

commalists-tools

Macros for manipulating numeral
comma separated lists:
sorting, adding, removing, etc

Version 0.1.0 - 25/09/2024

Cédric Pierquet

c pierquet - at - outlook . fr

<https://github.com/cpierquet/commalists-tools>

```
\def\mytestlist{14,10,15,11,9,10}  
\minoflist*\mytestlist  
\maxoflist*{14,10,15,11,9,10}  
\sortasclist*\mytestlist  
\meanoflist*\mytestlist  
\removevalinlist*{10}{\mytestlist}  
\xintifvalinlist{15}{\mytestlist}{true}{false}
```

We consider the list: 14,10,15,11,9,10

The minimum value is 9 and the maximum is 15

Ascending sorted list is 9,10,10,11,14,15

Meaning value of the list is 11.5

If we remove the value 10, then the list is 14,15,11,9

We test if 15 is in the list : true

Contents

1 Loading, useful packages **3**

2 The Macros **3**

2.1 Global usage 3

2.2 The macros for calculing 3

2.3 Macros for manipulating 4

2.4 Macros with testing 6

3 History **7**

4 The code **7**

1 Loading, useful packages

In order to load `randintlist`, simply use:

```
\usepackage{commalists-tools}
```

Loaded packages are `listofitems`, `xintexpr` and `xstring`.

The idea is to propose "reusage" of lists with `listofitems` or other commands, for example.

2 The Macros

2.1 Global usage

Package `commalists-tools` supports (basic) manipulations on numeral comma separated lists:

- sorting;
- adding values;
- removing values;
- counting values;
- mean, sum, product;
- etc.

Starred versions only print the result, whereas non starred versions store the result into a macro.

All engines \TeX are compatible with this package.

2.2 The macros for calculing

```
%getting min of list, only printing
\minoflist*{list}
%storing min of list into \macro (\resmin by default)
\minoflist{list}[\macro]
%getting max of list, only printing
\maxoflist*{list}
%storing max of list into \macro (\resmax by default)
\maxoflist{list}[\macro]
```

```
%only printing
\minoflist*{14,10,15,11,9,10} and \maxoflist*{14,10,15,11,9,10} \\
%storing min/max of list
\def\mytestlist{10,14.5,20,12,8.5}
\minoflist{\mytestlist}[\mymin]Min of list is \mymin\ \&
\maxoflist{\mytestlist}[\mymax]Max of list is \mymax

9 and 15
Min of list is 8.5 & Max of list is 20
```

```
%getting mean of list, only printing
\meanoflist*{list}
%storing mean of list into \macro (\resmean by default)
\meanoflist{list}[\macro]
%getting sum of list, only printing
\sumoflist*{list}
%storing max of list into \macro (\ressum by default)
\sumoflist{list}[\macro]
%getting prod of list, only printing
\prodoflist*{list}
%storing max of list into \macro (\resprod by default)
\prodoflist{list}[\macro]
```

```
%only printing
\meanoflist*{14,10,15,11,9,10} \\
%storing
\meanoflist{14,10,15,11,9,10}[\mymean]\mymean \\
%only printing
\sumoflist*{14,10,15,11,9,10} and \prodoflist*{14,10,15,11,9,10} \\
%storing
\sumoflist{14,10,15,11,9,10}[\mysum]\prodoflist{14,10,15,11,9,10}[\myprod]
The sum is \mysum\ and the prod is \myprod
```

```
11.5
11.5
69 and 2079000
The sum is 69 and the prod is 2079000
```

2.3 Macros for manipulating

```
%sorting (asc), only printing
\sortasclist*{list}
%sorting (asc) and storing (overwrite)
\sortasclist{list}
%sorting (des), only printing
\sortdeslist*{list}
%sorting (des) and storing (overwrite)
\sortdeslist{list}
```

```
%sorting (asc), only printing
\sortasclist*{14,10,15,11,9,10}\\
%sorting (asc) and storing (overwrite)
\def\tmpsortlist{14,10,15,11,9,10}
\sortasclist{\tmpsortlist}\tmpsortlist\\
%analysing
\readlist*\tmpSORTlist{\tmpsortlist}
\showitems{\tmpSORTlist}
```

```
9,10,10,11,14,15
9,10,10,11,14,15


|   |    |    |    |    |    |
|---|----|----|----|----|----|
| 9 | 10 | 10 | 11 | 14 | 15 |
|---|----|----|----|----|----|


```

```
%sorting (des), only printing
\sortdeslist*{14,10,15,11,9,10}\\
%sorting (asc) and storing (overwrite)
\def\tmpsortlist{14,10,15,11,9,10}
\sortdeslist{\tmpsortlist}\tmpsortlist\\
%analysing
\readlist*\tmpSORTlist{\tmpsortlist}
\showitems{\tmpSORTlist}
```

15,14,11,10,10,9
 15,14,11,10,10,9

15	14	11	10	10	9
----	----	----	----	----	---

```
%adding, only printing
\addvalinlist*{values}{list}
%adding and storing (overwrite)
\addvalinlist{values}{list}
%removing, only printing
\removevalinlist*{value}{list}
%removing and storing (overwrite)
\removevalinlist{value}{list}
```

```
%only printing
\addvalinlist*{3}{1,2,5,6}\\
%defining and adding
\def\tmpaddlist{1,2,4,5,6}
\addvalinlist{3}{\tmpaddlist}\tmpaddlist\\
%analysing
\readlist*\tmpADDlist{\tmpaddlist}
\showitems{\tmpADDlist}
```

1,2,5,6,3
 1,2,4,5,6,3

1	2	4	5	6	3
---	---	---	---	---	---

```
%only printing
\removevalinlist*{3}{1,2,3,6,3,1,5,7,3}\\
%defining and removing
\def\tmpprelist{1,2,3,6,3,1,5,7,3}
\removevalinlist{3}{\tmpprelist}\tmpprelist\\
%analysing
\readlist*\tmpRElist{\tmpprelist}
\showitems{\tmpRElist}
```

1,2,6,1,5,7
 1,2,6,1,5,7

1	2	6	1	5	7
---	---	---	---	---	---

```
%reversing, only printing
\reverselist*{list}
%reversing and storing (overwrite)
\reverselist{list}
```

```
%only printing
\reverselist*{14,10,15,11,9,10}\\
%reversing and storing
%storing
\reverselist{14,10,15,11,9,10}[\myreverse]\myreverse\\
%analysing
\readlist*\tmpREVERSElist{\myreverse}
\showitems{\tmpREVERSElist}
```

```
10,9,11,15,10,14
10,9,11,15,10,14


|    |   |    |    |    |    |
|----|---|----|----|----|----|
| 10 | 9 | 11 | 15 | 10 | 14 |
|----|---|----|----|----|----|


```

2.4 Macros with testing

```
%testing if value is in list, with boolean result in \macro (\resisinlist by default)
\boolvalinlist{value}{list}[\macro]
%conditionnal test if value is in list, according to xint syntax
\xintifvalinlist{3}{0,1,2,3}{true}{false}
```

```
%test with xint syntax
\xintifvalinlist{-1}{0,1,2,3}{true}{false}\\
%test with xint syntax
\xintifvalinlist{3}{0,1,2,3}{true}{false}\\
%boolean macro
\def\myteslist{0,5,10,5,6,9,7,8}
\boolvalinlist{5}{\myteslist}[\resisinlist]\resisinlist

false
true
1
```

```
%counting value, only printing
\countvalinlist*{value}{list}
%counting value, with result in \macro (\rescount by default)
\countvalinlist{value}{list}[\macro]
```

```
%only printing
\countvalinlist*{8}{1,2,3,4,5,6,6,7,8,8,8,9}\\
%storing
\def\tmpcountlist{1,2,3,4,5,6,6,7,8,8,8,9}
\countvalinlist{6}{\tmpcountlist}[\rescountsix]\rescountsix\\
\countvalinlist{10}{\tmpcountlist}[\rescountten]\rescountten

3
2
0
```

3 History

0.1.0: 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.txtf

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{commalists-tools}[2024/09/25 0.1.0 Basic operations for numeral comma separated lists]

%-----History
% 0.1.0 Initial version

%-----Packages
\RequirePackage{listofitems}
\RequirePackage{xintexpr}
\RequirePackage{xstring}

%-----Macros (LaTeX3...) for sorting
\ExplSyntaxOn
\NewDocumentCommand\sortasclist{ s m }
{
  \clist_set:Nx \l_foo_clist {#2}
  \clist_sort:Nn \l_foo_clist{
    \int_compare:nNnTF { ##1 } > { ##2 }
    { \sort_return_swapped: }
    { \sort_return_same: }
  }
  \IfBooleanTF{#1}%if star := just printing // if not, storing
  {%
    {\l_foo_clist}%
  }%
  {%
    \xdef#2{\l_foo_clist}%
  }%
}

\NewDocumentCommand\sortdeslist{ s m }
{
  \clist_set:Nx \l_foo_clist {#2}
  \clist_sort:Nn \l_foo_clist{
    \int_compare:nNnTF { ##1 } < { ##2 }
    { \sort_return_swapped: }
    { \sort_return_same: }
  }
  \IfBooleanTF{#1}%if star := just printing // if not, storing
  {%
    {\l_foo_clist}%
  }%
  {%
    \xdef#2{\l_foo_clist}%
  }%
}
\ExplSyntaxOff

%-----Macros (LaTeX2)
\NewDocumentCommand\addvalinlist{ s m m }{%
  \IfBooleanTF{#1}%if star := just printing // if not, storing
  {%
    #3,#2%
  }%
  {%
    \xdef#3{#3,#2}%
  }%
}
```

```

\NewDocumentCommand\removevalinlist{ s m m }{%
  \readlist*\tmplistread{#3}%
  \xdef\tmpresremovelist{}%
  \xintFor* ##1 in {\xintSeq{1}{\tmplistreadlen}}\do{%
    \xintifboolexpr{ \tmplistread[##1] == #2}%
    {%
      {%
        \xintifboolexpr{ ##1 == 1 }%
        {%
          \xdef\tmpresremovelist{\tmplistread[##1]}%
        }%
        {%
          \xdef\tmpresremovelist{\tmpresremovelist,\tmplistread[##1]}%
        }%
      }%
    }%
  }%
  \IfBooleanTF{#1}%if star := just printing // if not, storing
  {%
    \tmpresremovelist%
  }%
  {%
    \xdef#3{\tmpresremovelist}%
  }%
}

\NewDocumentCommand\boolvalinlist{ m m O{\resisinlist} }{%
  \IfSubStr{, #2,}{, #1,}{\def#3{1}}{\def#3{0}}%
}

\NewDocumentCommand\xintifvalinlist{ m m m m }{%
  \IfSubStr{, #2,}{, #1,}{\xdef\RESTMPVALUE{1}}{\xdef\RESTMPVALUE{0}}%
  \xintifboolexpr{ \RESTMPVALUE == 1}{#3}{#4}%
}

\NewDocumentCommand\countvalinlist{ s m m O{\rescount} }{%
  \readlist*\tmplistread{#3}%
  \xdef#4{0}%
  \xintFor* ##1 in {\xintSeq{1}{\tmplistreadlen}}\do{%
    \xintifboolexpr{ \tmplistread[##1] == #2}%
    {%
      \xdef#4{\fpeval{#4+1}}%
    }%
    {}%
  }%
  \IfBooleanTF{#1}%if star := just printing // if not, storing
  {%
    #4%
  }%
}

\NewDocumentCommand\minoflist{ s m O{\resmin} }{%
  \IfBooleanTF{#1}%
  {%
    \fpeval{min(#2)}%
  }%
  {%
    \xdef#3{\fpeval{min(#2)}}%
  }%
}

\NewDocumentCommand\maxoflist{ s m O{\resmax} }{%
  \IfBooleanTF{#1}%
  {%
    \fpeval{max(#2)}%
  }%
  {%
    \xdef#3{\fpeval{max(#2)}}%
  }%
}

\NewDocumentCommand\meanoflist{ s m O{\resmean} }{%
  \xdef#3{0}%
}

```



```

\readlist*\tmpmeanlist{#2}%
\mintFor* ##1 in {\mintSeq{1}{\tmpmeanlistlen}}\do{%
  \xdef#3{\fpeval{#3+\tmpmeanlist[##1]}}%
}%
\xdef#3{\fpeval{(#3)/\tmpmeanlistlen}}%
\IfBooleanT{#1}%if star := just printing // if not, storing
{
  #3%
}%
}

\NewDocumentCommand\sumoflist{ s m O{\ressum} }{%
  \xdef#3{0}%
  \readlist*\tmpsumlist{#2}%
  \mintFor* ##1 in {\mintSeq{1}{\tmpsumlistlen}}\do{%
    \xdef#3{\fpeval{#3+\tmpsumlist[##1]}}%
  }%
  \IfBooleanT{#1}%if star := just printing // if not, storing
  {
    #3%
  }%
}

\NewDocumentCommand\prodoflist{ s m O{\resprod} }{%
  \xdef#3{1}%
  \readlist*\tmpprodlist{#2}%
  \mintFor* ##1 in {\mintSeq{1}{\tmpprodlistlen}}\do{%
    \xdef#3{\fpeval{(#3)*\tmpprodlist[##1]}}%
  }%
  \IfBooleanT{#1}%if star := just printing // if not, storing
  {
    #3%
  }%
}

\NewDocumentCommand\reverselist{ s m O{\resrevlist} }{%
  \readlist*\tmpreverselist{#2}%
  \xdef#3{\tmpreverselist[-1]}%
  \mintFor* ##1 in {\mintSeq{2}{\tmpreverselistlen}}\do{%
    \xdef#3{#3,\tmpreverselist[-##1]}%
  }%
  \IfBooleanT{#1}%if star := just printing // if not, storing
  {
    #3%
  }%
}

\NewDocumentCommand\getvaluefromlist{ s m m O{\resmyelt} }{%
  \readlist*\tmpreadlist{#2}%
  \xdef#4{\tmpreadlist[#3]}%
  \IfBooleanT{#1}{#4}%
}

\endinput

```