randintlist-l3

LaTeX3

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

Version 0.1.6 – 17/10/2025

Cédric Pierquet
c pierquet – at – outlook . fr
https://github.com/cpierquet/latex-packages/tree/main/randintlist

---

10 numbers, between 1 and 100, without repetition:
9,71,88,40,26,7,60,14,56,55
The 5th value is:
26

10 numbers, between 1 and 100, without multiples of 5:
69,69,2,12,6,71,48,18,69,66
The 9th value is:
69

15 numbers, between 1 and 20, with repetition:
8,3,19,10,17,20,4,1,5,11,7,13,18,7,2
The last value is:
2

6 sorted numbers, between 1 and 51, without repetition:
ascending: 3,17,22,35,38,49
descending: 51 > 40 > 24 > 22 > 15 > 14

---

1. *The luarandom package do the same things, but with the obligation to compile with LuaLaTeX.*
2. *The tuple package is so much better... take a look...*

Contents

1 Loading, useful packages

In order to load `randintlist-l3`, simply use:

```
\usepackage{randintlist-l3}
```

All code is written in LaTeX3, so no extra packages are needed.

2 The Macros

2.1 Global usage

Package `randintlist-l3` supports the creation of random integer number lists where a number will appear only once or multiple times. Generated lists can be used with `listofitems`.

Macros are prefixed with `\rdl...` (for randintlist).

2.2 Generate the list

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

Available keys are:

- `min`: minimum value (default `1`);
- `max`: maximum value (default `100`);
- `nb`: number of values (default `6`);
- `sort`: sorting options, within `no/asc/dec` (default `no`);
- `repeat`: boolean to authorize repeating values (default `false`);
- `excluded`: list of excluded values (default `empty`);
- `seed`: random seed value according to used packages (default `-`).

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

52,88,35,78,58,53,70,34,100,42

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

9,13,20,21,30,30,36,37,40,43

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

2,6,6,13,14,15,16,16,18,20,20,22,43,44,46

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

7,13,14,16,19,21,21,24,27,28,34,36,44,46,49

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

47,71,28,56,66,33,9,73,17,83
47 71 28 56 66 33 9 73 17 83
47; 83

```
%list, with alt sep, used with listofitems
\rdlgenlist<+>{\mylistZ}\mylistZ\par
\setsepchar{+}\readlist*\mylistused{\mylistZ}\showitems{\mylistused}\par
\mylistused[1]; \mylistused[-1]
```

53+21+80+36+94+26+19+2+43+2
53 21 80 36 94 26 19 2 43 2
53; 2

2.3 Accessing elements

```
%accessing item
\rdlgetitem(*){\macro}{index}[\macrores]
```

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

raw list: 27,39,27,68,89,81,61,21,83,94
items list:
27
39
27
68
89
81
first element: 27

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

third element: 27

3 History

0.1.6: Improvements with LaTeX3
0.1.5: Initial version

4 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}
\ProvidesExplPackage{randintlist-l3}{2025-10-17}{0.1.6}{Create a list of random numbers with or without multiple
     values}

%------History
% 0.1.6 Improvements with latex3
% 0.1.5 Initial version

%------Main macro
%variables
\clist_new:N \l_randintlist_clist
\clist_new:N \l_randintlist_excluded_clist
\clist_new:N \l_randintlist_result_clist
\int_new:N \l_randintlist_min_int
\int_new:N \l_randintlist_max_int
\int_new:N \l_randintlist_nb_int
\int_new:N \l_randintlist_random_number
\int_new:N \l_randintlist_seed_int
\bool_new:N \l_randintlist_repeat_bool
\bool_new:N \l_randintlist_keepvalue_bool
%\tl_new:N \l_randintlist_sep_tl
\tl_new:N \l_randintlist_sort_tl
\str_new:N \l_randintlist_sort_str

%keys
\keys_define:nn { randomlistintegers }
{
    min      .int_set:N    = \l_randintlist_min_int,
    max      .int_set:N    = \l_randintlist_max_int,
    nb       .int_set:N    = \l_randintlist_nb_int,
    seed     .int_set:N    = \l_randintlist_seed_int,
    excluded .clist_set:N  = \l_randintlist_excluded_clist,
    repeat   .bool_set:N   = \l_randintlist_repeat_bool,
    %sep      .tl_set:N     = \l_randintlist_sep_tl,
    sort     .tl_set:N     = \l_randintlist_sort_tl,
    min      .initial:n    = 1,
    max      .initial:n    = 100,
    nb       .initial:n    = 10,
    seed     .initial:n    = -1,
    excluded .initial:n    = {},
    repeat   .initial:n    = true,
    %sep      .initial:n    = {,},
    sort     .initial:n    = {none},
    min      .default:n    = 1,
    max      .default:n    = 100,
    nb       .default:n    = 10,
    seed     .default:n    = -1,
    excluded .default:n    = {},
    repeat   .default:n    = true,
    %sep      .default:n    = {,},
    sort     .default:n    = {none},
}

%macro
\NewDocumentCommand\rdlgenlist { O{} D<>{,} m } %#1=keys / #2=sep / #3=macrolist
{
    %\group_begin:
    % key init
```

```
    \keys_set:nn { randomlistintegers } { #1 }

    % seed if necessary
    \int_compare:nNnT { \l_randintlist_seed_int } > { -1 }
    {
        \sys_rand_seed:n { \l_randintlist_seed_int }
    }

    % list init
    \clist_clear:N \l_randintlist_result_clist

    % repeating or not
    \bool_if:NTF \l_randintlist_repeat_bool
    {% w repeating
        \int_step_inline:nnnn { 1 } { 1 } { \l_randintlist_nb_int }
        {
            \bool_set_false:N \l_randintlist_keepvalue_bool

            \bool_until_do:Nn \l_randintlist_keepvalue_bool
            {
                \int_set:Nn \l_randintlist_random_number
                {
                    \fp_eval:n { randint( \l_randintlist_min_int , \l_randintlist_max_int ) }
                }
                \tl_set:Ne \l_tmpa_tl { \int_use:N \l_randintlist_random_number }

                \clist_if_in:NeTF \l_randintlist_excluded_clist { \l_tmpa_tl }
                {
                    \bool_set_false:N \l_randintlist_keepvalue_bool
                }
                {
                    \bool_set_true:N \l_randintlist_keepvalue_bool
                    \clist_put_right:Ne \l_randintlist_result_clist { \int_use:N \l_randintlist_random_number }
                }
            }
        }
    }
    {%w/o repetitions
        \seq_clear_new:N \l_tmpa_seq

        \int_step_inline:nnnn { \l_randintlist_min_int } { 1 } { \l_randintlist_max_int }
        {
            %creation of [min,...,max] list

            \clist_if_in:NnTF \l_randintlist_excluded_clist { ##1 }
            {
                \bool_set_false:N \l_randintlist_keepvalue_bool
            }
            {
                \bool_set_true:N \l_randintlist_keepvalue_bool
                \seq_put_right:Nn \l_tmpa_seq { ##1 }
            }
        }

        %shuffle
        \seq_shuffle:N \l_tmpa_seq

        %truncate
        \clist_clear_new:N \l_randintlist_result_clist
        \int_step_inline:nnnn { 1 } { 1 } { \l_randintlist_nb_int }
        {
            \seq_pop_left:NN \l_tmpa_seq \l_tmpa_tl
            \clist_put_right:Ne \l_randintlist_result_clist { \l_tmpa_tl }
        }
    }

    %sort or not
    \str_set:Ne \l_randintlist_sort_str { \tl_use:N \l_randintlist_sort_tl }
    \str_case:Nn { \l_randintlist_sort_str }
    {
        {asc}  { \clist_sort:Nn \l_randintlist_result_clist { \int_compare:nNnTF { ##1 } > { ##2 } {
     \sort_return_swapped: } { \sort_return_same: } } }
```

```
        {des}  { \clist_sort:Nn \l_randintlist_result_clist { \int_compare:nNnTF { ##1 } < { ##2 } {
     \sort_return_swapped: } { \sort_return_same: } } }
    }

    %storing
    \tl_gset:Ne #3 {
        \clist_use:Nn \l_randintlist_result_clist { #2 }
    }
    %\group_end:
}


\NewDocumentCommand\rdlgetitem{ s m m O{\resmyelt} }{%
    \clist_set:Ne \l_randintlist_clist {#2}
    \IfBooleanTF{#1}
    {
        \clist_item:Nn \l_randintlist_clist { #3 }
    }
    {
        \tl_gset:Ne #4{ \clist_item:Nn \l_randintlist_clist { #3 } }
    }
}

\endinput
```