FILE PATHS

argv[1] = MxN (row size and column size of map matrix)
argv[2] = I (size of key matrix)
argv[3] = input file for map matrix
argv[4] = input file for key matrix
argv[5] = output file

PROGRAM DESCRIPTION

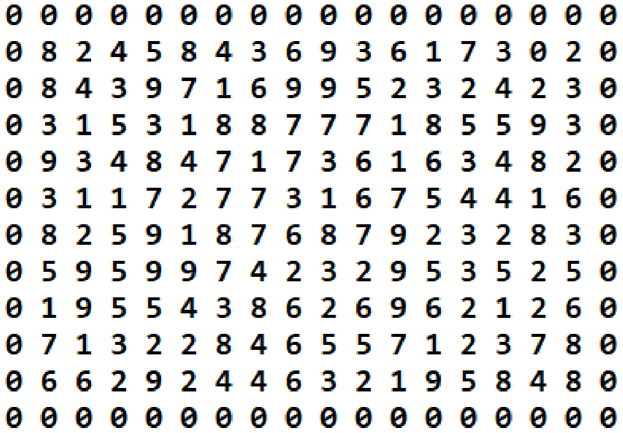
This program finds the hidden treasure within a treasure map designed as a matrix. Program reads the map and key data from input files and keep them in two dimensional arrays. Every element of the map matrix and key matrix is a number. Zeros in the edges of map matrix show the boundary of the map. Key is a square matrix. Size of key matrix is an odd value. By using the key matrix, the hidden treasure be found in the map. Key moved on the map depending on the rules. Search operation start from top left corner on the map matrix. By multiplying the key matrix by the obtained sub-matrix, we find the direction of the next sliding. The direction is computed by taking mod five of multiplication result. The result is interpreted as follows:

0 : FOUND TREASURE
1 : GO UP
2 : GO DOWN
3 : GO RIGHT
4 : GO LEFT

In case of key matrix is placed on the boundary of map where there is no way to slide through the determined direction, it moves in the opposite direction. If the result is zero, the treasure is found. The place of the treasure is the midpoint of the submatrix. Program prints out the midpoints of the submatrix in each step.

INPUT/OUTPUT FORMAT

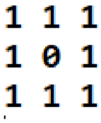
In the map matrix numbers are seperated by spaces and newlines. An example map matrix file:



```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 8 2 4 5 8 4 3 6 9 3 6 1 7 3 0 2 0
0 8 4 3 9 7 1 6 9 9 5 2 3 2 4 2 3 0
0 3 1 5 3 1 8 8 7 7 7 1 8 5 5 9 3 0
0 9 3 4 8 4 7 1 7 3 6 1 6 3 4 8 2 0
0 3 1 1 7 2 7 7 3 1 6 7 5 4 4 1 6 0
0 8 2 5 9 1 8 7 6 8 7 9 2 3 2 8 3 0
0 5 9 5 9 9 7 4 2 3 2 9 5 3 5 2 5 0
0 1 9 5 5 4 3 8 6 2 6 9 6 2 1 2 6 0
0 7 1 3 2 2 8 4 6 5 5 7 1 2 3 7 8 0
0 6 6 2 9 2 4 4 6 3 2 1 9 5 8 4 8 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

(12*18)

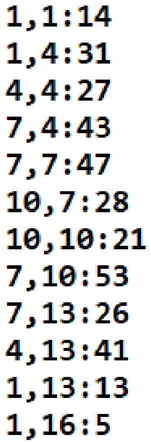
An example key matrix file:



```
1 1 1
1 0 1
1 1 1
```

(3*3)

Each line in the output file contains the midpoint of the submatrix and also the result of multiplication.An example of output file:



```
1,1:14
1,4:31
4,4:27
7,4:43
7,7:47
10,7:28
10,10:21
7,10:53
7,13:26
4,13:41
1,13:13
1,16:5
```

EXECUTION

Programs name is findtreasure.c. It will executed as follows:

```
./findtreasure [row size of map matrix]x[column size of map matrix] [size of key matrix] [map matrix file] [key matrix file] [output file]
```

FUNCTIONS

There are two functions in this program:

- move(…)
- output(…),

move function is a recursive function. It calculates direction and recall itself until treasure is found.

output function take the output array and write it to output file.