

Star Coders Contest

Elementary School Division - Sample Problems

1) Minimum Z

Output the following shape using as few Z's as possible in your code. The output has to be exactly the same. It has 9 rows and 9 characters in each row. Note that it has to be an upper case Z. You have to use at least 1 Z in your code.

OUTPUT:

Copyright © by AlphaStar Academy



2) Maze Walker

Your friend is experimenting with a program that navigates a maze using four commands:

- up move 2 spaces up (cannot cross walls)
- down move 3 spaces down (cannot cross walls)
- left move 1 space left (cannot cross walls)
- right move 2 spaces right (cannot cross walls)

The maze (seen below) consists of walls ('#'), empty spaces ('.'), a starting location ('S'), and an ending location ('E').

MAZE:

Write a sequence of commands that travels from the start and finishes at the end without crossing or landing on any walls.



3) Forbidden Math

Your goal is to multiply two positive integers X and Y, but with a twist: you're not allowed to use any of the following:

- The following arithmetic operators: +, -, *, /, %, ^, |, & The following keywords: if, or, and, range •
- •
- Built-in math libraries or functions •

SAMPLE INPUT	SAMPLE OUTPUT
3 4	12



Middle School Division - Sample Problems

1) Digitized Rectangle

Output the following shape using as few digits (numbers 0 through 9) as possible in your code. The output must be exactly the same. It has 9 rows and 20 characters in each row. You must use at least 1 digit in your code.

OUTPUT:

01234567890123456789
01
23
45
67
89
01
23
45678901234567890123



2) Message Decoder

Your friend at the NSA has received a secret encoded message:

```
r__asucongratnhwdp!melbotsv_latio_oe_sihoeyounvl_
```

They know the following program will decode the message, but they are unfamiliar with the programming language it is written in. Your friend has asked you to help! Trace through the logic below to decrypt the message.

Hint: Your answer will be a 7 x 7 grid of characters with no spaces!

```
array decryptedMessage[7][7]
encryptedMessage = "r__asucongratnhwdp!melbotsv_latio_oe_sihoeyounvl_"
bottom = 6
left = 0
right = 6
count = 0
while count < 49
    for i from bottom to top
        decryptedMessage[i][left] = encryptedMessage[count]
        count = count + 1
    for i from left to right
        decryptedMessage[top][i] = encryptedMessage[count]
        count = count + 1
    top = top + 1
    for i from top to bottom
        decryptedMessage[i][right] = encryptedMessage[count]
        count = count + 1
    for i from right to left
        decryptedMessage[bottom][i] = encryptedMessage[count]
        count = count + 1
    bottom = bottom -1
print decryptedMessage
```



3) Wall Builder

You are given a 10 x 10 grid representing a large field with resources ('R') scattered throughout. Your objective is to build a connected wall of hashmarks ('#') that divides the grid into exactly two connected regions, such that half of the R symbols are on each side of the wall.

It is guaranteed that no two resources are adjacent and there are an even number of resources. A side of the wall is determined to be all of the places reachable by traveling up/down/left/right from a point without stepping on the wall. The wall should split the grid into exactly two sides.

Your output must consist of a 10 x 10 grid of characters with only dots ('.'), resources ('R'), and walls ('#'). You may not remove, cover, or move any of the resources in the given grid.

Example	Solution 1	Solution 2	Solution 3	Solution 4
	#	#		#
.R.R	.R.R#	.R.R.#	.R.R.####.	.R.R.####.
	####	#	#	##
	• • • • • • • • • • •	#	#	#
• • • • • • • • • •		····#····	· · · · · # · · · ·	····#····
• • • • • • • • • •	• • • • • • • • • • •	· · · · # · · · · ·	····#····	····#····
• • • • • • • • • •		· · · · # · · · · · #	· · · · # · · · · · · · · · · · · · · ·	••••#•••
••••		····#····	····#····	##
····K.K.	· · · · · · · K · K ·	#.K.K.	.####.K.K.	•####•.K•K•
• • • • • • • • • •		#		#
Number of '#'s:	6 (best)	10 (good)	invalid (one region)	invalid (four regions)

Try to use the '#'s as few as possible.