Programming Fundamentals LAB – BSDSF23

(Both Morning and Afternoon)

Lab 08 – 07-11-2023

Note: YOU may USE Command Prompt or ***Mu Editor*** to interpret and execute all the PYTHON programs. Use of any IDE, except ***Mu Editor*** is not allowed for this LAB, despite you are expert. Unless and until you convinced me of it personally.

Also note, if the computer systems are not equipped with python interpreter, you may use online compiler at the following URL highlighted in yellow. And in case it is also difficult to use for any reason, you need to do the paperwork within the LAB time for all tasks, as discussed in the class sessions. Thanks

https://www.programiz.com/python-programming/online-compiler/

Tasks set 1 (15 each)

1. Create an array of reasonably big sizes, say 50, or at least above 20. Enter integer values in it as input or random values. Create a function that accepts the **array description** and two other parameters, say **v** and **sum**. The function returns the count of those values which when added **v** to them gives **sum**. Test the working of the function with a number of appropriate test cases.

2. Create a function that accepts the **array description** and another parameter say **sum**. The function returns the count of pairs of values in array when added in each other gives **sum**. Test the working of the function with a number of appropriate test cases.

3. Create a function that accepts the **array description** as parameters and return count of inversions in it. Inversion is a case in which for two integers i<j, ith element in the array is larger than jth element. Test the working of the function with a number of appropriate test cases. Note: you may use any method and also use similar arrays and similar concepts in the above tasks.

4. Create a **string** of reasonably big sizes, say a good sentence like “Pakistan played well in last match, scored 200 and defeat a team having 400+ in first inning”. Using loop display all the characters in array one character per line with space separated its **ascii** code, **ord** function is used to get **ascii** code of a character. Test the working of the function with a number of appropriate test cases other than specified above.

5. Demonstrate the working of **chr** built-in function by printing integer starting from 32 upto 126. On the same line space separated **character** to corresponding integer should also be printed. Later, also print the integer from range 0 to 31 and corresponding characters, and same from 128 to 255.

Tasks set 2 (25 each)

6. Create three arrays for midterm, final term and sessional marks, midterm’s marks are upto 35, final term’s marks are upto 40 and sessional’s upto 25. Enter your expected hard code marks (type in program don’t get them as input) in all these arrays. Later calculate and print total marks of each subject and averages of all four (mid, final, sessional, and total).

Thanks, for your **patience**

If you got time, solve some pending tasks from previous labs