UNIQUE MATRIX!!

# Given an odd integer K, find a matrix of size K x K with following conditions:

* Each cell contains an integer from 1 and K (inclusive).
* All 1’s must be at every possible distance from the centre of the matrix.
* No integer appears twice in the same row or the same column.

NOTE: - Following things must be assumed:

* The distance between cell (x, y) and (i, j) is equal to |x-i|+|y-j|.
* The centre of a K x K square is cell ((K+1)/2, (K+1)/2) for odd K.
* For a value K, all possible distances from centre will be 1,2, 3…up to (K-1). So, 1’s should be at all these distances from the centre.

Examples:

Input: K = 1

Output: 1

Input: K = 3

Output: 3 2 1

1 3 2

2 1 3

**We strongly recommend you to minimize your browser and try this yourself first.**

**Explanation And Approach: -**

The idea is very simple, for a given K and Array [] [],we keep the 1 at the position Array[1][K] (rightmost column)And then fill that column down till Array[K][K] with values after1, i.e. 2,3, 4,..K

in the successive cells.Then move leftmost column at Array [2][1], check if that column doesn’t

have 1, if it has 1 in that column, proceed to the next column and fill 1 in the cell (if that column

has no 1) and again move down to fill the remaining numbers i.e. 2,3,4,5,..K like before. If the row ends, move to the upper row and continue until you reach the cell from where you had started.

Again, move rightmost column and check if that column contains 1 or not and continue the process.

Ultimately you will get the array with desired properties.

Below is C++ implementation of above idea.

// C++ program for above approach

#include <iostream>

#include<bits/stdc++.h>

using namespace std;

int arr[100][100];

//Declaring the 2-D array

**void** UNIQUE\_MATRIX(int K)

**{**

int i, j, z;

j = K;

z = 1;

//Filling the 1’s in the desired positions

**for** (i=1; i <= K; i++)

{

if (i%2 == 0)

{

arr[i][z] = 1;

z++;

}

else

{

arr[i][j] = 1;

j--;

}

}

//defining an array a[100] and initialising it with integers from 1 to 100

int a[100];

**for** (i = 1; i <= 100; i++)

a[i] = i;

//Filling remaining integers in the array arr[][]

**for** (j=1; j <= K; j++)

{

**for** (i=1; i<=K; i++)

{

**if**(arr[i][j] == 1)

{

**for** (z=2; z <= K; z++)

{

**if**((i+z-1) %K==0)

arr[K][j] = z;

**else**

arr[(i+z-1) %K] [j]=z;

}

}

}

}

}

// Driver program to test above function

int main()

{

int i, j;

int K = 3;

//Passing K to UNIQUE\_MATRIX function

UNIQUE\_MATRIX (K);

//Printing the desired unique matrix

**for** (i=1; i <= K; i++)

{

**for**(j=1 ;j <= K; j++)

{

printf(**"%d ",**arr[i][j]);

}

printf (**"\n"**);

}

**return** 0;

}

CONTRIBUTED BY: -

Name: **Mazhar Imam Khan**

My GeeksForGeeks A/C link: http://qa.geeksforgeeks.org/user/Mazhar+MIK

Username: **Mazhar MIK**