

Assignment 02

Data Structures CS218

Note: The submission deadline is 15th April 2020. In case of plagiarism in any question, straight zero marks will be assigned for the complete assignment.

Total Points: 50

Question 01:

Given below is a 2D matrix, find the location of cells for the hidden word "TOUGH".

R\C	0	1	2	3
0	T	Q	P	V
1	S	O	J	K
2	R	U	G	D
3	A	C	B	H

Hint: Make use of backtracking technique

Your answer should be (0,0), (1,1), (2,1),(2,2), (3,3)

Question 02:

Implement Queue data structure using circular array.

Question 03:

In an operating system, the execution of program follows Queue data structure. The table below shows the order of execution and their execution time.

Google Chrome	VScode	VLC	MS-Word	PDF	Jupyter	Pycharm
10s	5s	6s	4s	8s	10s	10s

Assumptions:

1. Assume that the queue size is fixed that is 7.
2. Two seconds are assigned to each application in one trip
3. As Queue follows First in First out FIFO structure, therefore the Google Chrome will be executed first.







To do:

- a. For how many extra seconds MS-Word has to wait in order to be executed.

- b. As the size of queue is fixed i.e., 7 some other applications such as virtual box and avast are in waiting list. The execution time for virtual box is 4s and for avast is 8s. For how many seconds virtual box and avast has to wait in order to be the part of execution list.

Question 4:

You must have an idea of maze game. The target is to basically reach the goal starting from some initial point. Your task for this question is to deduce a pathway which can lead to the final destination. For example, a case is shown below. Here wherever enemies are present, it means this is a barrier and thereby you cannot move forward from that point (i.e. it's a dead end).

start			
			
			
			goal

Thus the solution in this case is $\{(0,0), (0,1), (0,2), (1,2), (2,2), (3,2), (3,3)\}$. There will always be just one solution to the problem that means there will never be more than one path present.

The input maze may contain only one-path and you need to implement a recursive path finding approach that enumerate all cell of the array that are on the path in order of traverse from start to goal. All you need to develop a recursive routine for finding path, the search is only allowed to follow {Left, Right, Up and Down} from any location. The output for this problem is complete path from starting to end location. The validation of input cases required as start and end location must be at the boundary of the maze. There may be one or no path for all valid input cases.

Input

The input is from the file, the first line contains two integers; n and m, representing the dimension of the maze. The maze is a 2-dimensional array of char. The next n lines contain the rows of the maze; each row contains m columns. Hence there are m integers in each line.

Output

The output file contains coordinates of each point starting from the start to end.

Input file	Output file
4 5	0 0
s - - - -	0 1
* * * * -	0 2
* * * * -	0 3
* * * * e	0 4
	1 4
	2 4
	3 4

Question 05:

You are given an array of numbers, there can be n numbers where n is between ($1 \leq n \leq 30000$). Each number is between ($1 \leq a_i \leq 10^9$). You are also given “ t ” queries of the form $qt(i, j, k)$ where ($0 \leq i \leq j \leq n$) for each query you need to return the number of elements greater than k in the sub-sequence a_i, a_{i+1}, \dots, a_j you need to process all t queries in the same fashion.

Input

The first line of the input file contains the size of the array “ n ”. The next n lines contain n integers for the array. The next line contains the number of the queries that you need to process. Each next line contains three integers representing left and right indexes of arrays and an integer k for which you need to check how many numbers within those indexes are greater than or equal to k .

Output

The output contains the result of the queries per line.

Input file	Output file
10	3
12	3
7	1
13	1
22	5
34	
19	
102	
77	
23	
10	
5	
0 3 7	
0 9 30	
6 7 100	
2 5 30	
3 8 20	

