

## Day 1 Task 3

### Quality of Living

Cities in Alberta tend to be laid out as rectangular grids of blocks. Blocks are labeled with coordinates  $0$  to  $R - 1$  from north to south and  $0$  to  $C - 1$  from west to east.

The quality of living in each particular block has been ranked by a distinct number, called *quality rank*, between  $1$  and  $R * C$ , where  $1$  is the best and  $R * C$  is the worst.

The city planning department wishes to identify a rectangular set of blocks with dimensions  $H$  from north to south and  $W$  from west to east, such that the median quality rank among all blocks in the rectangle is the best.  $H$  and  $W$  are *odd* numbers not exceeding  $R$  and  $C$  respectively. The *median quality rank* among an odd number of quality ranks is defined to be the quality rank  $m$  in the set such that the number of quality ranks better than  $m$  equals the number of quality ranks worse than  $m$ .



You are to implement a procedure **rectangle**( $R, C, H, W, Q$ ) where  $R$  and  $C$  represent the total size of the city,  $H$  and  $W$  represent the dimensions of the set of blocks, and  $Q$  is an array such that  $Q[a][b]$  is the quality rank for the block labeled  $a$  from north to south and  $b$  from west to east.

Your implementation of **rectangle** must return a number: the best (numerically smallest) possible median quality rank of an  $H$  by  $W$  rectangle of blocks.

Each test run will only call **rectangle** once.

#### Example 1.

```
R=5, C=5, H=3, W=3,
Q=  5 11 12 16 25
    17 18  2  7 10
    4 23 20  3  1
    24 21 19 14  9
    6 22  8 13 15
```

For this example, the best (numerically smallest) median quality rank of  $9$  is achieved by the middle-right rectangle of  $Q$  shown in bold. That is,

**rectangle**( $R, C, H, W, Q$ ) =  $9$

#### Example 2.

```
R=2, C=6, H=1, W=5,
Q=  6  1  2 11  7  5
    9  3  4 10 12  8
```

For this example the correct answer is  $5$ .

## Day 1 Task 3

---

### Quality of Living

- 1 Subtask [20 points].** Assume  $R$  and  $C$  do not exceed 30.
- 2 Subtask [20 points].** Assume  $R$  and  $C$  do not exceed 100.
- 3 Subtask [20 points].** Assume  $R$  and  $C$  do not exceed 300.
- 4 Subtask [20 points].** Assume  $R$  and  $C$  do not exceed 1 000.
- 5 Subtask [20 points].** Assume  $R$  and  $C$  do not exceed 3 000.

#### Implementation Details.

- Use the *RunC programming and test environment*
- Implementation folder: `/home/ioi2010-contestant/quality/` (prototype: *quality.zip*)
- To be implemented by contestant: `quality.c` or `quality.cpp` or `quality.pas`
- Contestant interface: `quality.h` or `quality.pas`
- Grader interface: none
- Sample grader: `grader.c` or `grader.cpp` or `grader.pas`
- Sample grader input: `grader.in.1`, `grader.in.2` etc.  
*Note: The first line of input contains:  $R, C, H, W$  The following lines contain the elements of  $Q$ , in row-major order.*
- Expected output for sample grader input: `grader.expect.1`, `grader.expect.2` etc.
- Compile and run (command line): `runc grader.c` or `runc grader.cpp` or `runc grader.pas`
- Compile and run (gedit plugin): *Control-R*, while editing any implementation file.
- Submit (command line): `submit grader.c` or `submit grader.cpp` or `submit grader.pas`
- Submit (gedit plugin): *Control-J*, while editing any implementation or grader file.