

Hi!

I must first thank you for holding this fun programming challenge tradition. I missed it last year but I enjoyed the related video. I'm impressed by the ingenuity of the community and the various languages solutions are also sometimes quite surprising.

Here's a brief explanation of how my program works:

I noticed that this year's figure is 8-way symmetrical. So I made my program draw a basic shape that I rotated and mirrored 8 times. In the end, this added some complexity to the code and I have the intuition that a more easier solution would be to encode the entire figure in binary and simply decode it to the screen. Nevertheless, I opted for the symmetrical way because it was more fun to code and you said multiple times « Have fun! ».

Line 0 prepares an D\$() array that contains cursor movements (1,2,3,4) and cursor movements + «*» + cursor left (5,6,7,8). A cursor left is necessary after printing a « * » because of the automatic position incrementation of BASIC and the need to stay at the same position when the shape is drawn in other directions.

Line 1,2 prepares a I() array that is used as indirection to D\$() for drawing in the 8 needed directions (4 by rotation and 2 by mirroring). Those directions are 1234, 2341, 3412, 4123, 3214, 2143, 1432, 4321. They're all concatenated in the « 3214321234123 » string to save space by eliminating the overlaps.

Line 3 draws the shape by decoding the cursor movements of the long « 818181... » string where values of 1 to 4 represent cursor movements only and 5 to 8 represent printing a « * » then making a cursor move.

I had a lot of fun optimizing this even if I'm sure others will find more clever ways to do the same figure. The C64 was my first computer and, just for this contest, I installed a MiSTer with a screen and a keyboard that will now be permanently available.

Best regards,

Stéphane Edwardson
La Tuque, Québec, Canada

stedw@icloud.com