A5-Q2

Due on 2019-09-07, 23:59 IST

Course outline

About the Course

Announcements

Ask a Question

Progress

Mentor

How to access the portal

Assignment 0

Introduction - Variables, Expressions and Conditionals

Assignment 1

Loop Constructs in C

Assignment 2

More on Data Types and Operations

Functions

Assignment 3

Arrays and Pointers
Consider a matrix M of integers. Divide M into 4 sub-matrices. These sub-matrices are called as Quadrants. Report the Quadrant number which has the smallest minimum-element. If two or more quadrants have same smallest minimum, report the smallest quadrant index.

The matrix M is divided into four quadrants by halving the rows and columns. If row/column is an odd number, divide them in such a way that the first half of the row/column should be one smaller than the second half.

The four quadrants are numbered from 1 to 4 in the structure shown below:

```
Q1 | Q2
---+---
Q3 | Q4
```

**INPUT FORMAT:**
M is a matrix of integers. You would be given two numbers m and n specifying the number of rows and columns. This would be followed by m lines of n integers each specifying the data of the m*n matrix M.

n and m will be greater than 1 and less than 12.

**OUTPUT FORMAT:**
Print in a separate line, the quadrant number with the smallest minimum-element.

**Sample Test Cases**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case 1</td>
<td></td>
</tr>
</tbody>
</table>
| 2 3
| 1 2 3
| -1 3 4        | 3      |
| Test Case 2    |        |
| 4 4
| -10 5 6 7     | 1      |
| 1 2 3 4        |
| 8 9 6 -5       |
| -1 -1 -1 -1    |
Test Case 3

```
4 4
-10 5 6 7
1 2 3 4
8 9 6 -5
-1 -1 -1 -100
```

Test Case 4

```
2 2
-1 2
3 -1
```

Test Case 5

```
3 3
1 2 -1
-1 1 2
-9 2 3
```

The due date for submitting this assignment has passed.

As per our records you have not submitted this assignment.

Sample solutions (Provided by instructor)

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

int minimum(int matrix[11][11], int tx, int ty, int bx, int by) 
{
    int min = 32760; // a large positive number
    int i, j;
    for (i = tx; i <= bx; i++)
        for (j = ty; j <= by; j++)
            if (matrix[i][j] < min)
                min = matrix[i][j];
    return min;
}

void scan_matrix(int matrix[11][11], int rows, int columns)   //Scan
{
    int i,j;
    for( i = 0 ; i<rows ; i++)
        for( j = 0 ; j<columns ; j++)
            scanf("%d", &matrix[i][j]);
}

int main(void)
{
    int rows , columns;
    int matrix[11][11]; //Define two-dimensional matrix
    scanf("%d",&rows);
    scanf("%d",&columns);
    scan_matrix(matrix , rows , columns);  //Function call to scan the
    int quad1 = minimum(matrix, 0, 0, rows/2 - 1, columns/2 - 1);
    int quad2 = minimum(matrix, 0, columns/2, rows/2 - 1, columns - 1);
    int quad3 = minimum(matrix, rows/2, 0, rows - 1, columns/2 - 1);
    int quad4 = minimum(matrix, rows/2, columns/2, rows - 1, columns);
    int min = 32767; //Largest possible int value
    int min_quad_no = 0;  //Some initially set value
    if(quad1<min){min = quad1; min_quad_no = 1;}
    if(quad2<min){min = quad2;min_quad_no = 2;}
    if(quad3<min){min = quad3;min_quad_no = 3;}
    if(quad4<min){min = quad4;min_quad_no = 4;}
    printf("%d\n",min_quad_no); //Printing the Quadrant Number with the
    return 0;
```
52: }