertsheettitle}%

```

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

```

1520 {\ifdefined\exf@sheet@points@given%
1521 \exf@warnmismatch{\exf@sheetname}{\exf@sheetcounter}%
1522 {\exf@sheet@points@total}{\exf@sheet@points}%

```

Test whether points have changed since last compile:

```

1523 \else%
1524 \exf@warnrerun{\exf@sheetname}{\exf@sheetcounter}%
1525 {\exf@sheet@points@total}{\exf@sheet@points}%

```

Store points:

```

1526 \let\exf@sheet@points\exf@sheet@points@total%
1527 \fi%

```

Write sheet points to .aux file:

```

1528 \ifdefined\exf@sheet@points%
1529 \expandafter\exf@writesheetpoints\exf@sheet@points%
1530 \fi%

```

Insert solutions:

```

1531 \exf@ifis\exf@solutionbelow{sheet}{\insertsolutions}%

```

Insert hook code:

```
1532 \exf@config@insertsheetafter%
1533 \exf@config@insertsheetclearpage%
```

Restore original equation counter:

```
1534 \ifexf@style@sheetequation%
1535 \setcounter{exf@sheetequation}{\value{equation}}%
1536 \setcounter{equation}{\exf@eqsav}%
1537 \fi%
```

Done:

```
1538 \ignorespacesafterend}
```

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

```
1539 \newcommand{\exercisecleardoublepage}{%
1540 \clearpage\ifexf@twoside\ifodd\value{page}\else%
1541 \thispagestyle{empty}\hbox{} \newpage\fi\fi}
```

### 3.11 Problem Environment

#### Print Problems.

`exf@problem` Define options for problem environment:

```
1542 \define@key{exf@problem}{points}{\exf@scanpoints\exf@problem@points#1++@}
1543 \define@key{exf@problem}{label}{\def\exf@label{#1}}
1544 \define@key{exf@problem}{tag}{\def\problemtag{#1}}
1545 \define@key{exf@problem}{sollabel}{\xdef\exf@sollabel{#1}}
```

`printproblem` Define `printproblem` environment:

```
1546 \newenvironment{printproblem}[1]{%
```

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

```
1547 \par\exf@config@styletext\addvspace{\exf@config@skipproblemabove}}%
```

Step problem counter:

```
1548 \refstepcounter{\exf@problemcounter}}%
```

Insert hook code:

```
1549 \exf@config@insertproblembefore%
```

Begin inner group, mark in problem:

```
1550 \begingroup%
1551 \def\exf@in@problem{}}%
```

Use equation counter for problems:

```
1552 \ifexf@style@problemequation%
1553 \exf@eqsav\value{equation}\relax%
1554 \setcounter{equation}{\value{exf@problemequation}}%
1555 \let\theequation\theexf@problemequation%
1556 \let\theHequation\theHexf@problemequation%
1557 \fi%
```

Initialise variables, process arguments:

```
1558 \exf@init@block{\exf@config@skipprobleminfo}%
1559 \def\problemtag{\getexerciseconfig{tagproblem}}%
1560 \let\exf@problem@points\@undefined%
1561 \let\exf@label\@undefined%
1562 \global\let\exf@sollabel\@undefined%
1563 \let\exf@problem@points@total\@undefined%
1564 \setkeys{exf@problem,exf@probleminfo,exf@scanproblem}{#1}%
```

Process automatic and manual labels:

```
1565 \ifexf@autolabelproblem\label{\exf@config@labelproblem{\problemtag}}\fi%
1566 \ifdefined\exf@label\label{\exf@label}\fi%
1567 \exf@writedata{problem}{\problemtag}{\ifdefined\sheettag\sheettag\fi}%
```

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

```
1568 \global\let\exf@problem@solfirst\exf@empty%
1569 \xdef\exf@prevprob{\csname the\exf@problemcounter\endcsname}%
1570 \ifcsname the\exf@problemcounter\endcsname%
1571 \xdef\exf@prevprobhref{\exf@problemcounter.%
1572 \csname the\exf@problemcounter\endcsname}%
1573 \fi%
1574 \ifx\exf@data@problem@rawtitle\exf@empty%
1575 \global\let\exf@prevprobttitle\@undefined%
1576 \else%
1577 \protected\xdef\exf@prevprobttitle{\exf@data@problem@rawtitle}%
1578 \fi%
1579 \exf@clear@prevsubprob%
```

Set points from explicit input or from .aux storage:

```
1580 \ifdefined\exf@problem@points%
1581 \let\exf@problem@points@given\exf@empty%
1582 \else%
1583 \let\exf@problem@points@given\@undefined%
1584 \ifcsname exf@problempoints@\problemtag\endcsname%
1585 \exf@csdotwo\let\exf@tmp{\exf@problempoints@\problemtag}%
1586 \exf@exptwo\exf@scanpoints\exf@problem@points\exf@tmp++@%
1587 \fi\fi%
1588 \global\let\exf@prevpoints\exf@problem@points%
1589 \let\exf@problem@points@show\@undefined%
1590 \ifdefined\exf@problem@points%
1591 \let\exf@problem@points@show\exf@problem@points%
1592 \fi%
```

Disable points display if desired:

```
1593 \exf@ifis\exf@pointssat{off}{\let\exf@problem@points@show\@undefined}%
```

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

```
1594 \exf@ifis\exf@pointssat{start}{\exf@outpoints{\exf@append@intro}%
1595 {\exf@config@composepointspairstartproblem}{\exf@problem@points@show}}%
1596 \exf@ifis\exf@pointssat{start*}{\exf@outpoints{\exf@prepend@intro}%
1597 {\exf@config@composepointspairstartproblem}{\exf@problem@points@show}}%
```

Insert hook code, set problem body style:

```
1598 \exf@config@insertprobleminfo%
1599 \exf@config@styletextproblem%
```

Write title without item:

```
1600 \exf@addmargin\glueexpr\exf@config@skipproblemitem\relax%
1601 \ifdim\exf@addmargin=0pt%
1602 \exf@prepend@intro{%
1603 \exf@config@styletitle\exf@config@styletitleproblem%
1604 \exf@config@composetitleproblem{\csname the\exf@problemcounter\endcsname}%
1605 {\exf@data@problem@rawtitle}}%
```

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

```
1606 \else%
1607 \ifdim\exf@addmargin<0pt%
1608 \settowidth\exf@addmargin{%
1609 \exf@config@styletitle\exf@config@styletitleproblem%
1610 \exf@config@composeitemproblem{\exf@config@counterproblemmax}%
1611 \exf@config@composeitemproblemsep}%
1612 \fi%
```

Define item label:

```
1613 \def\exf@introitem{\makebox[0cm][r]{%
1614 \exf@config@styletitle\exf@config@styletitleproblem%
1615 \exf@config@composeitemproblem{\csname the\exf@problemcounter\endcsname}%
1616 \exf@config@composeitemproblemsep}}%
```

Compose title:

```
1617 \ifx\exf@data@problem@rawtitle\exf@empty\else%
1618 \exf@prepend@intro{%
1619 \exf@config@styletitle\exf@config@styletitleproblem%
1620 \exf@config@composetitleproblem{\exf@empty}{\exf@data@problem@rawtitle}}%
1621 \fi%
1622 \fi%
```

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

```
1623 \exf@ifis\exf@pointsat{margin}{%
1624 \exf@outpoints{\exf@prepend@def\exf@introitem}%
1625 {\exf@makepointsmarginproblem}{\exf@problem@points@show}}%
```

Write out opening line:

```
1626 \exf@open@block{\exf@config@skipproblemtitle}%
```

Add table of contents line:

```
1627 \exf@addcontentsline{\exf@config@toclevelproblem}%
1628 {\exf@config@composetocproblem{\csname the\exf@problemcounter\endcsname}%
1629 {\exf@data@problem@rawtitle}}%
```

Done:

```
1630 \@afterindentfalse%
```

End environment, show points if desired:

```
1631 {\exf@ifis\exf@pointsat{end}{\showpoints}}%
```

Perform sanity checks on total points if given explicitly:

```
1632 \ifdefined\exf@problem@points@given%
1633 \exf@warnmismatch{\exf@problemname}{\exf@problemcounter}%
1634 {\exf@problem@points@total}{\exf@problem@points}}%
```

Warn if calculated total points have changed:

```
1635 \else%
1636 \exf@warnrerun{\exf@problemmname}{\exf@problemcounter}%
1637 {\exf@problem@points@total}{\exf@problem@points}%
```

Read computed total points:

```
1638 \let\exf@problem@points\exf@problem@points@total%
1639 \fi%
```

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

```
1640 \ifdefined\exf@problem@points%
1641 \expandafter\exf@writeproblem@points\exf@problem@points%
1642 \exf@exptwo\exf@addtopoints\exf@sheet@points@total\exf@problem@points%
```

Warn if no points given for present problem but previously:

```
1643 \else\ifdefined\exf@sheet@points@total%
1644 \PackageWarning{exframe}{no points defined for \exf@problemmname}%
1645 \fi\fi%
```

Solutions to subproblems must be declared within problem environment:

```
1646 \exf@clear@prevsubprob%
```

End paragraph and environment:

```
1647 \par\exf@close@block%
```

Display solution if desired:

```
1648 \exf@ifis\exf@solutionbelow{problem}{-%
1649 \exf@config@insertproblemsolution%
1650 \exf@showsolutions{\exf@config@composetitlesolutionmulti}{}}%
```

Restore original equation counter:

```
1651 \ifexf@style@problemequation%
1652 \setcounter{exf@problemequation}{\value{equation}}%
1653 \setcounter{equation}{\exf@eqsav}%
1654 \fi%
```

End inner group:

```
1655 \endgroup%
```

Insert hook code, vertical skip:

```
1656 \exf@config@insertproblemafter%
1657 \addvspace{\exf@config@skipproblembelow}%
```

Display solution if desired:

```
1658 \exf@ifis\exf@solutionbelow{problem*}{-%
1659 \exf@showsolutions{\exf@config@composetitlesolutionmulti}{}}%
```

Done:

```
1660 \ignorespacesafterend}
```

**Read Problem Environment.**

`exf@scanproblem` Define options for scanning problem environment:

```
1661 \define@boolkey{exf@scanproblem}[exf@scanproblem@]{disable}[true]{}
```

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

```
1662 \newenvironment{exf@problem@direct}[1] []%
1663 {\printproblem{#1}}{\endprintproblem\ignorespacesafterend}
```

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

```
1664 \newenvironment{exf@problem@scan}%
1665 {\exf@scanblock{\exf@scanproblem}}{\endexf@scanproblem}%
1666 \newenvironment{exf@scanproblem}[2]{}%
```

Determine problem display:

```
1667 \global\exf@scanproblem@disablefalse%
1668 \setkeys*{exf@scanproblem}{#1}%
1669 \exf@config@insertproblemselect{#1}%
1670 \ifexf@scanproblem@disable\global\exf@scanproblem@disabletrue\fi%
```

Circumvent problem display:

```
1671 \ifexf@scanproblem@disable%
1672 \def\exf@verbatim@process{\@gobble}%
1673 \else%
```

Write separator and `printproblem` environment to buffer:

```
1674 \ifexf@lineno\exf@addline\exf@probbuf{\exf@linesep}%
1675 \exf@addline\exf@probbuf{\exf@lineno}\fi%
1676 \exf@addline\exf@probbuf%
1677 {\@backslashchar begin{printproblem}{#1}}%
```

Start verbatim processing:

```
1678 \def\exf@verbatim@process{\exf@append@buf\exf@probbuf}%
1679 \fi%
1680 \exf@verbatim#2}%
```

End verbatim processing; close `printproblem` environment:

```
1681 {\exf@endverbatim%
1682 \ifexf@scanproblem@disable\else%
1683 \exf@addline\exf@probbuf{\@backslashchar end{printproblem}}}%
1684 \global\exf@probbuf@cleanfalse%
1685 \fi%
```

Write buffer to file if output file open:

```
1686 \ifexf@probfile@open%
1687 \exf@write@buf\exf@probfile\exf@probbuf%
1688 \exf@clear@probbuf%
1689 \fi%
```

Output buffer immediately:

```
1690 \ifexf@problemmanual\else%
1691 \exf@source@buf\exf@probbuf%
1692 \exf@clear@probbuf%
1693 \fi%
```

Done:

```
1694 \ignorespacesafterend}
```

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

```
1695 \newenvironment{\exf@problemname}%  
1696 {\ifexf@problembuf\let\exf@tmp\exf@problem@scan%  
1697 \else\let\exf@tmp\exf@problem@direct\fi%  
1698 \exf@tmp}%  
1699 {\ifexf@problembuf\let\exf@tmp\endexf@problem@scan%  
1700 \else\let\exf@tmp\endexf@problem@direct\fi%  
1701 \exf@tmp}
```

## 3.12 Problem Blocks

### Problem Block Handling.

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

```
1702 \newcommand{\exf@problemstitle}{%
```

Check whether title is empty:

```
1703 \protected@edef\exf@problemstitleexp{\exf@config@composetitleproblems}%  
1704 \ifx\exf@problemstitleexp\exf@empty\else%
```

Output section line:

```
1705 \exf@section{\exf@config@skipproblemstitle}%  
1706 {\exf@config@styletitle\exf@config@styletitleproblems%  
1707 \exf@problemstitleexp}%  
1708 \exf@addcontentsline{\exf@config@toclevelproblems}%  
1709 {\exf@config@composetocproblems}%  
1710 \fi}
```

**\exf@showproblemsin** Output problem block intro:

```
1711 \newcommand{\exf@showproblemsin}{%
```

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

```
1712 \par\exf@config@styletext\advvspace{\exf@config@skipproblemsabove}%  
1713 \exf@config@insertproblemsbefore}
```

**\exf@showproblemsout** Output problem block outro:

```
1714 \newcommand{\exf@showproblemsout}{%
```

Insert hook code; close paragraph; add vertical space:

```
1715 \exf@config@insertproblemsafter%  
1716 \par\exf@config@styletext\advvspace{\exf@config@skipproblemsbelow}}
```

**\exf@showproblems** Output problem block in buffer:

```
1717 \newcommand{\exf@showproblems}{%
```

Do nothing if buffer is empty (avoid titles):

```
1718 \ifexf@probbuf@clean\else\beginngroup%
```

Execute output problem block intro:

```
1719 \ex@showproblemsin%
1720 \ex@problemstitle%
```

Source and clear buffer:

```
1721 \ex@source@buf\ex@probbuf%
1722 \ex@clear@probbuf%
```

Execute output problem block outro:

```
1723 \ex@showproblemsout%
1724 \endgroup\fi}
```

### Problems Buffer Interface.

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

```
1725 \newcommand{\writeproblems}[1][\jobname]{%
1726 \ex@close@probfile\ex@start@probfile{#1}}
```

`\closeproblems` Close problems output file (if open):

```
1727 \newcommand{\closeproblems}{\ex@close@probfile}
```

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

```
1728 \newcommand{\readproblems}[1][\jobname]{\ex@close@probfile%
1729 \begingroup%
1730 \ex@config@styletext\ex@config@styletextproblem%
1731 \ex@problemstitle%
1732 \input{#1\ex@config@extproblems}%
1733 \endgroup}
```

`\insertproblems` Show problems buffer:

```
1734 \newcommand{\insertproblems}{\ex@showproblems}
```

## 3.13 Subproblem Environment

### Print Subproblems.

`ex@subproblem` Define options for subproblem environment:

```
1735 \define@key{ex@subproblem}{points}{\ex@scanpoints\ex@subproblem@points#1++@}
1736 \define@key{ex@subproblem}{label}{\def\ex@label{#1}}
1737 \define@key{ex@subproblem}{tag}{\def\subproblemtag{#1}}
```

`printsubproblem` Define printsubproblem environment:

```
1738 \newenvironment{printsubproblem}[1]{%
```

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

```
1739 \par{\ex@config@styletext\addvspace{\ex@config@skipsubproblemabove}}%
1740 \refstepcounter{\ex@subproblemcounter}%
```

Insert hook code:

```
1741 \exf@config@insertsubproblembefore%
```

Start inner group, mark in subproblem:

```
1742 \begingroup%
1743 \def\exf@in@subproblem{}
```

Initialise variables, process arguments:

```
1744 \exf@init@block{\exf@config@skipsubprobleminfo}%
1745 \def\subproblemtag{\getexerciseconfig{tagsubproblem}}%
1746 \let\exf@subproblem@points\undefined%
1747 \let\exf@label\undefined%
1748 \setkeys{exf@subproblem,exf@probleminfo,exf@scansubproblem}{#1}%
```

Process automatic and manual labels:

```
1749 \ifexf@autolabelproblem\label{\exf@config@labelsubproblem%
1750   {\subproblemtag}}\fi%
1751 \ifdefined\exf@label\label{\exf@label}\fi%
1752 \exf@writedata{subproblem}{\subproblemtag}{\problemtag}%
```

Remember subproblem counter for solution:

```
1753 \xdef\exf@prevsubprob{\csname the\exf@subproblemcounter\endcsname}%
1754 \ifcsname theH\exf@subproblemcounter\endcsname%
1755   \xdef\exf@prevsubprobhref{\exf@subproblemcounter.%
1756     \csname theH\exf@subproblemcounter\endcsname}%
1757   \fi%
```

Remember points for display; disable points display if desired:

```
1758 \let\exf@subproblem@points@show\undefined%
1759 \ifdefined\exf@subproblem@points%
1760   \let\exf@subproblem@points@show\exf@subproblem@points\fi%
1761 \exf@ifis\exf@subpointsat{off}{\let\exf@subproblem@points@show\undefined}%
```

Write points to .aux file; add to problem total:

```
1762 \ifdefined\exf@subproblem@points%
1763   \global\let\exf@prevpoints\exf@subproblem@points%
1764   \expandafter\exf@writesubproblem@points\exf@subproblem@points%
1765   \exf@exptwo\exf@addtopoints\exf@problem@points@total\exf@subproblem@points%
```

Warn if no points given for present subproblem but previously:

```
1766 \else\ifdefined\exf@problem@points@total%
1767   \PackageWarning{exframe}{no points defined for \exf@subproblemname}%
1768   \fi\fi%
```

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

```
1769 \exf@ifis\exf@subpointsat{start}{\exf@outpoints{\exf@append@intro}%
1770   {\exf@config@composepointspairstartsubproblem}{\exf@subproblem@points@show}}%
1771 \exf@ifis\exf@subpointsat{start*}{\exf@outpoints{\exf@prepend@intro}%
1772   {\exf@config@composepointspairstartsubproblem}{\exf@subproblem@points@show}}%
```

Insert hook code:

```
1773 \exf@config@insertsubprobleminfo%
```

Write opening line without item:

```

1774 \exf@addmargin\glueexpr\exf@config@skipsubproblemitem\relax%
1775 \ifdim\exf@addmargin=0pt%
1776 \exf@prepend@intro{%
1777 \exf@config@styletitle\exf@config@styletitlesubproblem%
1778 \exf@config@composetitlesubproblem{%
1779 \csname the\exf@subproblemcounter\endcsname}}}%

```

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

```

1780 \else%
1781 \ifdim\exf@addmargin<0pt%
1782 \settowidth\exf@addmargin{%
1783 \exf@config@styletitle\exf@config@styletitlesubproblem%
1784 \exf@config@composeitemsubproblem{\exf@config@countersubproblemmax}%
1785 \exf@config@composeitemsubproblemsep}%
1786 \fi%

```

Define item label:

```

1787 \def\exf@introitem{\makebox[0cm][r]{%
1788 \exf@config@styletitle\exf@config@styletitlesubproblem%
1789 \exf@config@composeitemsubproblem%
1790 {\csname the\exf@subproblemcounter\endcsname}%
1791 \exf@config@composeitemsubproblemsep}}%
1792 \fi%

```

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

```

1793 \exf@ifis\exf@subpointsat{margin}{%
1794 \exf@outpoints{\exf@prepend@def\exf@introitem}%
1795 {\exf@makepointsmarginsubproblem}{\exf@subproblem@points@show}}%

```

Write out opening line:

```

1796 \exf@open@block{\exf@config@skipsubproblemtitle}%

```

Done:

```

1797 \@afterindentfalse}%

```

End environment, show points if desired:

```

1798 {\exf@ifis\exf@subpointsat{end}{\showpoints}%

```

End paragraph and environment:

```

1799 \par\exf@close@block%

```

Display solution if desired:

```

1800 \exf@ifis\exf@solutionbelow{subproblem*}{%
1801 \exf@config@insertsubproblemsolution%
1802 \exf@showsolutions{\exf@config@composetitlesolutionsingle}{}}%

```

End inner group:

```

1803 \endgroup%

```

Insert hook code, vertical skip:

```

1804 \exf@config@insertsubproblemafter%
1805 {\exf@config@styletext\addvspace{\exf@config@skipsubproblembelow}}%

```

Display solution if desired:

```

1806 \ex@ifis\ex@solutionbelow{subproblem}{%
1807 \ex@showssolutions{\ex@config@composetitlesolutionsingle}{}}%

```

Done:

```

1808 \ignorespacesafterend}

```

### Read Subproblem Environment.

`ex@scansubproblem` Define options for scanning subproblem environment:

```

1809 \define@boolkey{ex@scansubproblem}[ex@scansubproblem@]{disable}[true]{}

```

`ex@subproblem@direct` Define direct output version of subproblem environment; pass on to `printsubproblem` environment:

```

1810 \newenvironment{ex@subproblem@direct}[1] []%
1811 {\printsubproblem{#1}}{\endprintsubproblem\ignorespacesafterend}

```

`ex@subproblem@scan` Define scan version of subproblem environment; use `\ex@scanblock` to properly parse optional argument and pass on to `ex@scansubproblem`:

```

1812 \newenvironment{ex@subproblem@scan}%
1813 {\ex@scanblock{\ex@scansubproblem}}{\endex@scansubproblem}%
1814 \newenvironment{ex@scansubproblem}[2]{%

```

Determine subproblem display:

```

1815 \global\ex@scansubproblem@disablefalse%
1816 \setkeys*{ex@scansubproblem}{#1}%
1817 \ex@config@insertsubprobleselect{#1}%
1818 \ifex@scansubproblem@disable\global\ex@scansubproblem@disabletrue\fi%

```

Circumvent subproblem display:

```

1819 \ex@clear@subprobbuf%
1820 \ifex@scansubproblem@disable%
1821 \global\let\ex@prevsubprob\ex@empty%
1822 \def\ex@verbatim@process{\@gobble}%
1823 \else%

```

Write `printsubproblem` environment to buffer:

```

1824 \ex@addline\ex@subprobbuf%
1825 {\@backslashchar begin{printsubproblem}{#1}}%

```

Start verbatim processing:

```

1826 \def\ex@verbatim@process{\ex@append@buf\ex@subprobbuf}%
1827 \fi%
1828 \ex@verbatim#2}%

```

End verbatim processing; close `printsubproblem` environment:

```

1829 {\ex@endverbatim%
1830 \ifex@scansubproblem@disable\else%
1831 \ex@addline\ex@subprobbuf{\@backslashchar end{printsubproblem}}%
1832 \fi%

```

Output buffer immediately:

```

1833 \ex@source@buf\ex@subprobbuf%

```

Done:

```
1834 \ignorespacesafterend}
```

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

```
1835 \newenvironment{\exf@subproblemname}%
1836 {\ifexf@subproblembuf\let\exf@tmp\exf@subproblem@scan%
1837 \else\let\exf@tmp\exf@subproblem@direct\fi%
1838 \exf@tmp}%
1839 {\ifexf@subproblembuf\let\exf@tmp\endexf@subproblem@scan%
1840 \else\let\exf@tmp\endexf@subproblem@direct\fi%
1841 \exf@tmp}
```

## 3.14 Solution Environment

### Print Solutions.

**exf@solution** Define options for solution environment:

```
1842 \define@key{exf@solution}{prob}{\def\exf@solprob{#1}}
1843 \define@key{exf@solution}{subprob}{\def\exf@solsubprob{#1}}
1844 \define@key{exf@solution}{problemtag}{\def\problemtag{#1}}
1845 \define@key{exf@solution}{sheettag}{\def\sheettag{#1}}
1846 \define@key{exf@solution}{href}{\def\exf@solhref{#1}}
1847 \define@key{exf@solution}{label}{\def\exf@label{#1}}
1848 \define@key{exf@solution}{points}{\exf@scanpoints\exf@solution@points#1++@}
1849 \define@key{exf@solution}{probttitle}{\def\exf@solprobttitle{#1}}
1850 \define@key{exf@solution}{firstsol}[]{\let\exf@solfirst\exf@empty}
```

**printsolution** Define printsolution environment to display a previously read solution environment; this works analogously to problem and subproblem:

```
1851 \newenvironment{printsolution}[1]{%
```

Start new paragraph, add vertical space:

```
1852 \par{\exf@config@styletext\addvspace{\exf@config@skipsolutionabove}}%
```

Insert hook code:

```
1853 \exf@config@insertsolutionbefore%
```

Use equation counter for solutions:

```
1854 \ifexf@style@solutionequation%
1855 \exf@eqsav\value{equation}\relax%
1856 \setcounter{equation}{\value{exf@solutionequation}}%
1857 \let\theequation\theexf@solutionequation%
1858 \let\theHequation\theHexf@solutionequation%
1859 \fi%
```

Start a group, initialise variables, process arguments:

```
1860 \begingroup%
1861 \def\exf@in@solution{}%
1862 \def\exf@solprob{}%
1863 \def\exf@solsubprob{}%
1864 \let\exf@solfirst\@undefined%
1865 \let\exf@label\@undefined%
```

```

1866 \let\exf@solution@points\undefined%
1867 \let\exf@solution@points@total\undefined%
1868 \def\exf@solhref{}%
1869 \exf@init@block{\exf@config@skipsolutioninfo}%
1870 \setkeys{exf@solution,exf@probleminfo,exf@scansolution}{#1}%

```

Set solution counter to reflect associated problem:

```

1871 \exf@csdo\def{the\exf@solutioncounter}%
1872   {\exf@config@composeitemsolutionlabel{\exf@solprob}{\exf@solsubprob}}%
1873 \refstepcounter{\exf@solutioncounter}%

```

Set label:

```

1874 \ifdefined\exf@label\label{\exf@label}\fi%

```

Remember points for display; disable points display if desired:

```

1875 \let\exf@solution@points@show\undefined%
1876 \ifdefined\exf@solution@points%
1877   \let\exf@solution@points@show\exf@solution@points\fi%
1878 \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:

```

1879 \exf@ifis\exf@solpointsat{start}{\exf@outpoints{\exf@append@intro}%
1880   {\exf@config@composepointspairstartsolution}{\exf@solution@points@show}}%
1881 \exf@ifis\exf@solpointsat{start*}{\exf@outpoints{\exf@prepend@intro}%
1882   {\exf@config@composepointspairstartsolution}{\exf@solution@points@show}}%

```

Insert hook code, set solution body style:

```

1883 \exf@config@insertsolutioninfo%
1884 \exf@config@styletext\exf@config@styletextsolution%

```

Determine solution for problem or subproblem:

```

1885 \exf@addmargin\glueexpr\ifx\exf@solsubprob\exf@empty%
1886   \exf@config@skipsolutionitem\else\exf@config@skipsolutionitemsub\fi\relax%

```

Write title without item:

```

1887 \ifdim\exf@addmargin=0pt%
1888   \protected@edef\exf@solution@title{%
1889     \exf@composetitle{\exf@solprob}{\exf@solsubprob}}%
1890   \ifx\exf@solsubprob\exf@empty\ifdefined\exf@solfirst\else%
1891     \let\exf@solution@title\exf@empty\fi\fi%
1892   \ifx\exf@solution@title\exf@empty\else%
1893     \exf@prepend@intro{%
1894       \exf@config@styletitle\exf@config@styletitlesolution%
1895       \ifexf@solutionhref\exf@href{\exf@solhref}%
1896       {\exf@solution@title}\else\exf@solution@title\fi}%
1897   \fi%

```

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

```

1898 \else%
1899   \ifdim\exf@addmargin<0pt%
1900     \settowidth\exf@addmargin{%
1901       \exf@config@styletitle\exf@config@styletitlesolution%
1902       \ifx\exf@solsubprob\exf@empty%
1903         \let\exf@tmp\exf@config@composeitemsolutionfirst\else%
1904         \let\exf@tmp\exf@config@composeitemsolutionfirstsub\fi%

```

```

1905   \protected@edef\exf@solution@item{\exf@tmp%
1906     {\exf@config@counterproblemmax}{\exf@config@countersubproblemmax}}%
1907   \ifx\exf@solution@item\exf@empty\else%
1908     \exf@solution@item\exf@config@composeitemsolutionsep\fi}%
1909   \fi%

```

Set item label depending on problem or subproblem:

```

1910   \ifx\exf@solsubprob\exf@empty\ifdefined\exf@solfirst%
1911     \let\exf@tmp\exf@config@composeitemsolutionfirst\else%
1912     \let\exf@tmp\exf@config@composeitemsolution\fi%
1913   \else\ifdefined\exf@solfirst%
1914     \let\exf@tmp\exf@config@composeitemsolutionfirstsub\else%
1915     \let\exf@tmp\exf@config@composeitemsolutionsub\fi\fi%
1916   \protected@edef\exf@solution@item{\exf@tmp{\exf@solprob}{\exf@solsubprob}}%

```

Define item label:

```

1917   \def\exf@introitem{\ifx\exf@solution@item\exf@empty\else\makebox[0cm][r]{%
1918     \exf@config@styletitle\exf@config@styletitlesubproblem%
1919     \ifexf@solutionhref\exf@href{\exf@solhref}{\exf@solution@item}%
1920     \else\exf@solution@item\fi%
1921     \exf@config@composeitemproblemsep}\fi}%
1922   \fi%

```

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

```

1923   \exf@ifis\exf@solpointsat{margin}{%
1924     \exf@outpoints{\exf@prepend@def\exf@introitem}%
1925     {\exf@makepointsmarginsolution}{\exf@solution@points@show}}%

```

Write out opening line:

```

1926   \exf@open@block{\exf@config@skipsolutionontitle}%

```

Done:

```

1927   \@afterindentfalse}%

```

End environment, show points if desired, perform sanity check:

```

1928   {\exf@ifis\exf@solpointsat{end}{\showpoints}%
1929   \exf@warnmismatch{\exf@solutionname}{\exf@solutioncounter}%
1930   {\exf@solution@points@total}{\exf@solution@points}%

```

End paragraph and environment:

```

1931   \par\exf@close@block%

```

Restore original equation counter:

```

1932   \ifexf@style@solutionequation%
1933     \setcounter{exf@solutionequation}{\value{equation}}%
1934     \setcounter{equation}{\exf@eqsav}%
1935   \fi%

```

End inner group:

```

1936   \endgroup%

```

Vertical skip, insert hook code:

```

1937   {\exf@config@styletext\addvspace{\exf@config@skipsolutionbelow}}%
1938   \exf@config@insertsolutionafter%

```

Done:

```
1939 \ignorespacesafterend}
```

`\solutionssection` Define a section for a problem within a solution block:

```
1940 \newcommand{\solutionssection}[1]{\begingroup%
```

Initialise variables, process arguments:

```
1941 \def\exf@solprob{}%
1942 \def\exf@solsubprob{}%
1943 \def\exf@solprobttitle{}%
1944 \let\exf@label\@undefined%
1945 \let\exf@solhref\@undefined%
1946 \setkeys{exf@solution,exf@scansolution}{#1}%
```

Select title (and table of contents entry) corresponding to multiple problems vs. single problem:

```
1947 \let\exf@composesectitle\exf@config@composetitlesolutionsproblemmulti%
1948 \exf@ifis\exf@solutionbelow{problem}{\let\exf@composesectitle%
1949 \exf@config@composetitlesolutionsproblemsingle}%
1950 \exf@ifis\exf@solutionbelow{problem*}{\let\exf@composesectitle%
1951 \exf@config@composetitlesolutionsproblemsingle}%
1952 \def\exf@solutionstoc{\exf@addcontentsline{\exf@config@toclevelsolution}%
1953 {\exf@config@composetocsolution{\exf@solprob}{\exf@solprobttitle}}}%
```

Write section line:

```
1954 \addvspace{\exf@config@skipsolutionsproblemabove}%
1955 \exf@solutionssection{\exf@config@styletitlesolutionsproblem}%
1956 {\exf@composesectitle{\exf@solprob}{\exf@solprobttitle}}%
1957 {\exf@config@skipsolutionsproblemtitle}%
1958 {\exf@solutionstoc}{\exf@label}{\exf@solhref}%
1959 \endgroup}
```

## Read Solution Environment.

`exf@scansolution` Define options for scanning solution environment:

```
1960 \define@key{exf@scansolution}{forproblem}[]{\exf@clear@prevsubprob}
1961 \define@boolkey{exf@scansolution}[exf@scansolution@]{disable}[true]{}
```

`exf@clear@prevsubprob` Clear previous subproblem register:

```
1962 \newcommand{\exf@clear@prevsubprob}{%
1963 \global\let\exf@prevsubprob\@undefined%
1964 \global\let\exf@prevsubprohref\@undefined%
1965 \global\let\exf@prevpoints\@undefined%
1966 \global\exf@scansubproblem@disablefalse}
```

`\exf@process@solsec` If this is the first solution within a problem section, write section heading to buffer:

```
1967 \newcommand{\exf@process@solsec}{%
1968 \def\exf@probarg{\ifdefined\exf@prevprob prob={\exf@prevprob}\fi%
1969 \ifdefined\exf@prevprobttitle,probttitle={\exf@prevprobttitle}\fi%
1970 \ifdefined\exf@prevprohref,href={\exf@prevprohref}\fi%
1971 \ifdefined\exf@sollabel,label={\exf@sollabel}\fi}%
1972 \exf@ifis\exf@solutionbelow{here}{\let\exf@probarg\@undefined}%
```

```

1973 \ex@ifis\ex@solutionbelow{subproblem}{\let\ex@probarg\@undefined}%
1974 \ex@ifis\ex@solutionbelow{subproblem*}{\let\ex@probarg\@undefined}%
1975 \ifdefined\ex@probarg%
1976 \ifex@lineno\ex@addline\ex@solbuf{\ex@linesep}%
1977 \ex@addline\ex@solbuf{\ex@lineno}\fi%
1978 \ex@addline\ex@solbuf{\@backslashchar solutionssection{\ex@probarg}}%
1979 \ex@addline\ex@solbuf{}%
1980 \fi}

```

`@generate@solprobarg` Declare additional arguments to `printsolution` to describe corresponding problem and tags:

```

1981 \newcommand{\ex@generate@solprobarg}{%
1982 \edef\ex@solprobarg{%
1983 \ifdefined\ex@problem@solfirst firstsol,\fi%
1984 \ifdefined\ex@prevprob prob={\ex@prevprob},\fi%
1985 \ifdefined\ex@prevsubprob subprob={\ex@prevsubprob},%
1986 \ifdefined\ex@prevsubprobhref href={\ex@prevsubprobhref},\fi%
1987 \else%
1988 \ifdefined\ex@prevprobhref href={\ex@prevprobhref},\fi%
1989 \fi%
1990 \ifdefined\ex@prevpoints points=%
1991 {\expandafter\ex@formatpoints\ex@prevpoints},\fi%
1992 \ifdefined\sheettag sheettag={\sheettag},\fi%
1993 \ifdefined\problemtag problemtag={\problemtag},\fi}}

```

`ex@solution@direct` Define direct output version of solution environment; pass on to `printsolution` environment:

```

1994 \newenvironment{ex@solution@direct}[1][{}]{%
1995 {\showpoints%
1996 \ex@generate@solprobarg%
1997 \global\let\ex@problem@solfirst\@undefined%
1998 \ex@clear@prevsubprob%
1999 \ex@showsolutionsin%
2000 \let\ex@composetitle\ex@config@composetitlessolutionsingle%
2001 \ex@exptwo\printsolution{\ex@solprobarg#1}}%
2002 {\endprintsolution%
2003 \ex@showsolutionsout%
2004 \ignorespacesafterend}

```

`ex@solution@scan` Define scan version of solution environment; use `\ex@scanblock` to properly parse optional argument and pass on to `ex@scansolution`:

```

2005 \newenvironment{ex@solution@scan}%
2006 {\ex@scanblock{\ex@scansolution}}{\endex@scansolution}%
2007 \newenvironment{ex@scansolution}[2]{%

```

Determine solution association and display:

```

2008 \ex@scansolution@disablefalse%
2009 \setkeys*{ex@scansolution}{#1}%
2010 \ifdefined\ex@prevsubprob%
2011 \ifex@scansubproblem@disable\ex@scansolution@disabletrue\fi%
2012 \else%
2013 \ifex@scanproblem@disable\ex@scansolution@disabletrue\fi%
2014 \fi%

```

Circumvent solution display:

```

2015 \ifexf@scansolution@disable%
2016 \exf@clear@prevsubprob%
2017 \def\exf@verbatim@process{\@gobble}%
2018 \else%

```

If solution is to be displayed immediately, make sure to display points first; insert solution section heading in buffer; compose additional arguments to `printsolution`:

```

2019 \exf@ifis\exf@solutionbelow{here}{\showpoints}%
2020 \exf@generate@solprobarg%
2021 \ifdefined\exf@problem@solfirst\exf@process@solsec\fi%
2022 \global\let\exf@problem@solfirst\@undefined%
2023 \exf@clear@prevsubprob%

```

Write separator and `printsolution` environment to buffer:

```

2024 \ifexf@lineno\exf@addline\exf@solbuf{\exf@linesep}%
2025 \exf@addline\exf@solbuf{\exf@lineno}\fi%
2026 \exf@addline\exf@solbuf%
2027 {\@backslashchar begin{printsolution}{\exf@solprobarg#1}}%

```

Start verbatim processing:

```

2028 \def\exf@verbatim@process{\exf@append@buf\exf@solbuf}%
2029 \fi%
2030 \exf@verbatim#2}%

```

End verbatim processing; close `printsolution` environment:

```

2031 {\exf@endverbatim%
2032 \ifexf@scansolution@disable\else%
2033 \exf@addline\exf@solbuf{\@backslashchar end{printsolution}}%
2034 \global\exf@solbuf@cleanfalse%
2035 \fi%

```

Write buffer to file if output file open:

```

2036 \ifexf@solfile@open%
2037 \exf@write@buf\exf@solfile\exf@solbuf%
2038 \exf@clear@solbuf%
2039 \fi%

```

Drop buffer if solutions not to be displayed:

```

2040 \ifsolutions\else\exf@clear@solbuf\fi%

```

Display solution immediately in various cases:

```

2041 \exf@ifis\exf@solutionbelow{here}{%
2042 \exf@showsolutions{\exf@config@composetitlesolutionsingle}{}}%
2043 \ifdefined\exf@in@subproblem\else%
2044 \exf@ifis\exf@solutionbelow{subproblem}{%
2045 \exf@showsolutions{\exf@config@composetitlesolutionsingle}{}}%
2046 \exf@ifis\exf@solutionbelow{subproblem*}{%
2047 \exf@showsolutions{\exf@config@composetitlesolutionsingle}{}}\fi%
2048 \ifdefined\exf@in@problem\else%
2049 \exf@ifis\exf@solutionbelow{problem}{%
2050 \exf@showsolutions{\exf@config@composetitlesolutionmulti}{}}%
2051 \exf@ifis\exf@solutionbelow{problem*}{%
2052 \exf@showsolutions{\exf@config@composetitlesolutionmulti}{}}\fi%

```

Done:

```

2053 \ignorespacesafterend}

```

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

```
2054 \newenvironment{\exf@solutionname}%
2055  {\ifexf@solutionbuf\let\exf@tmp\exf@solution@scan%
2056   \else\let\exf@tmp\exf@solution@direct\fi%
2057   \exf@tmp}%
2058  {\ifexf@solutionbuf\let\exf@tmp\endexf@solution@scan%
2059   \else\let\exf@tmp\endexf@solution@direct\fi%
2060   \exf@tmp}
```

### 3.15 Solution Blocks

#### Solution Block Handling.

`\exf@solutionssection` Output solution block section:

```
2061 \newcommand{\exf@solutionssection}[6]{%
```

Check whether title is empty:

```
2062   \protected@edef\exf@solutionstitleexp{#2}%
2063   \ifx\exf@solutionstitleexp\exf@empty\else%
```

Define a label:

```
2064   \ifdefined#5%
2065   \exf@csdo\def{the\exf@solutioncounter}%
2066     {\exf@config@composeitemsolutionlabel{\exf@solprob}{\exf@solsubprob}}%
2067   \refstepcounter{\exf@solutioncounter}\label{#5}%
2068   \fi%
```

Output section line:

```
2069   \exf@section{#3}{\exf@config@styletitle\exf@config@styletitlesolution#1%
2070   \ifexf@solutionhref\exf@href{#6}{\exf@solutionstitleexp}%
2071   \else\exf@solutionstitleexp\fi}#4%
2072   \fi}
```

`\exf@solutionstitle` Compose the title for a solution block:

```
2073 \newcommand{\exf@solutionstitle}{\exf@solutionssection%
2074  {\exf@config@styletitlesolutions}{%
2075   \exf@config@composetitlesolutions}{\exf@config@skipsolutionstitle}%
2076  {\exf@addcontentsline{\exf@config@toclevelsolutions}%
2077   {\exf@config@composetocsolutions}}{\@undefined}{\@undefined}}
```

`\exf@showsolutionsin` Output solution block intro:

```
2078 \newcommand{\exf@showsolutionsin}{%
```

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

```
2079   \par\exf@config@styletext\addvspace{\exf@config@skipsolutionsabove}%
2080   \exf@config@styletextsolution%
2081   \exf@config@insertsolutionsbefore}
```

`\exf@showsolutionsout` Output solution block outro:

```
2082 \newcommand{\exf@showsolutionsout}{%
```

Insert hook code; close paragraph; add vertical space:

```
2083 \exf@config@insertsolutionsafter%
2084 \par\exf@config@styletext\advspace{\exf@config@skipsolutionsbelow}}
```

`\exf@showsolutions` Output solution block in buffer:

```
2085 \newcommand{\exf@showsolutions}[2]{%
```

Do nothing if buffer is empty (avoid titles):

```
2086 \ifexf@solbuf@clean\else\begingroup%
```

Execute output solution block intro:

```
2087 \exf@showsolutionsin%
2088 \let\exf@composetitle#1%
2089 #2%
```

Source and clear buffer:

```
2090 \exf@source@buf\exf@solbuf%
2091 \exf@clear@solbuf%
```

Execute output solution block outro:

```
2092 \exf@showsolutionsout%
2093 \endgroup\fi}
```

### Solutions Buffer Interface.

`\writesolutions` Open a file #1.sol for writing solutions; default is present main file name:

```
2094 \newcommand{\writesolutions}[1][\jobname]{%
2095 \exf@close@solfile\exf@start@solfile{#1}}
```

`\closesolutions` Close solutions output file (if open):

```
2096 \newcommand{\closesolutions}{\exf@close@solfile}
```

`\readsolutions` Read solutions from file #1.sol; default is present main file name; switch layout and add heading:

```
2097 \newcommand{\readsolutions}[1][\jobname]{\exf@close@solfile%
2098 \ifsolutions\begingroup%
2099 \exf@config@styletext\exf@config@styletextsolution%
2100 \let\exf@composetitle\exf@config@composetitlesolutionmulti%
2101 \exf@solutionstitle%
2102 \input{#1\exf@config@extsolutions}%
2103 \endgroup\fi}
```

`\insertsolutions` Show solutions buffer:

```
2104 \newcommand{\insertsolutions}{\exf@showsolutions%
2105 {\exf@config@composetitlesolutionmulti}{\exf@solutionstitle}}
```

## 3.16 Interaction with metastr

This package transfers some functionality to the package `metastr` when loaded previously (preferably with package option `course`) by the package option `metastr`. The following changes apply:

- The terms specified by the configuration options `term...`, see section 3.3, are transferred to the corresponding term registers `\metaterm{...}` in `metastr`. This can facilitate internationalisation, and suitable words for English, German, French and Spanish are provided.
- Metadata should be filled via `\metaset` rather than `\exercisedata`. The present package will read it from the `metastr` registers.
- Basic pdf metadata is written automatically or manually. This is done via `\metawritepdfinfo`, consequently, automatic writing of these basic metadata is disabled in `metastr`.
- Sheet-specific metadata (package option `pdfdata=sheet`) is handled via the `metastr` registers `sheetttitle` and `sheetdata` rather than `composemetasheet`.

Apply modifications only when package `metastr` is loaded:

```
2106 \ifdefined\metaset
```

**Transfer Metadata.** Import basic data from `metastr`:

```
2107 \exercisedata{author={\metapick[] {author}}}
2108 \exercisedata{title={\metapick[] {title}}}
2109 \exercisedata{subject={\metapick[] {subject}}}
2110 \exercisedata{keyword={\metapick[] {keyword}}}
2111 \exercisedata{date={\metapick[] {date}}}
```

Import course data from `metastr`:

```
2112 \ifdefined\mstr@def@course
2113 \exercisedata{course={\metapick[] {course}}}
2114 \exercisedata{instructor={\metapick[] {instructor}}}
2115 \exercisedata{institution={\metapick[] {institution}}}
2116 \exercisedata{period={\metapick[] {period}}}
2117 \exercisedata{material={\metapick[] {material}}}
2118 \fi
```

**Sheet Data.**

`sheetttitle` Registers for sheet title and author:

`sheetauthor`

```
2119 \metadef{sheetttitle}
2120 \metadef{sheetauthor}
```

`\exf@writemetadata` Write pdf metadata via `metastr` package, let `exframe` initiate process (especially for sheet):

```
2121 \metaunset[info]{writepdf}
2122 \def\exf@writemetadata{%
2123   \exf@ifis\exf@metadata{sheet}{\metaset[use]{sheetttitle}{}}%
2124   \metawritepdfinfo%
2125   \exf@ifis\exf@metadata{sheet}{\metaunset[use]{sheetttitle}}}
```

Fill registers:

```
2126 \metaset{sheetauthor}{\exerciseifempty{\getsheetdata{author}}%
2127   {\metapick[#1]{instructor}}{\getsheetdata{author}}}
2128 \metaset{sheetttitle}{\exerciseifempty{\getsheetdata{rawtitle}}%
2129   {\metatranslate[#1]{sheet} \thesheet}%
2130   {\getsheetdata{rawtitle}}}
2131 \metaset{author}{\exerciseifempty{\getsheetdata{author}}%
```

```

2132 {\metapick[#1]{instructor}}{\metapick[#1]{sheetauthor}}
2133 \metaset{subtitle}{%
2134 \metaif[use]{sheetttitle}%
2135 {\metapick[#1]{sheetttitle}}%
2136 {\metapick[#1]{material}}}

```

### Terms.

term... Transfer term definitions to metastr:

```

2137 \exerciseconfig{termsheet}{\metaterm{sheet}}
2138 \exerciseconfig{termsheets}{\metaterm{sheets}}
2139 \exerciseconfig{termproblem}{\metaterm{problem}}
2140 \exerciseconfig{termproblems}{\metaterm{problems}}
2141 \exerciseconfig{termsolution}{\metaterm{solution}}
2142 \exerciseconfig{termsolutions}{\metaterm{solutions}}
2143 \exerciseconfig{termpoint}{\metaterm{point}}
2144 \exerciseconfig{termpoints}{\metaterm{points}}

```

### Translations. English:

```

2145 \ifdefined\mstr@lang@en
2146 \metasetterm[en]{sheet}{Sheet}
2147 \metasetterm[en]{sheets}{Sheets}
2148 \metasetterm[en]{problem}{Problem}
2149 \metasetterm[en]{problems}{Problems}
2150 \metasetterm[en]{solution}{Solution}
2151 \metasetterm[en]{solutions}{Solutions}
2152 \metasetterm[en]{point}{point}
2153 \metasetterm[en]{points}{points}
2154 \fi

```

### German:

```

2155 \ifdefined\mstr@lang@de
2156 \metasetterm[de]{sheet}{Blatt}
2157 \metasetterm[de]{sheets}{Blätter}
2158 \metasetterm[de]{problem}{Aufgabe}
2159 \metasetterm[de]{problems}{Aufgaben}
2160 \metasetterm[de]{solution}{L"osung}
2161 \metasetterm[de]{solutions}{L"osungen}
2162 \metasetterm[de]{point}{Punkt}
2163 \metasetterm[de]{points}{Punkte}
2164 \fi

```

### French:

```

2165 \ifdefined\mstr@lang@fr
2166 \metasetterm[fr]{sheet}{Feuille}
2167 \metasetterm[fr]{sheets}{Feuilles}
2168 \metasetterm[fr]{problem}{Probl\‘eme}
2169 \metasetterm[fr]{problems}{Probl\‘emes}
2170 \metasetterm[fr]{solution}{Solution}
2171 \metasetterm[fr]{solutions}{Solutions}
2172 \metasetterm[fr]{point}{point}
2173 \metasetterm[fr]{points}{points}
2174 \fi

```

### Spanish:

```
2175 \ifdefined\mstr@lang@es
2176 \metasetterm[es]{sheet}{Hoja}
2177 \metasetterm[es]{sheets}{Hojas}
2178 \metasetterm[es]{problem}{Problema}
2179 \metasetterm[es]{problems}{Problemas}
2180 \metasetterm[es]{solution}{Solucion}
2181 \metasetterm[es]{solutions}{Soluciones}
2182 \metasetterm[es]{point}{punto}
2183 \metasetterm[es]{points}{puntos}
2184 \fi
```

Close metastr conditional:

```
2185 \fi
```