Strings

1. **Predict the output**

**Send Feedback**

What will be the output of following code?

s = “abcd”

print(s[-2])

1. c
2. **Predict the output**

**Send Feedback**

What will the output of following code?

s="abcd"

s[0]='c'

print(s)

1. Error
2. **Predict the output**

**Send Feedback**

What will be the output of following code?

s =”abcd”

a = “abcd”

if id(s) == id(a):

print(“They are same”)

else:

print(“They are not same”)

1. They are same
2. **Predict the output**

**Send Feedback**

What will the output of the following code?

s = "abcd"

b = s + "ef"

print(s)

1. abcd
2. **Predict the output**

**Send Feedback**

What will the output of following code?

s = "abcd"

b = s + 2

print(b)

1. Error
2. **Predict the output**

**Send Feedback**

What will be the output of the following code?

s = "abcdef"

print (s[2:])

1. cdef
2. **Predict the output**

**Send Feedback**

What will be the output of the following code?

s = "abcdef"

print (s[4:2:-1])

1. ed
2. **Predict the output**

**Send Feedback**

What will be the output of the following code?

a = "abcdef" == "abcd"

print(a)

1. false
2. **Predict the output**

**Send Feedback**

What will be output of the following code?

a = "abcdef" >= "abcd"

print(a)

1. True
2. **Predict the output**

**Send Feedback**

What will be output of the following code?

a = "abce" >= "abcdef"

print(a)

1. True

Assignment

1. **Check Permutation**

**Send Feedback**

#### Given two strings, S and T, check if they are permutations of each other. Return true or false.

#### Permutation means - length of both the strings should same and should contain same set of characters. Order of characters doesn't matter.

##### Note : Input strings contain only lowercase english alphabets.

##### Input format :

Line 1 : String 1

Line 2 : String 2

##### Output format :

'true' or 'false'

##### Constraints :

0 <= |S| <= 10^7

0 <= |T| <= 10^7

where |S| represents the length of string, S.

##### Sample Input 1 :

abcde

baedc

##### Sample Output 1 :

true

##### Sample Input 2 :

abc

cbd

##### Sample Output 2 :

false

1. **Remove Consecutive Duplicates**

**Send Feedback**

#### Given a string, S, remove all the consecutive duplicates that are present in the given string. That means, if 'aaa' is present in the string then it should become 'a' in the output string.

##### Input format :

String S

##### Output format :

Modified string

##### Constraints :

0 <= |S| <= 10^7

where |S| represents the length of string, S.

##### Sample Input 1:

aabccbaa

##### Sample Output 1:

abcba

##### Sample Input 2:

xxyyzxx

##### Sample Output 2:

xyzx

1. **Reverse Each Word**

**Send Feedback**

#### Given a string S, reverse each word of a string individually. For eg. if a string is "abc def", reversed string should be "cba fed".

##### Input Format :

String S

##### Output Format :

Modified string

##### Constraints :

0 <= |S| <= 10^7

where |S| represents the length of string, S.

##### Sample Input 1:

Welcome to Coding Ninjas

##### Sample Output 1:

emocleW ot gnidoC sajniN

##### Sample Input 2:

Give proper names to variables and functions

##### Sample Output 2:

eviG reporp seman ot selbairav dna snoitcnuf

1. **Remove character**

**Send Feedback**

#### Given a string and a character x. Write a function to remove all occurrences of x character from the given string.

##### Leave the string as it is, if the given character is not present in the string.

##### Input Format :

Line 1 : String S

Line 2 : Character c

##### Output Format :

Modified string

##### Constraints :

0 <= |S| <= 10^7

where |S| represents the length of string, S.

##### Sample Input 1:

welcome to coding ninjas

o

##### Sample Output 1:

welcme t cding ninjas

##### Sample Input 2:

Think of edge cases before submitting solutions

x

##### Sample Output 2:

Think of edge cases before submitting solutions

1. **Highest Occurring Character**

**Send Feedback**

#### Given a string, S, find and return the highest occurring character present in the given string.

##### If there are 2 characters in the input string with same frequency, return the character which comes first.

##### Note : Assume all the characters in the given string are lowercase.

##### Input format :

String S

##### Output format :

Highest occurring character

##### Constraints :

0 <= |S| <= 10^7

where |S| represents the length of string, S.

##### Sample Input 1:

abdefgbabfba

##### Sample Output 1:

b

##### Sample Input 2:

xy

##### Sample Output 2:

x

**Compress the String**

**Send Feedback**

#### Write a program to do basic string compression. For a character which is consecutively repeated more than once, replace consecutive duplicate occurrences with the count of repetitions.

#### Exmple:

If a String has 'x' repeated 5 times, replace this "xxxxx" with "x5".

The string is compressed only when the repeated character count is more than 1.

##### Note :

Consecutive count of every character in the input string is less than equal to 9.

##### Input Format :

The first and the only line of input contains a string(no spaces in between).

##### Output Format :

The only line of output print the compressed string.

##### Note:

Return the compressed string and hence, no need to print.

##### Constraints :

0 <= |S| <= 10^7

Where |S| represents the length of string, S.

Time Limit: 1sec

##### Sample Input 1 :

aaabbccdsa

##### Sample Output 1 :

a3b2c2dsa

##### Sample Input 2 :

aaabbcddeeeee

##### Sample Output 2 :

a3b2cd2e5

1. **Compress the String**

**Send Feedback**

#### Write a program to do basic string compression. For a character which is consecutively repeated more than once, replace consecutive duplicate occurrences with the count of repetitions.

#### Exmple:

If a String has 'x' repeated 5 times, replace this "xxxxx" with "x5".

The string is compressed only when the repeated character count is more than 1.

##### Note :

Consecutive count of every character in the input string is less than equal to 9.

##### Input Format :

The first and the only line of input contains a string(no spaces in between).

##### Output Format :

The only line of output print the compressed string.

##### Note:

Return the compressed string and hence, no need to print.

##### Constraints :

0 <= |S| <= 10^7

Where |S| represents the length of string, S.

Time Limit: 1sec

##### Sample Input 1 :

aaabbccdsa

##### Sample Output 1 :

a3b2c2dsa

##### Sample Input 2 :

aaabbcddeeeee

##### Sample Output 2 :

a3b2cd2e5