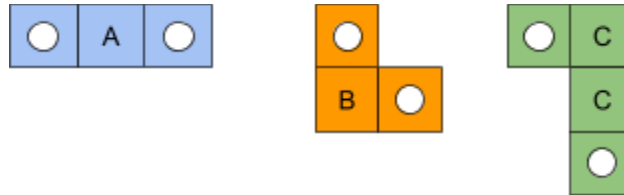


Intermediate Division - Stretch

PROBLEM: Given a rectangular grid and the 3 types of pieces shown below, the object of Stretch is to place pieces in the grid so that they form a connected path from the left side to the right side or from the right side to the left side. If the initial piece is in the leftmost column, the path goes to the rightmost column. If the initial piece is in the rightmost column, the path goes to the leftmost column. Regardless of which direction the path is created, the output is always from left to right.



- A piece cannot be rotated or flipped.
- A piece can only connect to the last piece that was placed and one column to the right of that piece if the path is left-to-right OR one column to the left of that piece if the path is right-to-left.
- A piece can connect only at a tile with a circle and the tiles with the circles are the only tiles that are allowed to touch.
- All tiles of the connecting piece must be to the right (if left-to-right) or to the left (if right-to-left) of all previously placed pieces.
- A piece cannot be placed in the grid such that it would cover any part of another piece, cover a blocked cell, or extend beyond the grid.
- The one and only tile allowed to touch the starting side is a circle tile.
- The one and only tile allowed to touch the opposite side is a circle tile.
- Pieces are placed in alphabetical order. If a piece does not fit, skip it and use the next piece that fits. When Piece C is either used or skipped, then begin again with Piece A.
- Grid cells are numbered consecutively starting with 1 in the upper left corner and continue from left to right and from top to bottom.
- We guarantee that if a piece can be placed, then that will be the only location it can be placed.

1	2	3	4	5	6	7	8	9	10
○	A	○	○	15	16	17	18	19	20
21	22		B	○	○	A	○	○	30
31	32	33	34	35	36		38	B	○
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60

Intermediate Division - Stretch

EXAMPLE: The following example is a 6 x 10 grid with a starting cell of 11. There are blocked cells at cells 23 and 37. Piece A is placed at 11. Piece B can only be placed at 14. The next piece must connect at 26. Piece C cannot be placed at 26 because there is a blocked cell at 37. Therefore, Piece C is skipped. Piece A is placed at Location 26. The next piece must connect at 29. Piece B is placed at 29 and touches the right side at 40. Therefore, the path is ABAB.

INPUT: There will be 5 lines of data. Each line will contain the numbers: r , c , s , n , followed by n numbers. r indicates the number of rows in the grid. c indicates the number of columns in the grid. s indicates the starting cell number for the first piece. n indicates the number of blocked cells. The next n numbers are the cells that are designated as blocked.

OUTPUT: Form a path from the starting cell to the opposite edge of the grid using the algorithm above. Print the sequence of pieces that were used to form the path from left to right.

SAMPLE INPUT <http://www.datafiles.acsl.org/2019/contest3/int-sample-input.txt>

```
6 10 11 2 23 37
4 9 10 2 23 26
5 9 1 2 13 26
4 10 40 2 13 26
5 15 75 1 40
```

SAMPLE OUTPUT

1. ABAB
2. AAA
3. ACBB
4. ABCA
5. ACBABA