Random C-64/128

rickmk.com/rmk/Com/rand.html

RANDOMIZER

by Rick Kephart 3/20/95

Here is a short, fast, efficient way of finding random numbers, without repeating, as high as 255 in a row. It will then automatically start over again finding a new arrangement of random numbers. It is a 1-line subroutine in C-128 BASIC, or 2 lines on the C-64. The first time it is called, it takes slightly more than 1 second if there are 255 numbers to choose from (less for a smaller range), and then just 1-2 jiffies for the rest of the numbers, until it has to start over.

It uses 2 variables, which are here named U and V, used only within the subroutine. They must be renamed if the program was using variables with those names. There is also a variable R$, which cannot be changed by the program calling the subroutine. If R$ is used somewhere else, give that string variable another name everywhere it occurs in the subroutine.
The random value is returned after a GOSUB to this line, as variable R. You can replace R with any variable name you want.

Replace whatever line number you use with the '10' in 'GOTO 10' in the sample lines.
If you have less than 255 things to choose from, replace the '255' in '1 TO 255' in the sample lines, with the highest number to choose from.

Here is the single line subroutine for the C-128

```
10 U=LEN(R$):IF U=0 THEN FOR U=1 TO 255:R$=R$+CHR$(U):NEXT:GOTO 10:ELSE:
V=RND(0)*U+1:R=ASC(MID$(R$,V,1)):R$=LEFT$(R$,V-1)+RIGHT$(R$,U-V+1):RETURN
```

Here is the 2-line subroutine to use on the C-64

```
10 U=LEN(R$):IF U=0 THEN FOR U=1 TO 255:R$=R$+CHR$(U):NEXT:GOTO 10
20 V=RND(0)*U+1:R=ASC(MID$(R$,V,1)):R$=LEFT$(R$,V-1)+RIGHT$(R$,L-V+1):RETURN
```

HOW IT WORKS

R$ is set up as a string of numbers, in CHR$ form, in order from 1 to 255 or however many numbers you have to choose from.

U is the length of that string. If it is 0, then either R$ has not been set up yet or it has been used up. If necessary, the string R$ is generated (this takes less than 2 seconds).

V is made some random number from 1 to however many choices are left: from 255 (or the maximum) numbers to choose from, to as few as 1 choice left.

Then, MID$ is used to find whatever number is at that spot in R$. ASC turns it into a simple number.

Finally, that number that was chosen is removed from R$. This is done by combining whatever is on the left up to that number to everything on the right after that number. So each time after that the subroutine is called, all the choices are there except whatever numbers have already been selected.

end of file

You can write to me at .

rmkq@lphrc.org