**2017-2018**

# American Computer Science League

**Contest #3**

## INTERMEDIATE DIVISION
## ACSL Walk

**PROBLEM**: Given an array containing both 1's and 0's to direct a walker along a path, find the ending location of the walker. This problem will use an 8x8 array. In the diagram below, location (1, 1) is at the upper left hand corner of the array and contains a 0. If the walker encounters a 0, he continues in the direction he entered that array location. If he encounters a 1, the direction depends on the direction the walker enters that location and the turn value at that location. Each 1 location starts out as a 90 degree turn but increases by 90 degrees each time it is reentered.

If the walker enters (2, 2) from the Left, 90 degrees is a vertical move up, 180 continues in the same direction, 270 degrees is a vertical move down and 360 degrees (or 0 degrees) is a move in the opposite direction. See Diagram #1.

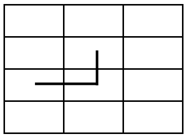
If the walker enters (2, 2) from the Right, 90 degrees is a vertical move down, 180 degrees continues in the same direction, 270 degrees is a vertical move up and 360 degrees is a move in the opposite direction. See Diagram #2.

If the walker enters (2, 2) from Below, 90 degrees is a move to the left, 180 degrees continues in the same direction, 270 degrees is a move to the right and 360 degrees is a move in the opposite direction. See Diagram #3.

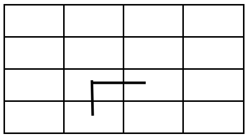
If the walker enters (2, 2) from Above, 90 degrees is a move to the right, 180 degrees continues in the same direction, 270 degrees is a move to the left and 360 degrees is a move in the opposite direction. See Diagram #4.

If a cell value directs a walker to a boundary of the array, he is transported to the opposite side of the array. Ex. If a move starts at (2, 8) and must go right, then the move will be to (2, 1).

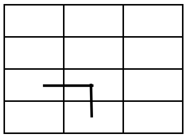
1.



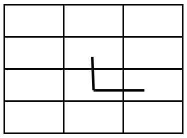
2.



3.



4.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

**2017-2018**

# American Computer Science League

**Contest #3**

## INTERMEDIATE DIVISION
## ACSL Walk

**INPUT**: There will be 6 lines of input. The first line will contain 8 hexadecimal values that when converted to binary will give the row values of the array. If the conversion produces fewer than 8 characters fill with leading zeros. Row #1 is at the top of the array shown above and starts with location (1, 1) on the left. Inputs 2-6 will each contain two integers giving the starting location in row/col order, a one-character string giving the direction **A**bove, **B**elow, **L**eft, **R**ight and an integer giving the number of moves to make. Always revert back to the original array to start each walk.

**OUTPUT**: For each line of input (2 – 6) print the final location of the walker in row/col order.

**SAMPLE INPUT**

1. C, C7, 85, D6, 46, D7, E6, 87
2. 2, 3, L, 2
3. 2, 7, B, 8
4. 4, 5, R, 3
5. 6, 7, A, 5
6. 8, 8, L, 7

**SAMPLE OUTPUT**

1. 2, 5
2. 2, 5
3. 6, 4
4. 3, 7
5. 6, 1

**TEST DATA**

**2017-2018** **Contest #3**

# American Computer Science League

## INTERMEDIATE DIVISION
## ACSL Walk

| TEST INPUT | TEST OUTPUT |
|---|---|
| 1. B2, F, F3, A1, 4E, 3, 78, BC | 1. 1, 5 |
| 2. 1, 4, L, 5 | 2. 5, 3 |
| 3. 6, 5, B, 9 | 3. 3, 7 |
| 4. 4, 3, A, 7 | 4. 3, 5 |
| 5. 8, 8, R, 12 | 5. 8, 6 |
| 6. 1, 1, A, 20 | |