randintlist

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

Version 0.1.3 – 04/01/2025

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

---

10 numbers, between 1 and 100, without repetition:
3,53,7,75,46,24,96,62,26,59
The 5th value is:
46

10 numbers, between 1 and 100, without multiples of 5:
48,56,6,74,84,43,89,86,53,93
The 9th value is:
53

15 numbers, between 1 and 20, with repetition:
17,7,20,6,5,6,5,10,5,15,18,17,4,6,11
The last value is:
11

6 sorted numbers, between 1 and 51, without repetition:
ascending : 16,18,31,45,48,50
descending : 50>48>47>44>35>33

---

1. *The luarandom package do the same things, but with the obligation to compile with LuaLaTeX.*
2. *The tuple package is so much better... but I keep* `randintlist`*, without new features...*

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
```

30,17,46,11,50,37

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

1,2,7,15,17,21,24,25,33,39

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

3,4,7,15,17,19,20,23,30,32,36,41,42,45,45

```
%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
```

1,4,7,9,11,14,14,18,19,22,23,24,31,47,48

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

11,48,4,32,26,45
11 48 4 32 26 45
11; 45

2.3 Accessing elements

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

```
%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}\par
```

raw list: 2,29,4,27,30,16
items list:
2
29
4
27
30
16
first element: 2

```
\getitemfromrandintlist{\mylistE}{3}[\myres]%
third element: \myres
```

third element: 4

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}[\macrores]
```

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

6/9/10/11/11/12/15
11

```
%liste
\ListeRandint[Min=50,Max=100,Nb=10,Repet,Tri=croiss]{\malisteB}\malisteB\\
%troisième élément
\ExtraireEltListeRandint{\malisteB}{3}[\montroisieme]%
troisième élément : \montroisieme
```

53,56,58,61,63,80,80,83,84,91
troisième élément : 58

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}
```

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

4 History

0.1.3: Bugfix
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.txt

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{randintlist}[2025/01/04 0.1.3 Create a list of random numbers with or without multiple values]

%------History
% 0.1.3 new usage of extracting element (with storing \macro)
% 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 o }{%
    \IfStrEq{#1}{/}%
        {%
            \setsepchar[.]{#1}%
        }%
        {%
            \setsepchar{#1}%
        }%
    \readlist*\TMPLISTRANDINT{#2}%
    \IfNoValueTF{#4}{\TMPLISTRANDINT[#3]}{\itemtomacro\TMPLISTRANDINT[#3]#4}%
}

%-----french version
\defKV[randomlisteentiers]{%
    Min=\def\TAEEmin{#1},%
    Max=\def\TAEEmax{#1},%
    Nb=\def\TAEEnb{#1},%
    Sep=\def\TAEEsep{#1},%
    Tri=\def\TAEEtri{#1},%
    Graine=\def\TAEEseed{#1},%
    Exclure=\def\TAEEexcluded{#1}
}

\setKVdefault[randomlisteentiers]{%
    Min=1,%
    Max=50,%
    Nb=6,%
    Sep={,},%
    Tri=non,%
    Repet=false,%
    Graine={-},%
    Exclure={}
}

\NewDocumentCommand\ListeRandint{ O{} m }{%1=keys,2=listname
    \useKVdefault[randomlisteentiers]%
    \setKV[randomlisteentiers]{#1}%
    \ifboolKV[randomlisteentiers]{Repet}%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}{croiss}%if ascending
        {\intascsortlist{#2}}%
        {}%
    \IfStrEq{\TAEEtri}{decroiss}%if descending
        {\intdessortlist{#2}}%
        {}%
    \StrSubstitute{#2}{,}{\TAEEsep}[#2]%swipping separator if necessary
}

%-----Macro for extracting
\NewDocumentCommand\ExtraireEltListeRandint{ O{,} m m o }{%
    \IfStrEq{#1}{/}%
        {%
            \setsepchar[.]{#1}%
        }%
        {%
            \setsepchar{#1}%
        }%
    \readlist*\TMPLISTRANDINT{#2}%
    \IfNoValueTF{#4}{\TMPLISTRANDINT[#3]}{\itemtomacro\TMPLISTRANDINT[#3]#4}%
}
```