T2_UNIT_4_QB_Solution _VHA

November 4, 2025

1 T2 UNIT-4 QB SOLUTION VHA

Prepared by: Prof. Vishal Acharya

2 291Write a Python program to check if a string is palindrome or not.

[19]:

```
s=input("enter a string")
if s==s[::-1]:
    print("Pallindrome")
else:
    print("Not Palindrome")
```

```
enter a stringANANA
Pallindrome
```

3 292 Write a Python program to Find length of a string in python.

[20]:

```
s=input("enter a string")
string_length = len(s)
print("Length of the string:", string_length)
```

```
enter a stringVishal Acharya
Length of the string: 14
```

4 293 Write a Python function to find length of a string in python without using len function.

[21]:

```
s=input("enter a string")
def slen(s):
    count = 0
    for char in s:
        count += 1
    return "Length of the string:",count
print(slen(s))
```

```
enter a stringVishal Acharya
('Length of the string:', 14)
```

5 294 Write a Python function that accepts a string and calculate the number of uppercase letters and lowercase letters.

[22]:

```
def slen(s):
    u=0
    l=0
    for char in s:
        if char.isupper():
            u=u+1
        if char.islower():
            l=l+1
    print("Uppercase count:", u)
    print("Lowercase count:", l)
s=input("enter a string")
slen(s)
```

```
enter a stringVishal H AchaRya
Uppercase count: 4
Lowercase count: 10
```

6 295 Write a Python program to demonstrate the negative index in a Tuple

[23]:

```
my_tuple = eval(input("enter a tuple"))
print("Last element:", my_tuple[-1])
print("Second to last element:", my_tuple[-2])
```

```
enter a tuple(1,2,3,4)
Last element: 4
Second to last element: 3
```

7 296 Write a program to remove I'th character from string in python.

[24]:

```
s = input("enter a string")
i = int(input("enter a value of index "))
print(s[:i] + s[i+1:])
```

```
enter a stringVishal Acharya
enter a value of index 7
Vishal charya
```

8 297 Write a program to create a string made of first,middle and last character.

[27]:

```
s = input("enter a string")
print(s[0] + s[len(s)//2] + s[-1])
```

```
enter a stringVishal HAcharya
VHa
```

9 298 Write a program to find all occuences of a sub string in a given string by ignoring the case.

[28]:

```
s = input("enter a string")
sub = "b"
for i in range(len(s)):
    if s[i:i+len(sub)].lower() == sub.lower():
        print(i)
```

```
enter a stringB2,B7 and D1 are best in python
0
3
17
```

10 299 Write a program to calculate the sum and average of the digits present in a string.

[29]:

```
s=(input("enter string: "))
sum=0
c=0
for i in s:
    if i.isdigit()==True:
        sum+=int(i)
        c=c+1
print("sum",sum)
print(c)
print("avg=",sum/c)
```

```
enter string: B2,B7 and D1 are best in python
sum 10
3
avg= 3.3333333333333335
```

11 300 Write a program to reverse a given string

[30]:

```
original_string = input("enter string: ")
reversed_string = original_string[::-1]
print("Original String:", original_string)
print("Reversed String:", reversed_string)
```

```
enter string: B2,B7 and D1 are best in python
Original String: B2,B7 and D1 are best in python
Reversed String: nohtyp ni tseb era 1D dna 7B,2B
```

12 301 Write a Python program to print even length words in a string.

[31]:

```
s=input("enter string: ")
n=s.split(" ")
for i in n:
    if len(i)%2==0:
        print(i,end=" ")
```

```
enter string: B2,B7 and D1 are best in python
D1 best in python
```

13 302 Write a Python program to Uppercase Half String from the given string.

[32]:

```
s=input("enter string: ")
new_string1=s[0:len(s)//2:1].upper()
new_string2=s[len(s)//2:len(s)+1:1].lower()
print(new_string1+new_string2)
```

```
enter string: B2,B7 and D1 are best in python
B2,B7 AND D1 ARe best in python
```

14 303 Write a Python program to capitalize the first and last character of each word in a string

[33]:

```
s=input("enter a string: ")
s = s.title()
result = ""
for word in s.split():
    result += word[:-1] + word[-1].upper() + " "
print(result)
```

```
enter a string: B2,B7 and D1 are best in python
B2,B7 AnD D1 ArE BesT IN PythoN
```

15 304 Write a program to Create a string made of the middle three charactershree characters:**

[40]:

```
s=input("enter string: ")
new_string=s[(len(s)//2)-1:(len(s)//2)+2]
print(new_string)
```

```
enter string: VishalHAcharya
HAc
```

16 305 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.

[41]:

```
s1=input("enter string: ")
s2= input("enter string: ")
flag=0
for i in s1:
    if i in s2:
        continue
    else:
        flag=1
        break
if(flag ==1):
    print("not balanced")
else:
    print("balanced")
```

```
enter string: Vishal
enter string: Vishal10
balanced
```

17 306 Write a program to Split a string on hyphens

[1]:

```
input_string = input("enter a string:")
split_string = input_string.split('-')
print("Split String:", split_string)
```

```
enter a string:VISHAL-IS-BEST
Split String: ['VISHAL', 'IS', 'BEST']
```

18 307 Write a program to print maximum and minimum elements in given Tuple.

[43]:

```
my_tuple = eval(input("enter a tuple"))
max_element = max(my_tuple)
min_element = min(my_tuple)
print("Maximum element:", max_element)
print("Minimum element:", min_element)
```

```
enter a tuple(1,2,5,8,-9)
Maximum element: 8
Minimum element: -9
```

19 308 Write a Program to print even numbers from given Tuple.

[46]:

```
my_tuple = eval(input("enter a tuple"))
for i in my_tuple:
     if i%2==0:
        print("Even numbers:", i)
```

```
enter a tuple(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
Even numbers: 2
Even numbers: 4
Even numbers: 6
Even numbers: 8
Even numbers: 10
```

20 309 Write a program to print sum of even numbers and sum of odd numbers from elements given in tuple.

[47]:

```
t=eval(input())
sum_e=sum_o=0
for i in t:
    if i%2==0:
        sum_e+=i
    else:
        sum_o+=i

print("Sum of even numbers:", sum_e)
print("Sum of odd numbers:", sum_o)
```

```
(1,2,3,4,5,6,7,8,9,0)
Sum of even numbers: 20
Sum of odd numbers: 25
```

21 310 Write a Python program using function to shift the decimal digits n places to the left, wrapping the extra digits around. If shift > the number of digits of n, then reverse the string.

- Note:

Function will take two parameters:

1. The number
2. How much shift user want

Example:

Input: n=12345 shift=1

Output: Result=23451

Input: n=12345 shift=3

Output: Result=45123

Input: n=12345 shift=5

Output: Result=12345

Input: n=12345 shift=6

Output: Result=54321

[2]:

```python
def shift_decimal_digits(number, shift):
    str_number = str(number)
    num_digits = len(str_number)
    # Reverse the string if shift > number of digits
    if shift > num_digits:
        result = str_number[::-1]
    else:
        # Shift the digits to the left, wrapping around
        result = str_number[shift:] + str_number[:shift]
    return int(result)
n1 = 12345
shift1 = 1
result1 = shift_decimal_digits(n1, shift1)
print(f"Input: n={n1} shift={shift1}\nOutput: Result={result1}")

n2 = 12345
shift2 = 3
result2 = shift_decimal_digits(n2, shift2)
print(f"\nInput: n={n2} shift={shift2}\nOutput: Result={result2}")

n3 = 12345
shift3 = 5
result3 = shift_decimal_digits(n3, shift3)
```

```
print(f"\nInput: n={n3} shift={shift3}\nOutput: Result={result3}")

n4 = 12345
shift4 = 6
result4 = shift_decimal_digits(n4, shift4)
print(f"\nInput: n={n4} shift={shift4}\nOutput: Result={result4}")
```

```
Input: n=12345 shift=1
Output: Result=23451

Input: n=12345 shift=3
Output: Result=45123

Input: n=12345 shift=5
Output: Result=12345

Input: n=12345 shift=6
Output: Result=54321
```

22 311 Write a Python programme that accepts a string and calculate the number of uppercase letters, lowercase letters and number of digits.

For example,

Input: Hello Pyth@n is 100% easy

Output:

Uppercase letters : 2

Lowercase letters : 14

Digits : 3

[53]:

```
u=0
l=0
d=0
s= input("enter a string:")
for char in s:
    if char.isupper():
        u=u+1
    if char.islower():
        l=l+1
    if char.isdigit():
        d+=1
print("Uppercase count:", u)
print("Lowercase count:", l)
print("Digits count:", d)
```

```
enter a string:Vishal 10 Acharya 10 VHA 10
Uppercase count: 5
Lowercase count: 11
Digits count: 6
```

23 312 Write a python program to check the validity of a Password.

- Primary conditions for password validation:

1. Minimum 8 characters.
2. The alphabet must be between [a-z]
3. At least one alphabet should be of Upper Case [A-Z]
4. At least 1 number or digit between [0-9]
5. At least 1 character from [_ or @ or $]

Examples:

Input: Ram@_f1234

Output: Valid Password

Input: Rama_fo$ab

Output: Invalid Password

Explanation: Number is missing

Input: Rama#fo9c

Output: Invalid Password

Explanation: Must consist from _ or @ or $

[55]:

```
def password_check(password):
    l, u, p, d = 0, 0, 0, 0
    if len(password) >= 8:
        for i in password:
        # counting lowercase alphabets
            if (i.islower()):
                l+=1
        # counting uppercase alphabets
            if (i.isupper()):
                u+=1
        # counting digits
            if (i.isdigit()):
                d+=1
        # counting the mentioned special characters
            if(i=='@'or i=='$' or i=='_'):
                p+=1
        if (l>=1 and u>=1 and p>=1 and d>=1 and l+p+u+d==len(password)):
            print("valid")
        else:
            print("invalid")
    else:
        print("invalid")
password=input("enter password")
password_check(password)
```

```
enter passwordVishal@10
valid
```

24 313 Write a Python program to return another string similar to the input string, but with its case inverted.

For example, input of “Mr. Ed” will result in “mR. eD” as the output string. Note: Use of built in swapcase function is prohibited.

[56]:

```
input_string = "Mr. Ed"
print(input_string)
inverted_string = ""
for char in input_string:
        if char.isupper():
            inverted_string += char.lower()
        elif char.islower():
            inverted_string += char.upper()
        else:
            inverted_string += char
print(inverted_string)
```

```
Mr. Ed
mR. eD
```

25 314 Write a Python program to create a Caesar encryption.

Note: In cryptography, a Caesar cipher, also known as Caesar’s cipher, the shift cipher, Caesar’s code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a right shift of 3, A would be replaced by D, E would become H, and so on. The method is named after Julius Caesar, who used it in his private correspondence. For Example: Input Text : LJIET ENG

Shift : 3

Cipher: OMLHW HQJ

[4]:

```
def encrypt(message,key):
    message = message.upper()
    alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    result = ""
    for letter in message:
        if letter in alpha: #if the letter is actually a letter
            #find the corresponding ciphertext letter in the alphabet
            letter_index = (alpha.find(letter) + key) % len(alpha)
            result = result + alpha[letter_index]
        else:
            result = result + letter
    return result
input_text = input("enter string")
shift_value = int(input("enter a key:"))
cipher_text = encrypt(input_text, shift_value)
print("Input Text  :", input_text)
print("Shift        :", shift_value)
print("Cipher       :", cipher_text)
```

```
enter stringVishal Acharya
enter a key:4
Input Text  : Vishal Acharya
Shift        : 4
Cipher       : ZMWLEP EGLEVCE
```

26 315 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 and length of s1 & s2 should be same. The character's position doesn't matter.

[2]:

```
def balance(s1,s2):
    flag=True
    if (len(s1) == len(s2)):
        for i in s1:
            if i in s2:
                continue
            else:
                flag = False
        return flag
s1 = input()
s2 = input()
if balance(s1, s2) == True:
    print("Balance")
```

```
HELLLO
OLLEHL
Balance
```