Assignment #02

Hope to Skills

Free Artificial Intelligence Advance Course

Instructor: Irfan Malik, Dr. Sheraz

Submission:

- Make a Google Collab notebook to implement this assignment.
- In case you face difficulty in creating the Google Collab Notebook Follow these **Steps**
- Submit a **.ipynb** file detailing all the information. No other format will be accepted
- Submission file should be named as **Assignment_02_StudentName.ipynb**
- Deadline for this Assignment is **Tuesday 20-02-2024.**
- Strictly follow the submission deadline.
- Make Submission in the **Assignment-02** Google Form and press the submit button.
- Click here to submit the Assignment

What you will learn

- How to create the google Collab NoteBook from Scratch.
- Using the basic built in functions and apply the following.

Solve the Following Task

1. Write a program that takes a string as input and counts the number of vowels and consonants in the string.   **(5 marks)**

    Input: Hello, World!

    Output:

    > Number of vowels: 3

    > Number of consonants: 7

2. Write a Python program that accepts a filename from the user and prints the extension of the file. **(5 marks)**

    Sample filename: document.docx

Output: docx

3. Write a Python program to check if value 200 exists in the following dictionary. **(5 marks)**

       Sample_dict = {'a': 100, 'b': 200, 'c': 300}

4. The given tuple is a nested tuple. Write a Python program to print the value 20. **(5 marks)**

       tuple1 = ("Orange", [10, 20, 30], (5, 15, 25))

5. Write a Python program that calculates the Body Mass Index (BMI) for a person based on their weight (in kilograms) and height (in meters). The BMI is calculated using the following formula: **(5 marks)**

       BMI = (weight) / (height^2)

Instructions:

- Prompt the user to enter their weight in kilograms.
- Prompt the user to enter their height in meters.
- Calculate the BMI using the provided formula.
- Display the calculated BMI to the user.

Additionally, *provide an interpretation of the BMI according to the following categories:*

- BMI < 18.5: Underweight
- 18.5 <= BMI < 25: Normal weight
- 25 <= BMI < 30: Overweight
- BMI >= 30: Obese

6. Write a Python program that prompts the user to input a string. The program should count the frequency of each character in the string and then print the result as a dictionary.**(5 marks)**

EXAMPLE: If the user enters "hello" , the program should output **{'h': 1, 'e': 1, 'l': 2, 'o': 1}** indicating that 'h' appears once, 'e' appears once, 'l' appears twice, and 'o' appears once in the input string.

7. Write a Python program that takes a string input from the user and checks if it is a valid email address. If it is valid, print "Valid email address", otherwise print "Invalid email address".**(5 marks)**

   > ***Hint:*** Check if the string contains **"@"** and ends with **".net"**, or **".com"** to check validity. ***Example:*** If the email is "abc@example.com", then it is a valid email, otherwise an Invalid email.

8. Given a tuple of tuples, write a Python program to print the sum of the elements in each inner tuple. **(5 marks)**
   > ***Input:*** ((1, 2, 3), (4, 5, 6), (7, 8, 9))
   > ***Output:*** 6, 15, 24

9. Create a list containing the 10 numbers . Print the first Three prime numbers from the list. **(5 marks)**
   > ***Expected Output:*** List of the first three prime numbers: [2, 3, 5]

10. Given a list of dictionaries representing students' information (name, age, grade), Write a Python program to sort the students based on their grades in descending order and print the sorted list. **(5 marks)**

    Dictionary: [
    > {"name": "John", "age": 20, "grade": 85},
    > {"name": "Alice", "age": 22, "grade": 90},
    > {"name": "Bob", "age": 21, "grade": 80}

    ]

11. Create a program that takes a list of multiple numbers separated with commas (,) from the user and finds the total number of unique values in the list and also prints them. **(5 marks)**

> ***Expected_format:***
>
> > Total number of unique values: 6
> >
> > Unique values in the list: [1, 2, 3, 4, 5, 6]

12. Write a program that takes two lists as input and returns a new list containing elements that are common to both lists.***(Hint:*** Use list methods to implement this.) **(5 marks)**

> ***Input Lists:***
>
> > List 1: [1, 2, 3, 4, 5]
> >
> > List 2: [4, 5, 6, 7, 8]
> >
> > Common Elements: [4, 5]

13. Write a program that takes two dictionaries as input and merges them into a new dictionary. If there are common keys, combine their values. **(5 marks)**

> **dict1**= {'a':5, 'b':10, 'c':15}
>
> **dict2**= {'b':20, 'c':25, 'd':30}
>
> ***Output:*** {'a': 5, 'b': 30, 'c': 40, 'd': 30}

14. Write a program that takes two numbers, **base** and **exponent**, and calculates the result of raising base to the power of exponent using the ** operator. **(5 marks)**

> ***Input:*** Base: 2
>
> > Exponent: 3
>
> ***Result:*** 8

15. Write a program that takes a temperature in Celsius as input and converts it to Fahrenheit using the formula. **(5 marks)**

> **Fahrenheit = (Celsius * 9/5) + 32.**

16. Write a Python program to reverse a list **without using built-in functions**.
    (5 marks)

17. Given a string "Hello World", write a Python program to reverse the string without reversing the individual words. **(5 marks)**

     Expected Output: "World Hello"

18. Implement a Python function **is_palindrome()** that checks whether a passed string is a palindrome (reads the same forward and backward) and tests it with "radar".**(5 marks)**

19. Write a program that returns a list of even numbers from a given list. **(5 marks)**

     Input: [2, 5, 3, 7, 9, 53, 10, 32, 65, 76, 98]

20. Given two lists **keys = ['a', 'b', 'c']** and **values = [1, 2, 3]**, write a Python program to create a dictionary from these lists. **(5 marks)**

     Expected Output: {'a': 1, 'b': 2, 'c': 3}