

For students with odd PC numbers

There are several ways in which the elements of a matrix can be printed. The standard row-major order of printing is discussed in the class. In this exercise, you are asked to print the elements of a matrix in the clockwise spiral order. You start printing from the $(0, 0)$ -th element and then move right along the 0-th row. Once you hit the right boundary, move downward till you reach the bottom. Then move leftward and print the last row (in the reverse order). Finally, print the 0-th column from bottom to top. Since the 0-th row has already been printed, this upward movement should stop after printing the $(1, 0)$ -th element. You then move right along the 1-st row.

Consider the following 4×3 matrix:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}.$$

The clockwise spiral listing of the elements of A is:

1 2 3 6 9 12 11 10 7 4 5 8.

Complete the following program for printing the clockwise spiral listing of a matrix. More precisely, the `main()` function is provided. Your task is to write the two functions `initMatrix()` and `spiralPrint()`. The function `initMatrix()` takes three arguments: a 2-dimensional array A , a row dimension r and a column dimension c , and initializes A to the $r \times c$ matrix whose (i, j) -th entry is $i^2 + 4j^2$. The other function `spiralPrint()` takes the same three arguments and iteratively prints the elements of the matrix in the spiral order described above.

```
#include <stdio.h>
#define MAXROW 10
#define MAXCOL 10

/* Write the functions initMatrix() and spiralPrint() */

int main ()
{
    int A[MAXROW][MAXCOL];

    initMatrix(A,5,5); spiralPrint(A,5,5);
    initMatrix(A,5,7); spiralPrint(A,5,7);
    initMatrix(A,10,7); spiralPrint(A,10,7);
}
```

You may use the following algorithm for spiral traversal along the array. Maintain four boundaries: top, bottom, left and right. Start at index $(0, 0)$. Move right as long as you do not hit the right boundary. Once you do so, listing of the 0-th row is complete. Increment the top boundary and start downward motion till you hit the bottom boundary. Repeat in this fashion until each element of the array is printed once and only once. The loop terminates when two opposite boundaries (top,bottom or left,right) cross one another.

Sample output

For each of the three matrices indicated in the `main()` function, first print the matrix as a standard two-dimensional array. This printing may be done in the function `initMatrix()`. Try to justify the entries of the matrix neatly as shown below.

Next print the clockwise spiral listing of the elements of the matrix. Take care that your print-out cannot accommodate more than 80 characters in a line. Insert appropriate line breaks during the spiral listing.

The 5x6 matrix is:

```
0  4  16  36  64 100
1  5  17  37  65 101
4  8  20  40  68 104
9  13 25  45  73 109
16 20 32  52  80 116
```

The clockwise spiral listing is:

```
0  4  16  36  64 100 101 104 109 116  80  52  32  20  16  9  4  1  5  17
37 65 68 73 45 25 13  8 20 40
```