Birla Institute of Technology and Science, Pilani

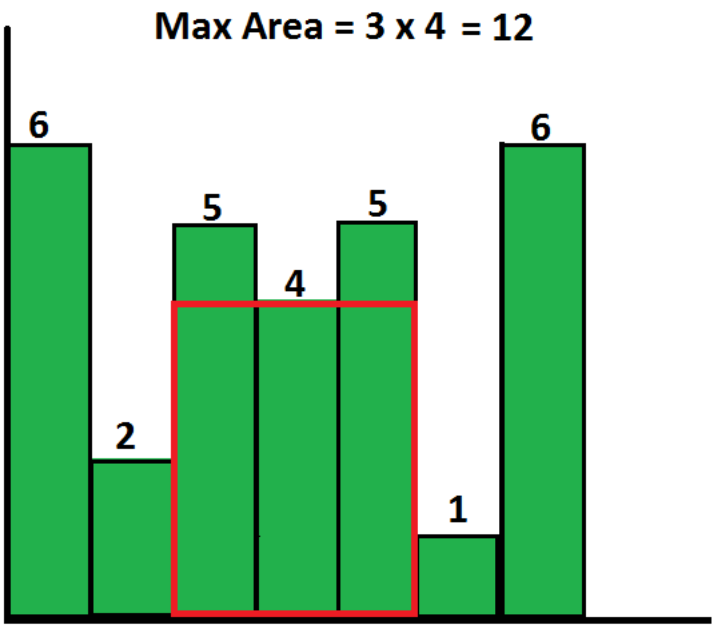
Second Semester 2017-2018, DSA (CS F211)

Lab Assignment #6

1. Raju is a painter and he has to paint N consecutive buildings whose heights are given by h1,h2,h3,……hN. He wants to estimate the time it takes to paint those n buildings, which he feels depends on the largest rectangular area formed by a continuous segment of buildings(Buildings from 1-5 or 4-7 etc…). For simplicity, assume that all ba have same width and the width is 1 unit. Can you help Raju out by finding the largest rectangular area possible formed by a continuous segment of buildings?
   Input: N

   N space separated integers denoting heights of buildings.

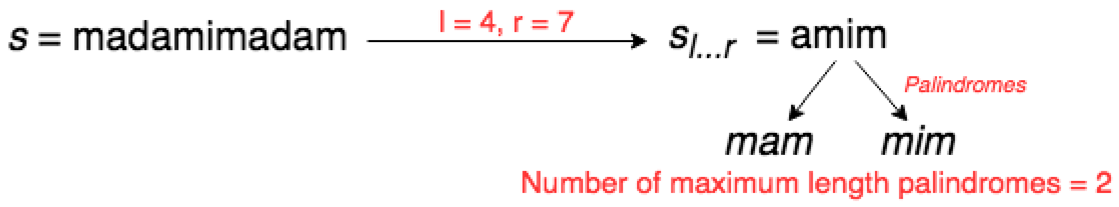
   Output: Largest rectangular area possible.
   Sample Input:
   7
   6 2 5 4 5 1 6
   Sample Output: 12
   Explanation:

   Constrains: 1<=n<=10000000; 0<=h_i<=1000000000.
   Time Limit: 2 seconds.
   Note: You have to generate random numbers denoting height for n = 10000000 and show that your code passes time limit of 2 seconds.

2. John provides a string consisting of lowercase English letters. Every day, for days, he would select two integers l and r, take the substring $S_{l...r}$ (the substring of S from index l to index r), and ask the following question: Consider all the palindromes that can be constructed from

some of the letters from $S_{l...r}$. You can reorder the letters as you need. Some of these palindromes have the maximum length among all these palindromes. How many maximum-length palindromes are there?
For example, if S = madamimadam, l = 4 and r = 7, then we have,



Your job is to answer all the queries. Since the answers can be very large, you are only required to find the answer modulo $10^9 + 7$.

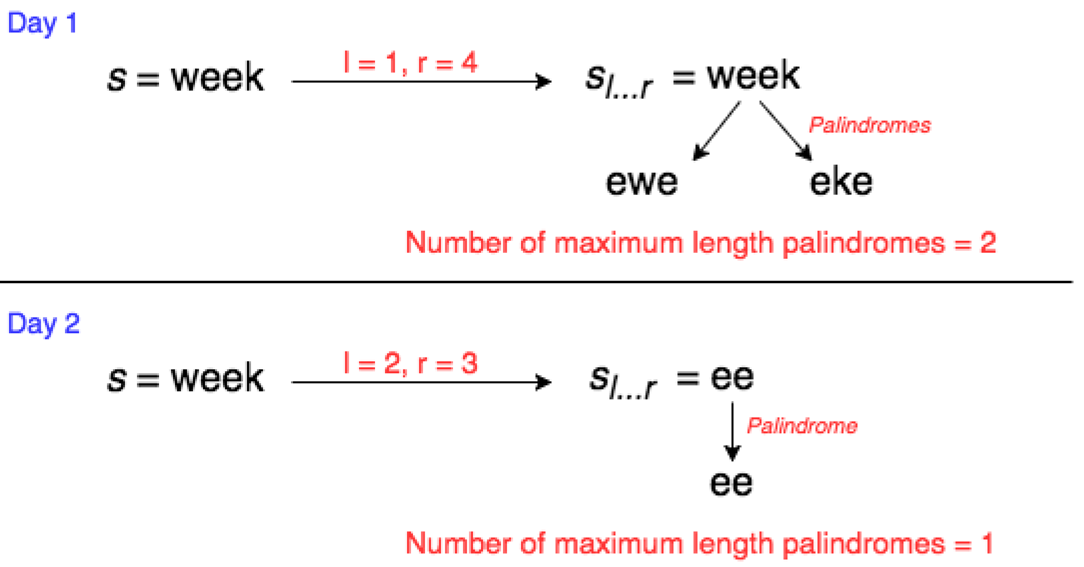
Input:
The first line contains the string s.
The second line contains a single integer q.
The ith of the next lines contains two space-separated integers l_i, r_i denoting the l and r values John selected on the ith day.

Output: Return the number of maximum-length palindromes modulo $10^9 + 7$.

Sample Input:
week
2
1 4
2 3

Sample Output:
2
1

Explanation:

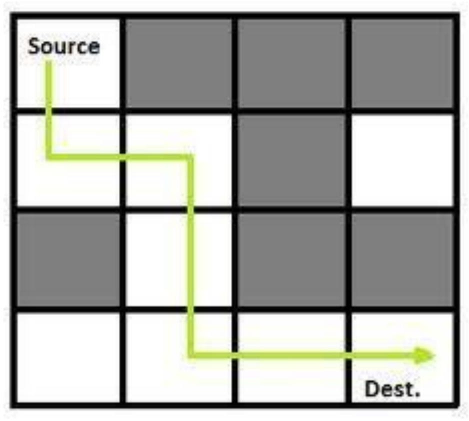


Constraints:
1 <= Length of S <= 10^5
1 <= q <= 10^5
1 <= l_i <= r_i <= Length of S

Note:

Generate a random string of length 100000 and also 100000 queries which follow the constraint that 1 <= l_i <= r_i <= Length of S and prove that your code passes time limit of 2 seconds.

3. You have a NxN maze in the form of a matrix. This maze will have some cells as open and some cells will be closed (obstacles).



A Rat can move one cell at a time in one of the four directions with following priority:

First it tries to move to one cell right, if it can't move right because of obstacle or matrix bound, it tries to go Down, if it can't move Down also, it tries to move Left, lastly if it can't move in any of the other directions it tries to move Up. The rat tries to exhaust all these possibilities before backtracking to a path it has come from. So the priority of directional movement is:

Right > Down > Left > Up

Initially, the Rat is at the top left position *i.e.* at matrix[0][0]

There's a way out at the bottom right which the Rat has to reach *i.e.* position matrix[N-1][N-1].

The rat starts to move one cell at a time from the beginning and its goal is to reach the bottom right position. If the rat finds a dead-end while trying to move and it has exhausted all its movement possibilities, it tries to backtrack upto the last position where it has some options open to move according to its priority order.

Write a program that uses your own stack for two different 10x10 matrices given below that will print the complete movement of the rat in reaching the exit at bottom right, or will print "No path exists to exit" and show its movements in the matrix after trying all the movements. If there is a path, the program should print how many times the rat has moved in a cell during the journey. In the matrix below, elements with 1 show path open and elements 0 mean it is closed.

The output should also be in the form of a matrix that shows all the paths the rat has tried and the final taken i.e., if a cell is in the path taken by the rat, you can increment it by one each time it passes through the block.

Note: <u>Do NOT use recursive calls in your program</u>. Create and use explicit stack(s). Apply backtracking method by using the stack(s).

The matrices are:

```
1 1 1 1 1 1 1 0 0 0
1 0 1 1 0 0 0 0 0 0
1 1 0 0 1 1 0 0 0 0
1 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 0 0
1 1 1 1 1 1 1 1 1 1
0 0 0 0 1 1 1 0 0 0
0 0 0 0 1 0 0 1 1 1
0 0 0 0 1 1 1 1 0 1
0 0 0 1 1 1 1 1 0 1
```

AND

```
1 1 1 1 1 1 1 0 0 0
1 0 1 1 0 0 0 0 0 0
1 1 0 0 1 1 0 0 0 0
1 0 1 1 1 1 1 1 1 1

1 1 1 1 1 1 0 0 0 0
1 1 1 1 1 1 1 1 1 1
0 0 0 0 1 1 1 0 0 0
0 0 0 0 1 0 0 0 1 1
0 0 0 0 1 1 1 1 0 1
0 0 0 1 1 1 1 1 0 1
```

4. Write a program to convert a given correct infix expression into postfix expression, print the postfix expression, evaluate the postfix using stack and print the result. You can assume that only the following operators are allowed: **+ -* / ( )**

5. A) Write a menu driven C program to implement a **queue** with 2 **stacks**. Your queue should have an enqueue() and a dequeue() function and it should be "first in first out" (FIFO). Your program must support the following operations.

   a. Insert an element (enqueue)

   b. Delete an element (dequeue) – print the element

   c. Print the elements of the stack (1. stack1, 2. stack2)

   d. Exit

   B) Write a menu driven C program to implement a **stack** with 2 **queues**. Your stack should have a push and a pop function and it should be "last in first out" (LIFO). Your program must support the following operations.

   a. Insert an element (Push)

   b. Delete an element (Pop) – print the element

   c. Print the elements of the queue (1. queue1, 2. queue2)

   d. Exit

6. Create a program to create and handle a Doubly-Linked List. The program has to perform the following functions:
   a. Insert an element in the beginning of the list.
   b. Insert an element at the end of the list.
   c. Remove the element at the i^{th} place.
   d. Remove the first element.
   e. Remove the last element.
   f. Print the list.

   The program should be running in an infinite loop and has to perform the tasks according to user’s choice. In every iteration, the program should ask the user for the task to be performed and take appropriate inputs. Proper functions are to be implemented for each task and no global declarations are allowed except for the data structure itself.

7. Create a program to take user input and create a linked list and then reverse the list. In the end, print the linked list. While reversing, the same linked list should be used. A new list should not be created. Use proper functions like insertAtBeg() and reverse() to do so. No global declarations are allowed except for the Linked List. Create large Linked lists using random numbers for testing.
   a. Implement reverse1() using iterative approach.
   b. Implement reverse2() using recursive approach.

8. Write a program which takes the given linked list and swaps the elements pairwise. In the end, print the linked list. While doing so, the same linked list should be used. A new list

should not be created. Use proper functions. No global declarations are allowed except for the Linked List. If there are odd number of elements, leave the last one as it is. Create large Linked lists using random numbers for testing.
Example: 1->2->3->4->NULL gives 2->1->4->3->NULL.

9. Cycle detection: Given a singly-linked list, it has to be detected whether there is a cycle in it or not. The linked list may be very long. Generate linked lists randomly (random length and content) based on input. 0 – acyclic; 1 – cyclic. Your function must take the head pointer of the list and detect a cycle if there is any. Output 1 if there is a cycle or else output 0.
   Input – 0 or 1
   Output – Same integer as input
   Sample Input: 0
   Sample Output: 0
10. Shikhar is going through DSA lab evaluation and people are to be seated in a circle to code. Computers are labelled from 1 to n continuously in the circle where n people are participating. The evaluation happens in a different fashion: it starts at 1st computer and the instructor goes around the circle from computer 1 towards n skipping m-1 people at a time. The students leave after their evaluation. Shikhar knows that the instructor is in a hurry and will give full marks to the last person to be evaluated. Since he is very cunning, guess where would he sit to get this full marks. Use circular linked lists to solve this question.
    Input: n m
    Output: number of the computer on which Shikhar sits.
    Sample Input 1: 4 2
    Sample Output 1: 3
    Sample Input 2: 4 2
    Sample Output 2: 4
    Sample Input 3: 5 3
    Sample Output 3: 4