m2_case_study_2

October 10, 2024

1 Module 2 – Sequences and File Operations

1.1 Case Study – 2

[4]:
```
#1. What is the output of the following code?
nums = set([1,1,2,3,3,3,4,4])
print(nums)
print(len(nums))
#Hint: Set consists of a unique element.
```

```
{1, 2, 3, 4}
4
```

[9]:
```
#2. What will be the output?
d = {"john":40, "peter":45}
print(list(d.keys()))
print(list(d.values()))
for k,v in d.items():
    print(k,v)
#Hint: d.keys() is the function that will show keys.
```

```
['john', 'peter']
[40, 45]
john 40
peter 45
```

1.1.1 3. A website requires a user to input a username and password to register. Write a program to check the validity of the password given by the user. Following are the criteria for checking password:

1. At least 1 letter between [a-z]
2. At least 1 number between [0-9]
3. At least 1 letter between [A-Z]
4. At least 1 character from [$#@]
5. Minimum length of transaction password: 6
6. Maximum length of transaction password: 12 Hint: In the case of input data being supplied to the question, it should be assumed to be a console input.

[10]:
```python
import re

def check_password_validity(password):
    """
    This function checks the validity of a password based on several criteria:
    1. At least 1 letter between [a-z]
    2. At least 1 letter between [A-Z]
    3. At least 1 number between [0-9]
    4. At least 1 character from [$#@]
    5. Length of password must be between 6 and 12 characters

    Parameters:
    password (str): The password to be validated.

    Returns:
    None: It prints whether the password is valid or not.
    """

    # Criteria checks using regular expressions
    if (len(password) < 6 or len(password) > 12):
        print("Invalid password! The password must be between 6 and 12␣
↪characters.")
        return

    if not re.search("[a-z]", password):
        print("Invalid password! The password must contain at least one␣
↪lowercase letter.")
        return

    if not re.search("[A-Z]", password):
        print("Invalid password! The password must contain at least one␣
↪uppercase letter.")
        return

    if not re.search("[0-9]", password):
        print("Invalid password! The password must contain at least one digit.")
        return

    if not re.search("[$#@]", password):
        print("Invalid password! The password must contain at least one special␣
↪character from [$#@].")
        return

    # If all criteria are met
    print("Password is valid.")

# Input: Asking the user to input a password
```

```
password = input("Enter a password to check its validity: ")

# Function call
check_password_validity(password)
```

```
Enter a password to check its validity:  Pa$$W0rd

Password is valid.
```

1.1.2 4. Write a for loop that prints all elements of a list and their position in the list.

a = [4,7,3,2,5,9]

Hint: Use Loop to iterate through list elements

[12]:
```
# Given list
a = [4, 7, 3, 2, 5, 9]

# Using a for loop to iterate through the list elements
for index in range(len(a)):
    # Print the index and the corresponding element
    print(f"Element at position {index}: {a[index]}")
```

```
Element at position 0: 4
Element at position 1: 7
Element at position 2: 3
Element at position 3: 2
Element at position 4: 5
Element at position 5: 9
```

1.1.3 6. Please write a program that accepts a string from the console and print it in reverse order.

Example: If the following string is given as input to the program: rise to vote sir Then, the output of the program should be: ris etov ot esir

[15]:
```
def reverse_string(input_string):
    """
    This function takes a string as input and prints it in reverse order.

    Parameters:
    input_string (str): The string to be reversed.

    Returns:
    None: It prints the reversed string.
    """

    # Reverse the string using string slicing
    reversed_string = input_string[::-1]
```

```
    # Print the reversed string
    print("Reversed string:", reversed_string)

# Input: Asking user to input a string
input_string = input("Enter a string: ")

# Function call
reverse_string(input_string)
```

```
Enter a string:  rise to vote sir
```

```
Reversed string: ris etov ot esir
```

1.1.4 7. Please write a program that counts and prints the numbers of each character in a string input by the console.

Example: If the following string is given as input to the program: abcdefgabc Then, the output of the program should be: a,2 c,2 b,2 e,1 d,1 g,1 f,1

[16]:
```
def count_characters(input_string):
    """
    This function counts and prints the occurrences of each character in the␣
 ↪input string.

    Parameters:
    input_string (str): The string to be analyzed.

    Returns:
    None: It prints each character and its count.
    """

    # Create an empty dictionary to store character counts
    char_count = {}

    # Loop through each character in the input string
    for char in input_string:
        # Update the count for each character
        if char in char_count:
            char_count[char] += 1
        else:
            char_count[char] = 1

    # Print the characters and their counts
    for char, count in char_count.items():
        print(f"{char},{count}")

# Input: Asking user to input a string
```

```
input_string = input("Enter a string: ")

# Function call
count_characters(input_string)
```

```
Enter a string:  Gimme! Gimme! Gimme!
```

```
G,3
i,3
m,6
e,3
!,3
 ,2
```

1.1.5 8. With two given lists [1,3,6,78,35,55] and [12,24,35,24,88,120,155], write a program to make a list whose elements are intersection of the above given lists.

[18]:
```
def list_intersection(list1, list2):
    """
    This function finds the intersection of two lists using the set␣
 ↪intersection method
    and returns a new list with common elements.

    Parameters:
    list1 (list): The first list.
    list2 (list): The second list.

    Returns:
    list: A new list containing the elements that are common to both list1 and␣
 ↪list2.
    """

    # Convert both lists to sets and find the intersection using the␣
 ↪intersection() method
    intersection = list(set(list1).intersection(set(list2)))

    # Return the intersection
    return intersection

# Given lists
list1 = [1, 3, 6, 78, 35, 55]
list2 = [12, 24, 35, 24, 88, 120, 155]

# Function call and result
result = list_intersection(list1, list2)

# Print the result
print("The intersection of the two lists is:", result)
```

```
The intersection of the two lists is: [35]
```

1.1.6 9. By using list comprehension, please write a program to print the list after removing the value 24 in [12,24,35,24,88,120,155].

[23]:
```
# Original list
numbers = [12, 24, 35, 24, 88, 120, 155]

# Using list comprehension to remove all occurrences of 24
filtered_list = [num for num in numbers if num != 24]

# Print the resulting list
print("List after removing 24:", filtered_list)
```

```
List after removing 24: [12, 35, 88, 120, 155]
```

1.1.7 10.By using list comprehension, please write a program to print the list after removing the 0th,4th, and 5th numbers in [12,24,35,70,88,120,155].

[31]:
```
# Original list
numbers = [12, 24, 35, 70, 88, 120, 155]

# Using list comprehension to exclude the 0th, 4th, and 5th elements
filtered_list = [num for index, num in enumerate(numbers) if index not in (0,␣
↪4, 5)]

# Print the resulting list
print("List after removing the 0th, 4th, and 5th elements:", filtered_list)
```

```
List after removing the 0th, 4th, and 5th elements: [24, 35, 70, 155]
```

1.1.8 11.By using list comprehension, please write a program to print the list after removing deleted numbers that are divisible by 5 and 7 in [12,24,35,70,88,120,155].

[36]:
```
# Original list
numbers = [12, 24, 35, 70, 88, 120, 155]

# Using list comprehension to remove numbers divisible by both 5 and 7
filtered_list = [num for num in numbers if not (num % 5 == 0 and num % 7 == 0)]

# Print the resulting list
print("List after removing numbers divisible by both 5 and 7:", filtered_list)
```

```
List after removing numbers divisible by both 5 and 7: [12, 24, 88, 120, 155]
```

1.1.9 12.Write a program to compute 1/2+2/3+3/4+...+n/n+1 with a given n input by console (n>0).

Example:

If the following n is given as input to the program: 5

Then, the output of the program should be: 3.55

[42]:
```python
def compute_series(n):
    """
    This function computes the sum of the series 1/2 + 2/3 + 3/4 + ... + n/
 ↪(n+1).

    Parameters:
    n (int): The value of n for the series.

    Returns:
    float: The computed result of the series.
    """

    # Initialize the sum
    series_sum = 0

    # Loop to compute the sum of the series
    for i in range(1, n+1):
        series_sum += i / (i + 1)

    return series_sum

# Input: Asking the user to input a positive integer n
n = int(input("Enter a positive integer n: "))

# Check if n is greater than 0
if n > 0:
    result = compute_series(n)
    # Print the result, rounded to 2 decimal places
    print(f"The result of the series is: {round(result, 2)}")
else:
    print("Please enter a number greater than 0.")
```

```
Enter a positive integer n:  100
The result of the series is: 95.8
```

[ ]:

[6]:
```
#### Mr Akram M'Tir 10-10-2024
```

[ ]: