**Date: 2024--**

**Assignment 2**

This assignment consists of the programming questions related to the topics of week 1 and week 2. The main topics of questions are: Python functions and data structures.

All the students are required to follow the format of the program as specified in the guideline below.

1. All the programs should have initial **doc string** comment (‘’’ description of program‘’’) mentioning what your program will do.
2. Try to maintain single/multi-line comments in the place where needed to make the program understandable.
3. Maintain proper indention and newline spaces to increase the readability of the program.
4. The deliverable are 2 type of files (a single word file and multiple python program files):
   1. Separate python program files with **.py** extension (e.g. program\_name.py). Provide a relevant name to your program file on the basis of functionality of the program.
   2. A word file describing the working of all the programs according to their number. The details required in this is the description of program, screenshot of the testing (input given and output obtained in the execution environment such as IDLE or Command prompt or terminal whichever you prefer.). It is preferred that you work with multiple inputs and outputs.

**Questions**

1. Write a program to create a function which accepts 2 numbers and displays the sum, difference, product and the remainder values.
2. Write a program to create separate functions for below mentioned mathematical calculations which would return the values back to the program. The functions should accept the 2 number which are inputs from the user and passed to them. The program should display the output in a proper format.
3. Addition
4. Subtraction
5. Multiplication
6. Division
7. Modulo division
8. Floor division
9. Write a program to create a function which will accept a parameter and return the factorial of the number. The output should be displayed in a proper format.
10. Write a function to accept a list of numbers and print the occurrence of each number. The function should be tested well in the program by calling and sending various list of numbers.
11. Write a function to accept a list of names and return the sorted order of names back.
12. Write a function to accept a 2 parameters, one is the list of cities and another is the city that the user wants to search. The function should search the city in the list of cities and return the index of the list where the city is available. If the city is not available, the program should return a proper message.
13. Write a function word\_frequency(sentence) that takes a sentence as input and returns a dictionary containing the frequency of each word in the sentence. [Hint: split the sentence into words and iterate to check the word frequency.]
14. Write a program to accept a list of numbers from the user and should return a list by removing the duplicate values, if any.
15. Write a program to ask the details of 5 books (title, author, ISBN, cost), add them in the dictionary and print them.
16. Create a program to ask a user to give continuous input of numbers until they like. The program should keep on segregating the user input numbers into even and odd lists separately. Once the user completes the input and opts for exiting the program, the program should display the separate list of even and odd lists in a proper format.
17. Write a program to generate a card guessing game for the users in an interesting way. The card should have property such as name and value (e.g. ace 10). Specifications are as mentioned below.
18. The program should have a list of card values like 2, 3, 4,…., Jack, Queen, King, Ace
19. The program should have a list of card suits like heart, diamond, club, spades.
20. The program should randomly pick up a number and a suit and keep as an answer in a separate list.
21. The program should ask the player to guess the card value and the suit.
22. The program should check the player guessed value with the computer answer value. If both the parts don’t match, the program should show a broken heart and game over to the player. If any one of the part of answer matches, the program should show a smily face to the player. If both the guesses of the player matches with the program answer, the program should show a heart and a smily face to the user.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***End of Assignment 2** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*