**STRINGS ASSIGNMENT**

Q1. String Palindrome

**Given a string, determine if it is a palindrome, considering only alphanumeric characters.**

**Palindrome**

A palindrome is a word, number, phrase, or other sequences of characters which read the same backwards and forwards.

**Example:**

If the input string happens to be, "malayalam" then as we see that this word can be read the same as forward and backwards, it is said to be a valid palindrome.

The expected output for this example will print, 'true'.

**From that being said, you are required to return a boolean value from the function that has been asked to implement.**

**Input Format:**

The first and only line of input contains a string without any leading and trailing spaces. All the characters in the string would be in lower case.

**Output Format:**

The only line of output prints either 'true' or 'false'.

**Note:**

You are not required to print anything. It has already been taken care of.

**Constraints:**

0 <= N <= 10^6

Where N is the length of the input string.

Time Limit: 1 second

**Sample Input 1 :**

abcdcba

**Sample Output 1 :**

true

**Sample Input 2:**

coding

**Sample Output 2:**

false

Q2. Replace Character

**Given an input string S and two characters c1 and c2, you need to replace every occurrence of character c1 with character c2 in the given string.**

**Input Format :**

Line 1 : Input String S

Line 2 : Character c1 and c2 (separated by space)

**Output Format :**

Updated string

**Note :**

You don't need to output anything. Just implement the given function.

**Constraints :**

**1 <= Length of String S <= 10^6**

**Sample Input :**

abacd

a x

**Sample Output :**

xbxcd

Q3. Trim Spaces

**Given an input string S that contains multiple words, you need to remove all the spaces present in the input string.**

**There can be multiple spaces present after any word.**

**Input Format :**

String S

**Output Format :**

Updated string

**Constraints :**

**1 <= Length of string S <= 10^6**

**Sample Input :**

abc def g hi

**Sample Output :**

abcdefghi

Q4. Change the case

**Write a function that converts a given string to lower case.**

**Assume, given string contains uppercase alphabets only. But input can contain multiple words.**

**Sample Input 1 :**

ABCDE

**Sample Output 1 :**

abcde

**Sample Input 2 :**

ABC DEF

**Sample Output 2 :**

abc def

Q5. Find All Substrings

**For a given input string(str), write a function to print all the possible substrings.**

**Substring**

A substring is a contiguous sequence of characters within a string.

Example: "cod" is a substring of "coding". Whereas, "cdng" is not as the characters taken are not contiguous

**Input Format:**

The first and only line of input contains a string without any leading and trailing spaces. All the characters in the string would be in lower case.

**Output Format:**

Print the total number of substrings possible, where every substring is printed on a single line and hence the total number of output lines will be equal to the total number of substrings.

**Note:**

The order in which the substrings are printed, does not matter.

**Constraints:**

0 <= N <= 10^6

Where N is the length of the input string.

Time Limit: 1 second

**Sample Input 1:**

abc

**Sample Output 1:**

a

ab

abc

b

bc

c

**Sample Input 2:**

co

**Sample Output 2:**

c

co

o

Q6. Rotate string(important concept to clear doubts in strings)

**Given a string and an integer n, shift and rotate the given string by n characters in right.**

**Note : Assume n is always less than input string length.**

**Input format :**

Line 1 : Input string

Line 2 : Integer n

**Output Format :**

Rotated string

**Sample Input :**

CodingNinjas

3

**Sample Output :**

jasCodingNin

Q7. Remove character

**For a given a string(str) and a character X, write a function to remove all the occurrences of X from the given string.**

**The input string will remain unchanged if the given character(X) doesn't exist in the input string.**

**Input Format:**

The first line of input contains a string without any leading and trailing spaces.

The second line of input contains a character(X) without any leading and trailing spaces.

**Output Format:**

The only line of output prints the updated string.

**Note:**

You are not required to print anything explicitly. It has already been taken care of.

**Constraints:**

0 <= N <= 10^6

Where N is the length of the input string.

Time Limit: 1 second

**Sample Input 1:**

aabccbaa

a

**Sample Output 1:**

bccb

**Sample Input 2:**

xxyyzxx

y

**Sample Output 2:**

xxzxx

Q8. Remove Consecutive Duplicates

**For a given string(str), remove all the consecutive duplicate characters.**

**Example:**

Input String: "aaaa"

Expected Output: "a"

Input String: "aabbbcc"

Expected Output: "abc"

**Input Format:**

The first and only line of input contains a string without any leading and trailing spaces. All the characters in the string would be in lower case.

**Output Format:**

The only line of output prints the updated string.

**Note:**

You are not required to print anything. It has already been taken care of.

**Constraints:**

0 <= N <= 10^6

Where N is the length of the input string.

Time Limit: 1 second

**Sample Input 1:**

aabccbaa

**Sample Output 1:**

abcba

**Sample Input 2:**

xxyyzxx

**Sample Output 2:**

xyzx

Q9. Highest Occuring Character

**For a given a string(str), find and return the highest occurring character.**

**Example:**

Input String: "abcdeapapqarr"

Expected Output: 'a'

Since 'a' has appeared four times in the string which happens to be the highest frequency character, the answer would be 'a'.

**If there are two characters in the input string with the same frequency, return the character which comes first.**

**Consider:**

Assume all the characters in the given string to be in lowercase always.

**Input Format:**

The first and only line of input contains a string without any leading and trailing spaces.

**Output Format:**

The only line of output prints the updated string.

**Note:**

You are not required to print anything explicitly. It has already been taken care of.

**Constraints:**

0 <= N <= 10^6

Where N is the length of the input string.

Time Limit: 1 second

**Sample Input 1:**

abdefgbabfba

**Sample Output 1:**

b

**Sample Input 2:**

xy

**Sample Output 2:**

x

**Q10. Merge two arrays**

**You have been given two sorted arrays(ARR1 and ARR2) of size N and M respectively, merge them into a third array such that the third array is also sorted.**

**Input Format :**

First line contains an integer 'N' representing the size of the first array/list.

Second line contains 'N' single space separated integers representing the elements of the first array.

Third line contains an integer 'M' representing the size of the second array.

Fourth line contains 'M' single space separated integers representing the elements of the second array.

**Output Format :**

Print the sorted array(of size N + M) in a single row, separated by a single space.

Output to be printed in a separate line.

**Constraints :**

0 <= N <= 10^5

0 <= M <= 10^5

**Sample Input 1 :**

1

5

1 3 4 7 11

4

2 4 6 13

**Sample Output 1 :**

1 2 3 4 4 6 7 11 13

**Sample Input 2 :**

3

10 100 500

7

4 7 9 25 30 300 450

**Sample Output 2:**

4 7 9 10 25 30 100 300 450 500

**Sample Input 3:**

4

7 45 89 90

1

70

**Sample Output 3 :**

7 45 70 89 90

**Sample Input 4:**

2

0 1

1

1

**Sample Output 4:**

0 1 1