C++ Programming Multidimensional Arrays Homework 2

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher PhD from Simon Fraser University - Canada Bachelor / Msc from Cairo University - Egypt Ex-(Software Engineer / ICPC World Finalist)

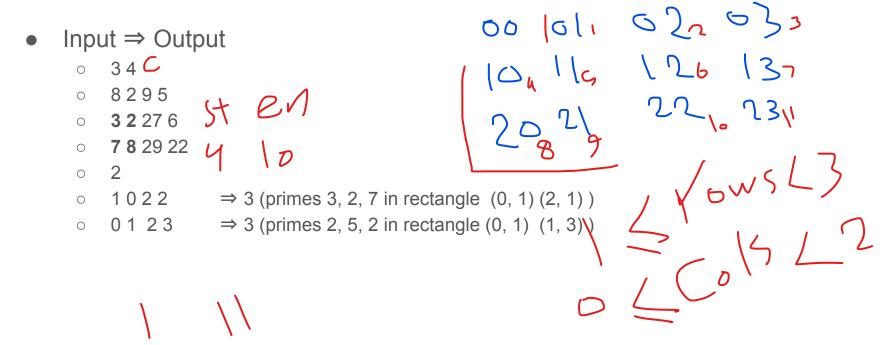


Problem #1: How many primes \(\J \ighta \(\cdot \)



- Read 2 integers N and M, then Read matrix NxM.
- Then read integer Q, for Q queries.
- For each guery read 4 integers: i i r c
 - Represents a grid (submatrix): **top left** (i, j) and (r, c) for (# rows, # cols)
- For each query, print how many <u>prime</u> numbers in the requested grid.
 - Each query should be answered using nested loops maximum not more
 - In future: with smart precomputation it can be done without loops
 - It is called Image integral in Computer Vision field.

Problem #1: How many primes



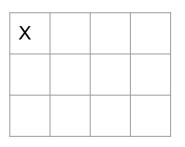
Problem #2: Find mountains

- Read integers N, M, then Read matrix NxM. Print all positions that are mountain. Position is mountain if its value > 8 neighbours values
- Code smartly
- Input
 - 0 33
 - 0 861
 - 0 329
 - 0 164
- Output
 - o 00 (8 > 6, 3, 2)
 - o 12 (9 > 1, 2, 5, 4, 6)

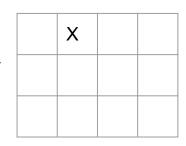
Problem #3: Active Robot

- Read integers N, M represents a matrix. A robot start at cell (0, 0).
- Read integer K, then K commands. Each command is 2 values
 - Direction from 1 to 4: up, right, down, left
 - Steps: a number to number steps to take in the direction. Steps [1, 1000000000]
 - If the robot hits the wall during the move, it **circulates** in the matrix.
 - o For every command, print where is the robot now
- Input
 - 34 4 21 32 42 13
 - 2 1 means to right 1 step 3 2 means down 2 steps
- Output
 - 0 (0, 1) (2,1) (2, 3) (2, 3)

Problem #3: Active Robot



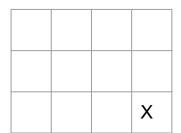
2 1 (right 1 step) \Rightarrow New pos (0, 1)



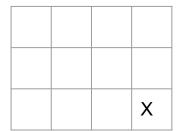
3 2 (down 2 steps) = New pos (2, 1)

⇒		
	X	

4 2 (left 2 steps) ⇒ New pos (2, 3)
Circulation



1 3 (up 3 steps) ⇒ New pos (2, 3) Circulation



Problem #4: Flatten 3D Array

- 10/1/0/5-
- Read 3 numbers: DEPTH, ROWS, COLS the dimensions of 3D array
- Then read integer either 1 (convert 3D to 1D) or 2 (1D to 3D)
- If input was 1, then read 3 integers d, r, c then convert to position in 1D array
- If input was 2, then read 1 integer position, then convert to 3D array position
- Try to generalize if we have e.g. 6D array

```
Input ⇒ Outputs' 3 √15
                                        int idx = 0;
     3 4 5 2
                20
                        \Rightarrow 100
                                            for (int row = 0; row < 4; ++row)
                                                for (int col = 0; col < 5; ++col)</pre>
     3 4 5
           1 \ 111 \Rightarrow 26
                                                    cout<<idx++ << "
                                                       <<dep << " " << row << " " << col<< "\n":
     3451232 \Rightarrow 57
     3451200 \Rightarrow 40
     3 4 5 2
                 59
                      \Rightarrow 2 3 4
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."