| | |
|---|---|
| **Started on** | Thursday, 18 April 2024, 6:35 PM |
| **State** | Finished |
| **Completed on** | Saturday, 4 May 2024, 11:11 AM |
| **Time taken** | 15 days 16 hours |
| **Overdue** | 13 days 16 hours |
| **Marks** | 10.00/10.00 |
| **Grade** | **100.00** out of 100.00 |

Question **1**

Correct

Mark 1.00 out of 1.00

Write a python program to count all letters, digits, and special symbols respectively from a given string

**For example:**

| Input | Result |
|---|---|
| rec@123 | 3<br>3<br>1 |

**Answer:** (penalty regime: 0 %)

```
x = input()
d = 0
l = 0
s = 0
for char in x:
    if char.isdigit():
        d +=1
    elif char.isalpha():
        l += 1
    else:
        s += 1


print (l)
print (d)
print (s)

```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | rec@123 | 3<br>3<br>1 | 3<br>3<br>1 | ✓ |
| ✓ | P@#yn26at^&i5ve | 8<br>3<br>4 | 8<br>3<br>4 | ✓ |
| ✓ | abc@12& | 3<br>2<br>2 | 3<br>2<br>2 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

**Reverse** a string **without affecting special characters**

Given a string **S**, containing special characters and all the alphabets, reverse the string without affecting the positions of the special characters.

**Input:**

A&B

**Output:**

B&A

**Explanation**: As we ignore '&' and

As we ignore '&' and then reverse, so answer is "B&A".

**For example:**

| Input | Result |
|---|---|
| A&x# | x&A# |

**Answer:** (penalty regime: 0 %)

```
x=input("")
c=['@','#','$','^','&']
n=[]
for i in range(len(x)):
    if x[i] not in c:
        n.append(x[i])
n.reverse()
for j in x:
    if j in c:
        idx=x.index(j)
        n.insert(idx,j)
reverse="".join(n)
print(reverse)
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | A&B | B&A | B&A | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Write a program that takes as input a string (sentence), and returns its second word in uppercase.

For example:

If input is “Wipro Technologies Bangalore” the function should return “TECHNOLOGIES”

If input is “Hello World” the function should return “WORLD”

If input is “Hello” the program should return “LESS”

NOTE 1: If input is a sentence with less than 2 words, the program should return the word “LESS”.

NOTE 2: The result should have no leading or trailing spaces.

**For example:**

| Input | Result |
|---|---|
| Wipro Technologies Bangalore | TECHNOLOGIES |
| Hello World | WORLD |
| Hello | LESS |

**Answer:** (penalty regime: 0 %)

```
a = input()
b = a.upper()
c = b.split()
for i in c:
    if len(c)>=2:
        print(c[1])
        break
    else:
        print("LESS")

```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | Wipro Technologies Bangalore | TECHNOLOGIES | TECHNOLOGIES | ✓ |
| ✓ | Hello World | WORLD | WORLD | ✓ |
| ✓ | Hello | LESS | LESS | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

In this exercise, you will create a program that reads words from the user until the user enters a blank line. After the user enters a blank line your program should display each word entered by the user exactly once. The words should be displayed in the same order that they were first entered. For example, if the user enters:

first

second

first

third

second

then your program should display:

first

second

third

**Answer:** (penalty regime: 0 %)

```
x=[]
for i in range(5):
    x.append(input())
y=[]
for i in x:
    if i not in y:
        y.append(i)
for i in y:
    print(i)
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | first<br>second<br>first<br>third<br>second | first<br>second<br>third | first<br>second<br>third | ✓ |
| ✓ | rec<br>cse<br>it<br>rec<br>cse | rec<br>cse<br>it | rec<br>cse<br>it | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

Given two Strings s1 and s2, remove all the characters from s1 which is present in s2.

**Constraints**

1<= string length <= 200

**Sample Input 1**

experience
enc

**Sample Output 1**

xpri

**Answer:** (penalty regime: 0 %)

```
x_str = input()
x2 = input()
result = ''.join([char for char in x_str if char not in x2])
print(result)
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | experience<br>enc | xpri | xpri | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **6**

Correct

Mark 1.00 out of 1.00

Assume that the given string has enough memory.

Don't use any extra space(IN-PLACE)

**Sample Input 1**

a2b4c6

**Sample Output 1**

aabbbbccccc

**Answer:** (penalty regime: 0 %)

```
x=input("")
r=""
i=0
while i<len(x):
    c=x[i]
    cs=""
    i+=1
    while i <len(x) and x[i].isdigit():
        cs+=x[i]
        i+=1
    count=int(cs)
    r+=c*count
print(r)
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | a2b4c6 | aabbbbccccc | aabbbbccccc | ✓ |
| ✓ | a12b3d4 | aaaaaaaaaaabbbdddd | aaaaaaaaaaabbbdddd | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **7**

Correct

Mark 1.00 out of 1.00

String should contain only the words are not palindrome.

**Sample Input 1**

Malayalam is my mother tongue

**Sample Output 1**

is my mother tongue

**Answer:** (penalty regime: 0 %)

```
s=input()
low=s.lower()
x=low.split(' ')
li=[]
for i in x:
    li.append(i)
    if(str(i)==str(i[::-1])):
        li.remove(i)
z=' '.join(li)
print(z)
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | Malayalam is my mother tongue | is my mother tongue | is my mother tongue | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **8**

Correct

Mark 1.00 out of 1.00

Write a program to check if two strings are balanced. For example, strings s1 and s2 are balanced if all the characters in the s1 are present in s2. The character's position doesn't matter. If balanced display as "true" ,otherwise "false".

**For example:**

| Input | Result |
|---|---|
| Yn<br>PYnative | True |

**Answer:** (penalty regime: 0 %)

```
x = input()
y = input()
if x in y:
    print(True)
else:
    print(False)
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | Yn<br>PYnative | True | True | ✓ |
| ✓ | Ynf<br>PYnative | False | False | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **9**

Correct

Mark 1.00 out of 1.00

Two string values S1, S2 are passed as the input. The program must print first N characters present in S1 which are also present in S2.

**Input Format:**

The first line contains S1.
The second line contains S2.
The third line contains N.

**Output Format:**

The first line contains the N characters present in S1 which are also present in S2.

**Boundary Conditions:**

2 <= N <= 10
2 <= Length of S1, S2 <= 1000

**Example Input/Output 1:**

Input:

abcbde
cdefghbb
3

Output:

bcd

**Note:**

b occurs twice in common but must be printed only once.

**Answer:** (penalty regime: 0 %)

```
S1 = input().strip()
S2 = input().strip()
N = int(input())

common_chars = ""

for char in S1:
    if char in S2 and char not in common_chars:
        common_chars += char
        if len(common_chars) == N:
            break
print(common_chars)

```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | abcbde<br>cdefghbb<br>3 | bcd | bcd | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **10**

Correct

Mark 1.00 out of 1.00

Given a string S which is of the format USERNAME@DOMAIN.EXTENSION, the program must print the EXTENSION, DOMAIN, USERNAME in the reverse order.

**Input Format:**

The first line contains S.

**Output Format:**

The first line contains EXTENSION.
The second line contains DOMAIN.
The third line contains USERNAME.

**Boundary Condition:**

1 <= Length of S <= 100

**Example Input/Output 1:**

Input:

abcd@gmail.com

Output:

com
gmail
abcd

**Answer:** (penalty regime: 0 %)

```
S = input().strip()
a = S.index('@')
d = S.index('.')
extension = S[d + 1:]
domain = S[a + 1:d]
username = S[:a]
print(extension)
print(domain)
print(username)
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | abcd@gmail.com | com<br>gmail<br>abcd | com<br>gmail<br>abcd | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◄ Week5_MCQ

Jump to...

List ►