Print Matrix Diagonally

Given a 2D matrix, print all elements of the given matrix in diagonal order. For example, consider the following 5 X 4 input matrix.

```
      1     2     3     4
      5     6     7     8
      9    10    11    12
     13    14    15    16
     17    18    19    20
```

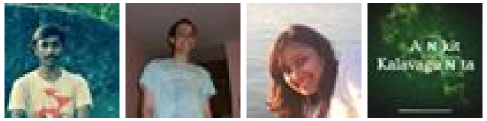
Diagonal printing of the above matrix is

```
      1
      5     2
      9     6     3
     13    10     7     4
     17    14    11     8
     18    15    12
     19    16
     20
```

Following is C++ code for diagonal printing.

The diagonal printing of a given matrix 'matrix[ROW][COL]' always has 'ROW + COL – 1′ lines in output

```
#include <stdio.h>
#include <stdlib.h>
```



Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

```
#define ROW 5
#define COL 4

// A utility function to find min of two integers
int min(int a, int b)
{ return (a < b)? a: b; }

// A utility function to find min of three integers
int min(int a, int b, int c)
{ return min(min(a, b), c);}

// A utility function to find max of two integers
int max(int a, int b)
{ return (a > b)? a: b; }

// The main function that prints given matrix in diagonal order
void diagonalOrder(int matrix[][COL])
{
    // There will be ROW+COL-1 lines in the output
    for (int line=1; line<=(ROW + COL -1); line++)
    {
        /* Get column index of the first element in this line of outpu
           The index is 0 for first ROW lines and line - ROW for remai
           lines  */
        int start_col =  max(0, line-ROW);

        /* Get count of elements in this line. The count of elements i
           equal to minimum of line number, COL-start_col and ROW */
         int count = min(line, (COL-start_col), ROW);

        /* Print elements of this line */
        for (int j=0; j<count; j++)
            printf("%5d ", matrix[min(ROW, line)-j-1][start_col+j]);

        /* Ptint elements of next diagonal on next line */
        printf("\n");
    }
}

// Utility function to print a matrix
void printMatrix(int matrix[ROW][COL])
{
    for (int i=0; i< ROW; i++)
    {
        for (int j=0; j<COL; j++)
            printf("%5d ", matrix[i][j]);
        printf("\n");
```

Popular Posts

- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

```
    }
}

// Driver program to test above functions
int main()
{
    int M[ROW][COL] = {{1, 2, 3, 4},
                        {5, 6, 7, 8},
                        {9, 10, 11, 12},
                        {13, 14, 15, 16},
                        {17, 18, 19, 20},
                       };
    printf ("Given matrix is \n");
    printMatrix(M);

    printf ("\nDiagonal printing of matrix is \n");
    diagonalOrder(M);
    return 0;
}
```

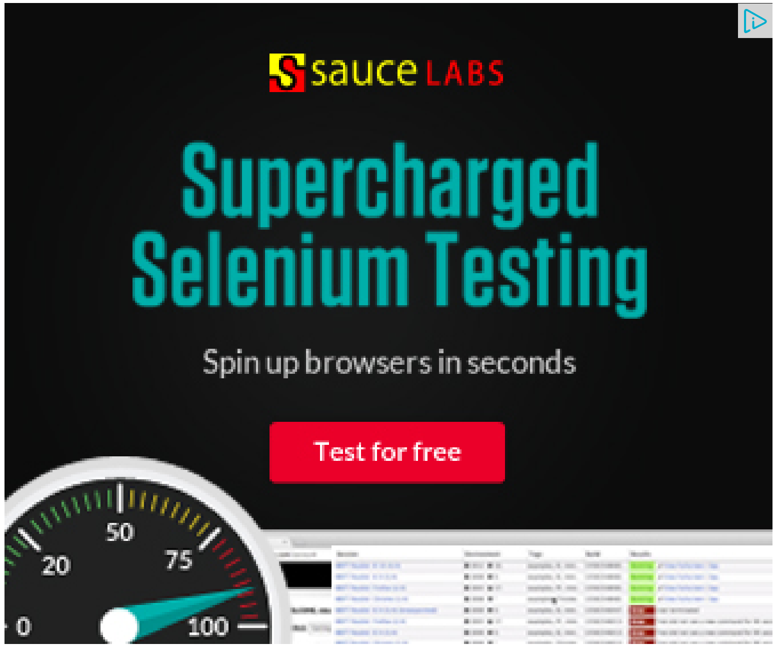
Output:

```
Given matrix is
    1     2     3     4
    5     6     7     8
    9    10    11    12
   13    14    15    16
   17    18    19    20

Diagonal printing of matrix is
    1
    5     2
    9     6     3
   13    10     7     4
   17    14    11     8
   18    15    12
   19    16
   20
```

This article is compiled by Ashish Anand and reviewed by GeeksforGeeks team. Please write

This article is compiled by Ashish Anand and reviewed by GeeksforGeeks team. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



Related Tpoics:

- Remove minimum elements from either side such that 2*min becomes more than max
- Divide and Conquer | Set 6 (Search in a Row-wise and Column-wise Sorted 2D Array)
- Bucket Sort
- Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1
- Find the number of zeroes
- Find if there is a subarray with 0 sum
- Divide and Conquer | Set 5 (Strassen’s Matrix Multiplication)
- Count all possible groups of size 2 or 3 that have sum as multiple of 3

23 Tweet 2 1

Writing code in comment? Please use **ideone.com** and share the link here.

49 Comments GeeksforGeeks

705 Subscribe

Recent Comments

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 9 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 12 minutes ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 37 minutes ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 39 minutes ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

newCoder3006 Code without using while loop. We can do it...

Find subarray with given sum · 1 hour ago

Sort by Newest ▾

Join the discussion…

Venu Gopal · 21 days ago

@geeksforgeeks when will people stop posting 50-100 lines of code in the cor their idea followed by the link of their code

^ | ⌄ · Reply · Share ›

alien · 2 months ago

can u please explain the algorithm?

^ | ⌄ · Reply · Share ›

raj · 3 months ago

print or access elements of a general matrix was never so straightforward. Th needed it to write a parallel code for finding the longest common subsequence

^ | ⌄ · Reply · Share ›

Nag · 3 months ago

```
class MatrixDiagonalDisplay {

public static void main(String[] args) {

int [][] ipArray = {{1, 2, 3, 4},

{5, 6, 7, 8},

{9, 10, 11, 12},

{13, 14, 15, 16},

{17, 18, 19, 20},
};

int m = 5, n = 4;

displayArray(ipArray, m, n);

displayArrayAsDigonals(ipArray, m,n);
```

see more

• Reply • Share ›

Nag • 3 months ago

```
class MatrixDiagonalDisplay {

public static void main(String[] args) {

int [][] ipArray = {{1, 2, 3, 4},

{5, 6, 7, 8},

{9, 10, 11, 12},

{13, 14, 15, 16},

{17, 18, 19, 20},

};

int m = 5, n = 4;

displayArray(ipArray, m, n);

displayArrayAsDigonals(ipArray, m,n);
```

see more

• Reply • Share ›

Venu Gopal ➦ Nag • 21 days ago

why do you need such a long function when u can minimize the code l
body... also the original code is not so much complex, If you are not giv
write this much sized naive code in the comment

^ | ˅ • Reply • Share ›

Sunny • 7 months ago

```
void printDiagonal(int (&m)[100][100], int row, int col)
{
int i = 0, j = 0;
        for(int k = 0; k < row + col - 1; k += 2)
        {
                while(j >= 0 && i < row)
                {
                        printf("%d ", m[i][j]);
                        i++;
j--;
}
printf("\n");

                if(i == row)
i--, j += 2;
else
j++;
```

see more

1 ^ | ˅ • Reply • Share ›

anonym ➦ Sunny • 7 months ago

```
void printDiagonal(int (&m)[100][100], int row, int col)
```

```
{
int i = 0, j = 0;

for(int k = 0; k < row + col - 1; k += 2)
{
while(j >= 0 && i < row)
{
printf("%d ", m[i][j]);
i++;
j--;
}
printf("\n");

if(i == row)
i--, j += 2;
else
j++;
```

see more

Reply • Share ›

Budweiser • 10 months ago

Can't it be done in O(n) time complexity with using queue and some auxiliary s

Enqueue left top
while(!queue.isEmpty())
dequeue element. Enqueue its right and lower elements if exists and havn't be

Correct me if I am wrong

Reply • Share ›

Aravindan B Budweiser • 6 months ago

I think there is a pattern in printing the newline in this method.
say 4 x 2
ROW = 4 col = 2
if(max(ROW,col)-1) - ODD then the \n is to printed after 1, 2, 3, 2, 1 el
elements after which \n has to be printed)
ROW = 4 col = 5
if(max(ROW,col)-1 is even) pattern is 1, 2, 3, 4, 4, 3, 2, 1

This pattern follows for all m x n. So instead of printing the element de
the queue itself, we can print the \n in the above fashion. Correct me if

^ | ˅ • Reply • Share ›

Aravindan B ➦ Aravindan B • 6 months ago

Sorry it is max(Row, col)-1 is odd or even.

^ | ˅ • Reply • Share ›

Ankit Chaudhary ➦ Budweiser • 7 months ago

ur approach is working, but one modification is required to denote whe
new line.

^ | ˅ • Reply • Share ›

Guest ➦ Budweiser • 7 months ago

ur approach is working, but one modification is reqiured to denote whe
new line.

Here is working code in c++.

```
#include<cstdio>
#include<queue>
#include<vector>
using namespace std;

void printDiagonally(vector<vector<int> > arr)
```

```
{
if(!arr.size()) return;
int row=arr.size();
int col=arr[0].size();
int i,j;
queue<pair<int,int> > q; // queue of 2d index
q.push(make_pair(0,0));
q.push(make_pair(-1,-1));
```

see more

^ | ⌄ • Reply • Share ›

Naruto_Koder • 10 months ago

Hope this will do :)

```
#define ROW 5
#define COL 4
void diag(int a[][COL])
{
int maxc=COL,maxr=ROW,i;
int r=ROW-1,l=0;
while( r>=0 && maxc>l )
{
for(i=0;i<=r;i++) printf("%d\n",a[i][l]);
l++;
for(i=l;i<maxc;i++) printf("%d \n",a[r][i]);
r--;
printf("\n");
}
}
int main()
{
```

see more

• Reply • Share ›

saroj07 • 10 months ago

<script type="text/javascript" src="http://ideone.com/api/embed.js..."></script

• Reply • Share ›

saroja • 10 months ago

```
#include
#define ROW 5
#define COL 4
void printMatrix(int M[ROW][COL]);

void diagonalOrder(int M[ROW][COL])
{
int a=1;
int i,j,k,l=1;
for(i=0;i<9;i++)
{
k=i;

if(k-1 < 0)
{
printf("%d ",M[k][k]);
}
else
{
```

see more

• Reply • Share ›

saroj • 10 months ago

just chek thi soln..

• Reply • Share ›

skulldude • 10 months ago

How about the following code? It does not require any in-depth analysis of the elements.

```
 void printMatrixDiagonally(int a[MAX_DIM][MAX_DIM],int r,int c){
        if(!a || r<=0 || c<=0)
                return;

        for(int i=0;i<r;++i)
                printDiagonal(a,i,0,c);

        for(int j=1;j<c;++j)
                printDiagonal(a,r-1,j,c);

        return;
}

void printDiagonal(int a[MAX_DIM][MAX_DIM],int x,int y,int c){
        int i=x,j=y;
```

see more

^ | ˅ • Reply • Share ›

Ashish • 11 months ago

Here is a simple C++ implemntation using two variables.

```
 #include<iostream>
using namespace std;
int main()
{
    int row,col,i,j;
```

```
cin>>row>>col;
int ar[row][col];
for(i=0;i<row;i++)
for(j=0;j<col;j++)
cin>>ar[i][j];
int a=0,b=0,a1=a,b1=b;
cout<<ar[a1][b1]<<endl;
while(true)
{

    if(a<row-1)
```

see more

Reply • Share ›

bpsingh • 11 months ago

//m=row count, n=column count, arr=Array that holds input

```
  for(i=0; i<m+n-1; i++){
  j=i<m?0:i-m+1; //initialize starting column position
  k=i<m?i:m; //initialize starting row position
  while(k>=0 && j<n){
   printf("%d ", arr[k][j]);
   k--; j++;
  }
 }
```

Reply • Share ›

bpsingh bpsingh • 11 months ago

missed printf("\n") after while loop

Reply • Share ›

pritybhudolia • a year ago

```
#include<stdio.h>
#include<conio.h>
# define R 5
# define C 4
int main()
{
int i=0,j,flag=0,count,k=0,p=0;
int M[6][5] = {{1, 2, 3, 4,5},
{6, 7, 8,9,10},
{11, 12,13,14,15},
{16, 17, 18,19,20},
{21, 22,23, 24,25},
{26,27,28,29,30}
};
count=(R+1)+(C+1)-1;

/* int M[5][4] = {{1, 2, 3, 4},
{5, 6, 7, 8},
```

see more

^ | ⌄ • Reply • Share ›

prity • a year ago

```
 /* #include<stdio.h>
#include<conio.h>
# define R 5
# define C 4
int main()
{
     int i=0,j,flag=0,count,k=0,p=0;
```

```
  int M[6][5] =         {{1, 2, 3, 4,5},
                     {6, 7, 8,9,10},
                     {11, 12,13,14,15},
                     {16, 17, 18,19,20},
                     {21, 22,23, 24,25},
                     {26,27,28,29,30}
                    };
  count=(R+1)+(C+1)-1;

/*  int M[5][4] = {{1, 2, 3, 4},
                     {5, 6, 7, 8},
```

see more

⌃ | ⌄ • Reply • Share ›

Muthukumar Suresh • a year ago

isnt this a much simpler solution:

```
[sourcecode language="C++"]
#include<iostream>
using namespace std;
#define row 5
#define column 4
void diagonalOrder(int abc[row][column]){
int i,j,k;
for(i=0;i<row;i++){
j=0;k=i;
while(k>=0&&j<=(column-1)){
cout<<abc[k][j]<<"\t";
k=k-1;
j=j+1;
}
cout<<"\n";
}
```

see more

^ | ⌄ • Reply • Share ›

cyberphyte • a year ago

```
[sourcecode language="C++"]
#include<iostream>
#include <stdio.h>
#include <stdlib.h>

#define ROW 5
#define COL 4
void printMatrix(int matrix[ROW][COL])
{
for (int i=0; i< ROW; i++)
{
for (int j=0; j<COL; j++)
printf("%5d ", matrix[i][j]);
printf("\n");
}
}
void print_matrix_diagonally(int a[][COL])
{
```

see more

^ | ⌄ • Reply • Share ›

AAA • a year ago

I think this will not work for something like this:

1 2 3 4 5 6

7 8 9 10 11 12
13 14 15 16 17 18

Reply • Share ›

aasshishh ➦ AAA • a year ago

Hi AAA

Thanks for pointing this out. The post have been updated. Cheers :)

Reply • Share ›

anonymous • a year ago

```
 void diagonalOrder(int a[ROW][COL]){
     int i,j;
     int flag;
     for(i=1;i<=ROW+COL-1;i++){
                              int startcol = max(0,i-ROW);
                              int count = min(i,COL-startcol);
                              flag = i<ROW?i:5;
                              for(j=0;j<count;j++){
                                                   flag--;
                                                   printf("%5d",a[fla
                              }
                              printf("\n");
     }
}
```

Reply • Share ›

anonymous ➦ anonymous • a year ago

```
change flag = i<ROW?i:5; to flag = i<ROW?i:ROW;
```

Reply • Share ›

Kanhaiya · a year ago

https://github.com/kanhaiyakum...

⌃ | ⌄ · Reply · Share ›

gh05t · a year ago

here you go,working code in ruby

```
[sourcecode language="Ruby"]
n=gets.to_i
m=gets.to_i
A=Array.new(m){Array.new(n)}
i=j=0
while i<m
while j<n
A[i][j] =gets.to_i
j+=1
end
i+=1
j=0
end

B=Array.new(m+n-1) {Array.new}

i=j=0
```

see more

⌃ | ⌄ · Reply · Share ›

rajiv kumar · a year ago

```
public static void main(String[] args) {
            int[][] arr = {{1,2,3},{4,5,6},{7,8,9}};
            int i=0,j=0;
```

```
        for(int k=0; k<arr.length + arr[0].length - 1; k++){
            for(int a=i,b=j; a>=0 && (b<=i || b<arr[0].le
                System.out.print(arr[a][b]+" ");
            System.out.println();
            int temp = (k<arr.length-1)?i++:j++;
        }
    }
```

^ | ⌄ • Reply • Share ›

Sambasiva • a year ago

```
#include <stdio.h>

void printDiagnolUpwards(int r, int c, int m[][c]) {
 int i, j, k;

 for (i = 0; i < r; i++) {
  for (j = i, k = 0; k < c && j >= 0; j--, k++) {
   printf("%d ", m[j][k]);
  }
  printf("\n");
 }

 for (i = 1; i < c; i++) {
  for (j = r - 1, k = i; k < c && j >= 0; j--, k++) {
   printf("%d ", m[j][k]);
  }
  printf("\n");
 }
```

see more

^ | ⌄ • Reply • Share ›

pooja • a year ago

```
int i,j,k;
for(k=0;k=0)
{
printf(matrix[i][j]);
i--;
j++;
}
printf("\n");
}
k=row-1;
m=1;
while(m=0)
{
printf(matrix[i][j]);
i--;
j++;
}
printf("\n");
}
```

^ | ⌄ • Reply • Share ›

somesh • a year ago

```
 void printDiagonal(int mat[][], int m, int n)
{
int row=0, col=0, start_row, start_col;
while(col < n)
{
start_row = row;
start_col = col;
while(row>=0 and col<n)
{
printf("%d ",mat[row][col]);
row--;
col++;
}
if(row < m-1)
{
row = start_row+1;
col = start_col;
}
else
{
row = m-1;
col = start_col + 1;
}
}
}
```

^ | ˅ • Reply • Share ›

$am • a year ago

```
//Driver.cpp
#include
using namespace std;
class matrix
{
int row,col;
int **ptr;
public:
matrix(int x,int y)
{
```

```
cout<<"matrix is\n";
row = x;
col = y;
ptr = (int **)new int[row];
for(int i = 0;i<row;i++)
{
ptr[i]=new int [col];
}
```

see more

^ | ⌄ • Reply • Share ›

Vishesh Srivastava • a year ago

```
 #include<stdio.h>
void print(int ar[100][100],int r,int c,int i,int j)
{
    if(i>=r||i<0||j>=c||j<0)
        return;
    printf("%d\t",ar[i][j]);
    print(&ar[0][0],r,c,i-1,j+1);
}
int main()
{
    int i,j,r,c,ar[100][100];
     ar=[[1,2,3,4,5],[6,7,8,9,10],[11,12,13,14,15],[16,17,18,19,20],[2
r=5;
c=5;
    for(i=0;i<r;i++)
    {
        print(&ar[0][0],r,c,i,0);
        printf("\n");
```

see more

• Reply • Share ›

Sunil Kumar • a year ago

Java function ::

```
public static void diagonalOrder(int matrix[][], int row, int col)
{
int x = 0;

for(int i=0; i= row)
{
k = row-1;
x++;
j = x;
}

while(j = 0)
{
System.out.print(matrix[k][j] + "\t");
j++;
k--;
}

System.out.println();
}
}
```

• Reply • Share ›

Sunil Kumar ➦ Sunil Kumar • a year ago

```
public static void diagonalOrder(int matrix[][], int row, int col)
{
int x = 0;
for(int i=0; i= row)
{
k = row-1;
x++;
j = x;
}

while(j = 0)
{
System.out.print(matrix[k][j] + "\t");
j++;
k--;
}

System.out.println();
}
}
```

Reply • Share ›

asitdhal • a year ago

```
 #include <stdio.h>
#include <stdlib.h>

#define ROW 5
#define COL 4

// Utility function to print a matrix
void printMatrix(int matrix[ROW][COL])
{
    for (int i=0; i< ROW; i++)
    {
        for (int j=0; j<COL; j++)
```

```
            printf("%3d ", matrix[i][j]);
        printf("\n");
    }
}

void diagonalOrder(int matrix[ROW][COL])
```

see more

Reply • Share ›

dk • a year ago

```
[sourcecode language="C++"]
#include<iostream>
#define row 5
#define col 4
using namespace std;
int main()
{ int a[row][col],i,j;
for(i=0;i<row;i++)
{for(j=0;j<col;j++)
cin>>a[i][j];
}
int len=row+col-1;
for(i=0;i<len;i++)
{
for(int k=i,j=0;k>=0&&j<col;k--,j++)
{
if(k>=row)
continue;
else cout<<a[k][j]<<"\t";
}
cout<<"\n\n";
```

}
return 0;
}

^ | ⌄ • Reply • Share ›

Niks • a year ago

```
 void diagonalOrder(int matrix[][COL])
{
        for(int i=0; i<(ROW+COL); i++)
        {
                static int x = 0;
                int k = i;
                int     j = 0;
                if(k >= ROW)
                {
                        k = ROW-1;
                        x++;
                        j = x;
                }

                while(j < COL && k>=0)
                {
                        printf("%d \n", matrix[k][j]);
                        j++;
```

see more

^ | ⌄ • Reply • Share ›

Sreenivas Doosa • a year ago

```
 for(int i = 0; i < ROWS + COLS - 1; i++)
{
        int row, col;

        if(i < ROWS)
        {
                row = i;
                col = 0; // first column
        }
        else
        {
                row = ROWS - 1; // last row
                col = i - ROWS + 1;
        }

        while(row >= 0 && col < COLS)
        {
```

see more

^ | ⌄ • Reply • Share ›

Aditya • a year ago

```
[sourcecode language="Python"]
def matrix_diagnol(matrix,row,col):
count = 0
lines = row + col
while count < row:
temp = count
result = ""
for i in range(0,count+1):
result = result +str(matrix[temp][i])+" "
temp = temp - 1
print result
count = count + 1
```

count = 1
Column = 1
while count < col+1:
result = ""
start = row - 1

see more

^ | ⌄ • Reply • Share ›

atul • a year ago

another way of printing :-

```
void diagonalOrder(int matrix[][COL])
{
int line,row=0,col=0,nextCol=0;
        for (line=2; line<=(ROW + COL); line++)
        {
                while(row>=0 && col>=0)
                {
                        if(col>=COL)
                                break;
                        printf("%5d ",matrix[row][col]);
                        row--;
                        col++;
                }
                if(line<=ROW)
                {
```

see more

^ | ⌄ • Reply • Share ›

atul ➦ atul • a year ago

changing while loop to below one.

```
while(row>=0 && col<COL)
       {
           printf("%5d ",matrix[row][col]);
           row--;
           col++;
       }
```

^ | ⌄ • Reply • Share ›

kiran • a year ago

This can be done by printing the upperhalf of diagonal matrix and Lower half in

```
[sourcecode language="JAVA"]
public class MatrixDiagonalPrint {

/**
* @param args
*/
public static void main(String[] args) {
int rows = 5, cols = 4;
int[][] a = new int[rows][cols];
int k=0;
for(int i=0; i<rows; i++) {
for(int j=0; j<cols; j++) {
a[i][j] = ++k;
}
}
System.out.println("Original Matrix");
for (int[] is : a) {
```

⌃ | ⌄ • Reply • Share ›

shadow ➦ kiran • a year ago

Did same way but i like the solution presented above

```
/* Paste your code here (You may delete these lines if not wr:
```

⌃ | ⌄ • Reply • Share ›

yelnatz • a year ago

Have 2 crawlers. One goes down the column first, the other goes right along t

```
 // given matrix[ROW][COLUMN]
// row = last row, column = last column

int Arow = 0, Acol = 0, Brow = 0, Bcol = 0;
int tempRow, tempCol;

do{
 tempRow = Arow; tempCol = Acol;

 // print first crawler
 cout << matrix[Arow][Acol] << " ";

 // print the stuff in between them
 while( tempRow != Brow || tempCol != Bcol ){
  tempRow--;
  tempCol++;
```

see more

⌃ | ⌄ • Reply • Share ›

✉ Subscribe D Add Disqus to your site