

Problems based on Stacks

Assignment Solutions



Q1. You are given an array of n digits. Your task is to generate a k-digit number such that it is the maximum possible number that can be generated from the given digits. Make sure that the relative ordering of the digits is not changed.

Note the value of k can be large so the number should be formed in the form of a string.

Input

n=4

arr=[1, 2, 3, 4]

k=3

Output

234

Input

n=4

arr=[1, 2, 3, 4]

k=2

Output

34

Ans: Code <https://pastebin.com/sQtXa6Su>

Output

```
PS D:\Coding\PW> cd "d:\Coding\PW\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
4
3 2 4 1
1 3 3 -1
```

Explanation

1. To make the largest number possible, at each position starting from the left, we need to place the maximum digit possible. To do this we can take help from a stack. It will store our answer.
2. Loop through all the digits available-
 - a. For each digit compare it with the top value of the stack, if there is any. If the value at the top of the stack is smaller than the current value and we can afford to erase that digit from our answer then remove it from the stack. We will continue to do this until one of the conditions becomes false.
 - b. Add the current digit to the stack.
3. Form the answer string from the values of the stack.

Q2. You are given two non-empty linked lists representing two non-negative integers. The most significant digit comes first and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

The first line of input contains n and m, the size of the 2 linked lists.

The second line of input contains n integers, the values stored in the first linked list.

The third line of input contains m integers, the values stored in the second linked list.

Input

3 4

123

123 4

Output

1357

Ans: Code <https://pastebin.com/SkvdZJTv>

Output

```
PS D:\Coding\PW> cd "d:\Coding\PW\" ; if ($?) { g++ main.cpp -o main } ; if (?) {  
3 4  
1 2 3  
1 2 3 4  
1357
```

Explanation

1. Since the addition is done from right to left, we need to read the linked lists in reverse order. We can use stacks for this.
2. Store the values of the linked lists in 2 stacks.
3. One by one take out the values from the stack, and add their sum at the head of the linked list representing the sum.
4. At the end if there is any carry left, add it to the sum linked list too.

Q3. Given a vector, print the index of the Next Smaller Element for every element.

The Next Smaller Element for an element x is the first smaller element on the right side of x in the vector.

Elements for which no smaller element exists, consider the next smaller element as -1.

The first line of input contains the size of the vector.

The second line of input contains the elements of the vector.

Input

```
4  
3 2 4 1
```

Output

```
1 3 3 -1
```

Explanation

For 3, the next smaller element is 2(index = 1).

For 2, the next smaller element is 1(index = 3).

For 4, the next smaller element is 1(index = 3).

For 1, there is no value smaller than it on its right side

Ans: Code <https://pastebin.com/YxvupNSS>

Output

```
PS D:\Coding\PW> cd "d:\Coding\PW\" ; if ($?) { g++ main.cpp -o main } ; if (?) { .\main }  
5  
1 10 3 2 4  
2 4
```

Explanation

The code for the next smaller element is similar to the code for the next greater integer.

1. Traverse the vector from right to left.
2. For each element-
 - a. Remove the indices from the stack at which the value stored is greater than or equal to the current element.
 - b. If the stack is not empty, the value of the top of the stack is the index of the next smaller element.
 - c. Push the current element into the stack.

Q4. You are given an array of n integers representing the heights of the buildings in an area. Mario can jump from one building of height arr[i] to another of height arr[j], i <= j, if the height of the second building is strictly greater than the height of the first building i.e. arr[j] > arr[i]. Each day mario makes a jump. Today he is a little busy so he wants to keep the length of jump(j - i) as small as possible. Help him find the location from where he must make his jump and to where.

Assume there always exists a valid answer. If there are multiple answers, print the answer with the minimum index(0-based indexing).

The first line of input contains the value of n.

The second line of input contains n integers, the heights of the buildings.

Input

5

1 10 3 2 4

Output

0 1

Explanation

There are 6 jumps possible for Mario to make.

1→10, distance = 1 - 0 = 1

1→3, distance = 2 - 0 = 2

1→2, distance = 3 - 0 = 3

1→4, distance = 4 - 0 = 4

3→4, distance = 4 - 2 = 2

2→4, distance = 4 - 3 = 1

Out of them 1→10 and 2→4 have the minimum distance.

Here 1 is located at index 0 and 2 is located at index 3. Since 1 has a lower index, we will print the indices of 1 and 10 i.e. 0 and 1

Ans: Code <https://pastebin.com/9yq034R0>

Output

```
PS D:\Coding\PW> cd "d:\Coding\PW\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
4
1 2 3 4
2
34
```

Explanation

1. To find the pair of increasing values of the array at minimum distance, we can find the next greater integer for each value in the array.
2. Out of all the possible answers, we can choose the ones with the minimum distance between them.