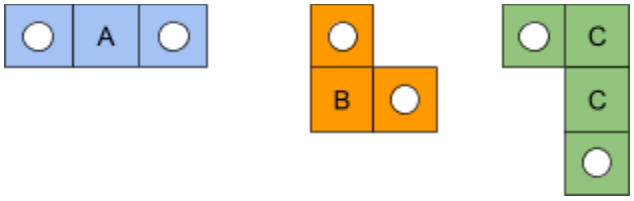
AMERICAN COMPUTER SCIENCE LEAGUE

2018-2019 **Contest #3**

Intermediate Division - Stretch

考号/Exam Code: _____ 姓名/Name:_____ 学校/School:_____

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

```
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

TEST INPUT

```
8 14 15 2 98 81
5 12 60 2 15 48
7 17 119 2 37 22
6 13 52 2 23 35
10 18 19 1 81
```

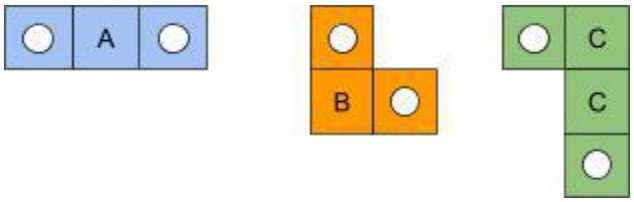
AMERICAN COMPUTER SCIENCE LEAGUE

2018-2019 **Contest #3**

Intermediate Division - Stretch

考号/Exam Code: _____ 姓名/Name:_____ 学校/School:_____

PROBLEM（问题）: 给定矩形网格和下面显示的 3 种类型的连接部件，要将连接部件放入在网格中，以便它们形成从左侧到右侧或从右侧到左侧的连接路径。如果初始块位于最左边的列中，则路径将连接至最右边的列。如果初始块位于最右边的列中，则路径将连接至最左边的列。无论创建路径的方向如何，总是从左到右输出路径。



- 部件不能旋转或翻转。
- 如果路径是从左到右，则一个部件只能连接到最后一个放置的部件和该部件右侧的一列。如果路径是从右到左，则只能连接到该部件的左侧的一列。
- 一个部件只有圆圈部分才能与另一个部件的圆圈部分连接。
- 连接部件必须位于之前所有放置部件的右侧（如果从左到右连接），或者左侧（如果从右到左连接）。
- 在网格中，一个部件不能覆盖另一个部件的任何部分，不能覆盖阻塞的单元，也不能延伸到网格之外。
- 唯一允许接触的起始点部分是一个部件的圆圈部分。
- 唯一允许接触的下一个部件部分也是部件的圆圈部分。
- 部件按英文字母顺序排列。如果一个部件不合适，则跳过它，使用下一个合适的。当部件 C 被使用或跳过时，再次从部件 A 开始。
- 网格单元从左上角的 1 开始连续编号，从左到右、从上到下依次编号。
- 我们保证，如果一个部件可以放置，那么这将是唯一可以放置的位置。

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（示例）: 下例为 6 x 10 网格，起始单元格为 11。单元格 23 和 37 处有阻塞的单元格。部件 A 放置在 11。部件 B 只能放置在 14。下一块必须在 26 连接。部件 C 不能放置在 26，因为 37 处有一个阻塞的单元格。因此，跳过部件 C。部件 A 位于位置 26。下一块必须在 29 连接。部件 B 放置在 29 处，与右侧 40 处接触。因此，该路径是 ABAB。

INPUT（输入）： 将有 5 行数据。每一行将包含数字：r、c、s、n，后跟 n 个数字。r 表示网格中的行数。c 表示网格中的列数。s 表示第一个单元格的开始编号。n 表示阻塞的单元格数。接下来的 n 个数字表示具体阻塞的单元格。

OUTPUT（输出）： 使用上述算法，形成一条从网格左侧的起始单元格到网格另一侧的单个单元格的路径。输出用于形成路径的序列。

SAMPLE INPUT（示例输入）

```
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（示例输出）:

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

TEST INPUT

```
8 14 15 2 98 81
5 12 60 2 15 48
7 17 119 2 37 22
6 13 52 2 23 35
10 18 19 1 81
```