

Python Assignment

Q1.

```
# Count the number of times a character appears in a given string
```

```
my_string = "welcome to cdac ditiss 2024"  
character = "w"  
count = my_string.count(character)  
print(count)
```

```
1
```

```
Process finished with exit code 0
```

Q2.

```
''' Accept n and Print the following  
a. If n is odd, print Odd  
b. If n is even and in the inclusive range of 2 to 5, print Accepted  
c. If n is even and in the inclusive range of 6 to 20, print Not Accepted  
d. If n is even and greater than 20, print Good '''
```

```
n = int(input("Enter a number: "))
```

```
if n % 2 != 0:  
    print("Odd")  
elif n % 2 == 0 and 2 <= n <= 5:  
    print("Accepted")  
elif n % 2 == 0 and 6 <= n <= 20:  
    print("Not Accepted")  
elif n % 2 == 0 and n > 20:  
    print("Good")
```

```
Enter a number: 24
```

```
Good
```

```
Process finished with exit code 0
```

Q3.

```
def is_leap_year(year):
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False

year = int(input("Enter a year: "))
if is_leap_year(year):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```

Enter a year: 2003
2003 is not a leap year.

Process finished with exit code 0

Q4.

```
# Write a program to accept IP address and port and display in list of list.
def ip():
    ip_port_list = []
    while True:
        ip_address = input("Enter IP address (or type 'exit' to finish): ")
        if ip_address.lower() == 'exit':
            break
        port = input("Enter port number: ")
        ip_port_list.append([ip_address, port])
        print("\nList of IP addresses and ports:") # Display the list of lists
    for entry in ip_port_list:
        print(entry)
ip()
```

Enter port number: 21

List of IP addresses and ports:

Enter IP address (or type 'exit' to finish): exit

['10.0.0.0', '10']

['25.25.25.25', '21']

Process finished with exit code 0

Q5.

In a given list insert a number at given position

```
def insert_number(lst, number, position):  
    if position < 0 or position > len(lst):  
        print("Invalid position")  
        return lst  
    lst.insert(position, number)  
    return lst  
  
my_list = [1, 2, 3, 4, 5]  
new_number = 10  
position = 2  
updated_list = insert_number(my_list, new_number, position)  
print(updated_list)
```

[1, 2, 10, 3, 4, 5]

Process finished with exit code 0

Q6

```
# Match the decimal in " Hello your score is 2.9" using regular expression  
  
import re  
  
text = "Hello your score is 33.2"  
match = re.search(r'\b\d+\.\d+\b', text)  
  
if match:  
    print(match.group())
```

33.2

Process finished with exit code 0

Q7

```
# Match the digit and non Digit using regular expression  
  
import re  
  
text = "hello I am Sanika Bhoyar my ccat rank is 2202, my phone no. 8779688500!"  
  
digit_matches = re.findall(r'\d', text)  
print("Digits found:", digit_matches)  
  
non_digit_matches = re.findall(r'\D', text)  
print("Non-digits found:", non_digit_matches)
```

Digits found: ['2', '2', '0', '2', '8', '7', '7', '9', '6', '8', '8', '5', '0', '0']

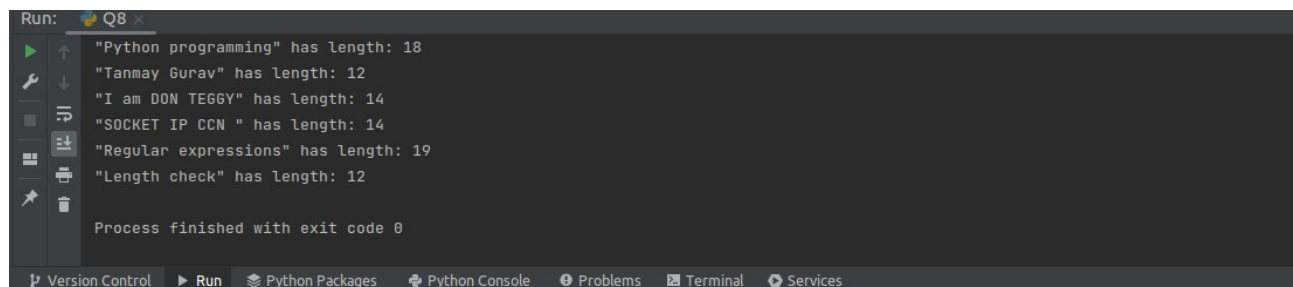
Non-digits found: ['h', 'e', 'l', 'l', 'o', ' ', 'I', ' ', 'a', 'm', ' ', 'S', 'a', 'n', 'i', 'k', 'a', ' ', 'B', 'h', 'o', 'y', 'a', 'r', ' ', 'm', 'y', ' ', 'c', 'c', 'a', 't', ' ', 'r', 'a', 'n', 'k', ' ', 'i', 's', ' ', ' ', ' ', ' ', 'm', 'y', ' ', 'p', 'h', 'o', 'n', 'e', ' ', ' ', 'n', 'o', ' ', ' ', ' ', '!']

Process finished with exit code 0

Q8

Create a list of string of varied length and display the length of individual string in list

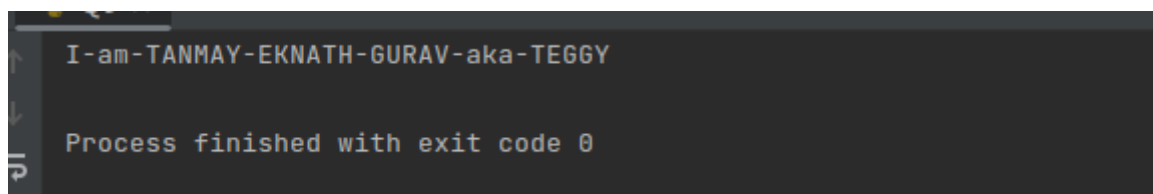
```
string_list = [  
    "Hello",  
    "World",  
    "Python programming",  
    "Tanmay Gurav",  
    "I am DON TEGGY",  
    "SOCKET IP CCN ",  
    "Regular expressions",  
    "Length check"  
]  
  
# Display the length of each string in the list  
for string in string_list:  
    print(f'"{string}" has length: {len(string)}')
```



```
Run: Q8 x  
"Python programming" has length: 18  
"Tanmay Gurav" has length: 12  
"I am DON TEGGY" has length: 14  
"SOCKET IP CCN " has length: 14  
"Regular expressions" has length: 19  
"Length check" has length: 12  
  
Process finished with exit code 0  
Version Control Run Python Packages Python Console Problems Terminal Services
```

Q9

```
# You are given a string. Split the string on a " " (space) delimiter and join using a - hyphen.  
input_string = "I am TANMAY EKNATH GURAV aka TEGGY"  
split_string = input_string.split(" ")  
joined_string = "-".join(split_string)  
print(joined_string)
```



```
I-am-TANMAY-EKNATH-GURAV-aka-TEGGY  
  
Process finished with exit code 0
```

Q8

#Write a function to check whether dictionary has duplicates

```
def has_duplicates(input_dict):  
    values = list(input_dict.values())  
    unique_values = set(values)  
    return len(unique_values) < len(values)
```

```
my_dict = {  
    'a': 1,  
    'b': 2,  
    'c': 3,  
    'd': 2  
}  
  
print(has_duplicates(my_dict))  
another_dict = {  
    'x': 1,  
    'y': 2,  
    'z': 3  
}  
  
print(has_duplicates(another_dict))
```

```
True  
False
```

```
Process finished with exit code 0
```

Q11.Perform cumulative sum of all the members in list and tuples

```
def cumulative_sum_of_lists_and_tuples(data):  
    total = 0  
    result = []  
  
    for item in data:  
        if isinstance(item, (list, tuple)):  
            for num in item:  
                total += num  
                result.append(total)  
        else:  
            raise ValueError("All items must be lists or tuples containing numbers.")  
  
    return result  
  
data = [1, (2, 3), [4, 5], (6,), [7]]  
cumulative_result = cumulative_sum_of_lists_and_tuples(data)
```

```
print(cumulative_result)
```

Output: [1, 3, 6, 10, 16, 23, 30]

Q12.

```
#Computes the sum of all numbers in a list of lists.
```

```
def sum_of_lists(lst_of_lists):  
    total_sum = 0  
  
    for sublist in lst_of_lists:  
        total_sum += sum(sublist)  
  
    return total_sum
```

```
data = [[1, 2, 3], [4, 5], [6]]  
total = sum_of_lists(data)  
print(total)
```

```
21
```

```
:du
```

```
Process finished with exit code 0
```

Q13

```
#Count a given letter in string
```

```
def count_letter(string, letter):  
    return string.count(letter)
```

```
input_string = "Hello, how many times does the letter 'o' appear?"
```

```
letter_to_count = 'o'
```

```
count = count_letter(input_string, letter_to_count)
```

```
print(f"The letter '{letter_to_count}' appears {count} times in the string.")
```

```
The letter 'o' appears 4 times in the string.
```

```
Process finished with exit code 0
```

Q14

```
#Invert the dictionary
def invert_dict(d):
    inverted = {}

    for key, value in d.items():

        if value in inverted:
            if not isinstance(inverted[value], list):
                inverted[value] = [inverted[value]]
            inverted[value].append(key)
        else:
            inverted[value] = key

    return inverted

original_dict = {'a': 1, 'b': 2, 'c': 1}
inverted_dict = invert_dict(original_dict)
print(inverted_dict)
```

output :

```
{1: ['a', 'c'], 2: 'b'}
```

```
Process finished with exit code 0
```

Q15

#15. Check whether a given word is palindrome or not

```
def is_palindrome(word):
    normalized_word = word.lower()
    return normalized_word == normalized_word[::-1]

word_to_check = "Racecar"
result = is_palindrome(word_to_check)
if result:
    print(f'{word_to_check} is a palindrome.')
else:
    print(f'{word_to_check} is not a palindrome.')
```

```
'Racecar' is a palindrome.
```

```
Process finished with exit code 0
```

#Q16. Remove First and last element of list

```
def remove_first_last(lst):
```

```
    if len(lst) <= 2:
```

```
        return []
```

```
    return lst[1:-1]
```

```
example_list = [1, 2, 3, 4, 5]
```

```
modified_list = remove_first_last(example_list)
```

```
print(modified_list)
```

Q17