randintlist

Creating random integer number lists,
with multiple numbers or not,
sorted or not.

Version 0.1.2 – 24/10/2024

Cédric Pierquet
c pierquet – at – outlook . fr
https://github.com/cpierquet/randintlist

---

10 numbers, between 1 and 100, without repetition:
15,53,71,33,44,30,27,12,36,34
The 5th value is:
44

10 numbers, between 1 and 100, without multiples of 5:
47,8,58,79,81,3,27,21,42,9
The 9th value is:
42

15 numbers, between 1 and 20, with repetition:
12,3,2,5,5,15,11,9,13,10,20,14,16,14,10
The last value is:
10

6 sorted numbers, between 1 and 51, without repetition:
ascending : 3,5,24,43,50,51
descending : 39>27>21>16>12>5

The luarandom package do the same things, but with the obligation to compile with LuaLaTeX.

Contents

1 Loading, useful packages

In order to load `randintlist`, simply use:

```
\usepackage{randintlist}
```

Loaded packages are `ifthen`, `simplekv`, `listofitems`, `randomlist`, `xintexpr` and `xstring`.

2 The Macros

2.1 Global usage

Package `randintlist` supports the creation of random integer number lists where a number will appear only once or multiple times. Generated lists can te used with `listofitems`.

All engines TEX are compatible with this package.

2.2 Generate the list

```
%generate list
\randintlist[keys]{\macro}
```

Available keys are:

- `min`: minimum value (default `1`);
- `max`: maximum value (default `50`);
- `nb`: number of values (default `6`);
- `sep`: separator for the list (default `,`);
- `sort`: sorting options, within `no/asc/dec` (default `no`);
- `repeat`: boolean to authorize repeating values (default `false`);
- `exclude`: list of excluded values (default `empty`);
- `seed`: random seed value according to used packages (default `-`).

```
%default values
\randintlist{\mylistA}\mylistA
```

4,36,12,6,18,5

```
%10 between 1 and 50, with ascending
\randintlist[sort=asc,min=1,max=50,nb=10]{\mylistB}\mylistB
```

7,14,18,21,22,23,36,38,39,46

```
%15 between 1 and 50, with ascending and repetitions allowed
\randintlist[sort=asc,min=1,max=50,nb=15,repeat]{\mylistC}\mylistC
```

3,7,9,16,18,22,25,27,32,33,35,37,37,45,46

```
%15 between 1 and 50, without multiples of 5
\randintlist[%
    sort=asc,min=1,max=50,nb=15,repeat,%
    exclude={5,10,15,20,25,30,35,40,45,50}]%
    {\mylistC}\mylistC
```

2,3,6,6,7,8,9,12,17,23,34,36,47,48,48

```
%list used with listofitems
\randintlist{\mylistD}\mylistD\par
\readlist*\mylistused{\mylistD}\showitems{\mylistused}\par
\mylistused[1]; \mylistused[-1]
```

35,42,45,36,12,5
35 42 45 36 12 5
35; 5

2.3 Accessing elements

```
%accessing item
\getitemfromrandintlist[separator]{\macro}{index}
```

```
%with default keys
\randintlist{\mylistE}raw list: \mylistE\par
items list:\par
\xintFor* #1 in {\xintSeq{1}{6}}\do{\getitemfromrandintlist{\mylistE}{#1}\par}
first element: \getitemfromrandintlist{\mylistE}{1}
```

raw list: 39,37,10,21,35,12
items list:
39
37
10
21
35
12
first element: 39

2.4 Version française

Voilà les commandes en version française, la syntaxe et les clés ne seront pas explicitées.

```
%obtenir la liste
\ListeRandint[Min=..,Max=..,Nb=..,Repet=..,Graine=..,Tri=..,Sep=..,Exclure=..]{\macro}

%extraire un élément
\ExtraireEltListeRandint[sep]{\macro}{position}
```

```
%liste
\ListeRandint[Min=5,Max=15,Nb=7,Repet,Tri=croiss,Sep={/}]{\maliste}\maliste\\
%élément
\ExtraireEltListeRandint[/]{\maliste}{4}
```

5/5/6/6/9/11/14
6

3 Example

The following example uses Ti*k*Z, and comes from `luarandom`'s documentation.

```
\begin{tikzpicture}[scale=0.75]
    \randintlist[min=1,max=100,nb=100]{\mylistsquare}
    \draw[thin,gray] (0,0) grid (10,10) ;
    \foreach \i in {1,...,100}{%
        \xdef\tmpnumber{\getitemfromrandintlist{\mylistsquare}{\i}}%
        \xdef\tmpnumberrow{\xinteval{\xintiiRem{\i-1}{10}}}%
        \xdef\tmpnumbercol{\xinteval{\xintiiQuo{\i-1}{10}}}%
        \draw ({0.5+\tmpnumbercol},{0.5+\tmpnumberrow}) node {\tmpnumber} ;
    }%
\end{tikzpicture}
```

55	54	80	11	5	63	19	23	97	21
51	32	41	49	75	17	20	67	62	57
10	53	61	8	90	15	29	96	47	25
43	85	89	3	14	68	31	98	4	77
2	33	24	81	78	95	6	59	87	82
88	65	50	35	69	7	37	1	99	13
60	9	70	58	93	79	27	52	45	76
42	44	74	34	56	66	36	86	22	73
94	46	12	100	92	26	18	72	91	16
71	64	39	30	84	38	48	83	28	40

4 History

0.1.2: Changing name of internal macro
0.1.1: Possibility to exclude values
0.1.0: Initial version

5 The code

```
% Author     : C. Pierquet
% licence    : Released under the LaTeX Project Public License v1.3c or later, see
     http://www.latex-project.org/lppl.txtf

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{randintlist}[2024/10/24 0.1.2 Create a list of random numbers with or without multiple values]

%------History
% 0.1.2 Changing name of macro
% 0.1.1 Possibility to exclude values
% 0.1.0 Initial version

%------Packages
\RequirePackage{simplekv}
\RequirePackage{listofitems}
\RequirePackage{randomlist}
\RequirePackage{xintexpr}
\RequirePackage{xstring}
\RequirePackage{ifthen}

%-----Macros (latex3) for sorting and seed
\ExplSyntaxOn
\cs_new_eq:NN \randintseed \sys_gset_rand_seed:n
\NewDocumentCommand\intascsortlist{m}
{
    \clist_sort:Nn #1
    {
        \fp_compare:nNnTF {##1} > {##2}
        { \sort_return_swapped: }
        { \sort_return_same: }
    }
}
\NewDocumentCommand\intdessortlist{m}
{
    \clist_sort:Nn #1
    {
        \fp_compare:nNnTF {##1} < {##2}
        { \sort_return_swapped: }
        { \sort_return_same: }
    }
}
\ExplSyntaxOff

%----Internal macro (latex2) for testing if element is in list
\newcommand\ifintvalueinlist[2]{\IfSubStr{,#2,}{,#1,}}

\newcommand\boolvalueinlist[2]{\IfSubStr{,#2,}{,#1,}{\def\resisinlist{1}}{\def\resisinlist{0}}}

\newcommand\testifintvalueinlist[4]{%
    \IfSubStr{,#2,}{,#1,}{\xdef\RESTMPVALUE{1}}{\xdef\RESTMPVALUE{0}}%
    \xintifboolexpr{ \RESTMPVALUE == 1}{#3}{#4}%
}

%----Macro for generating
\defKV[randomlistintegers]{%
    min=\def\TAEEmin{#1},%
    max=\def\TAEEmax{#1},%
    nb=\def\TAEEnb{#1},%
    sep=\def\TAEEsep{#1},%
    sort=\def\TAEEtri{#1},%
```

```
    seed=\def\TAEEseed{#1},%
    exclude=\def\TAEEexcluded{#1}
}

\setKVdefault[randomlistintegers]{%
    min=1,%
    max=50,%
    nb=6,%
    sep={,},%
    sort=no,%
    repeat=false,%
    seed={-},%
    exclude={}
}

\NewList{tmprandintlist}

\NewDocumentCommand\randintlist{ O{} m }{%1=keys,2=listname
    \useKVdefault[randomlistintegers]%
    \setKV[randomlistintegers]{#1}%
    \ifboolKV[randomlistintegers]{repeat}%repeat or not
        {%repeat allowed
            \IfStrEq{\TAEEseed}{-}%
                {}%
                {%
                    \randintseed{\TAEEseed}%
                }%
            %list creation of first element
            \def\resisinlist{1}%
            \whiledo{\resisinlist=1}{%
                \xdef\tmpresrandint{\fpeval{randint(\TAEEmin,\TAEEmax)}}%
                \boolvalueinlist{\tmpresrandint}{\TAEEexcluded}%
            }%
            \xdef#2{\tmpresrandint}%
            %list creation of other elements
            \xintFor* ##1 in {\xintSeq{2}{\TAEEnb}}%
                \do{%
                    \def\resisinlist{1}%
                    \whiledo{\resisinlist=1}{%
                        \xdef\tmpresrandint{\fpeval{randint(\TAEEmin,\TAEEmax)}}%
                        \boolvalueinlist{\tmpresrandint}{\TAEEexcluded}%
                    }%
                \xdef#2{#2,\tmpresrandint}%
                }%
        }%
        {%no repeating
            %randomize numbers
            \IfStrEq{\TAEEseed}{-}%
                {}%
                {%
                    \RLsetrandomseed{\TAEEseed}%
                }%
            \ClearList{tmprandintlist}%clearing the list
            \xintFor* ##1 in {\xintSeq{\TAEEmin}{\TAEEmax}}%
                \do{%
                    \ifintvalueinlist{##1}{\TAEEexcluded}%
                        {}%
                        {%
                            \InsertRandomItem{tmprandintlist}{##1}%
                        }%
                }%
            %list creation (first then other)
            \xdef#2{\tmprandintlist[0]}%
            \xintFor* ##1 in {\xintSeq{1}{\TAEEnb-1}}%
                \do{%
                    \xdef#2{#2,\tmprandintlist[##1]}%
                }%
        }%
    %sorting
    \IfStrEq{\TAEEtri}{asc}%if ascending
        {\intascsortlist{#2}}%
        {}%
```

```
    \IfStrEq{\TAEEtri}{des}%if descending
        {\intdessortlist{#2}}%
        {}%
    \StrSubstitute{#2}{,}{\TAEEsep}[#2]%swipping separator if necessary
}

%-----Macro for extracting
\NewDocumentCommand\getitemfromrandintlist{ O{,} m m }{%
    \IfEq{#1}{/}%
        {%
            \setsepchar[.]{#1}%
        }%
        {%
            \setsepchar{#1}%
        }%
    \readlist*\TMPLISTRANDINT{#2}%
    \TMPLISTRANDINT[#3]%
```