

# Programming Fundamentals LAB – BSDSF23

(Both Morning and Afternoon)

## Lab 09 – 14-11-2023

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 (10 each)

1. Demonstrate the working of **chr** and **ascii** built-in function by printing integer starting from 0 up to 255, the **ascii's** of their **chr's**. The output should be in two columns, the **number** and **ascii of chr**.
2. 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 returns another array with the running average of its parameter. Running average means **k<sup>th</sup>** item of the array is average of all the items at indices less than or equal to **k** in the original array. Test the working of the function with a number of appropriate test cases.
3. Create a function that accepts a **string description** as parameter and return the count of non-alphanumeric (not alphabets or digits) in it. Test the working of the function with a number of appropriate test cases.
4. Create a function that accepts a **string description** as parameter and verify it consists of only English alphabets and spaces, tabs, commas and only one period at its end. Test the working of the function with a number of appropriate test cases.
5. Create a function that accepts a **string description** as parameter and verify it consists of only digits, \$, commas, and period, e.g., \$23,000,00.00, 66,93,73,210, and \$332,553,765,324,213. The function removes unnecessary characters from the string and then using **int** function converts it into a number to return. Test the working of the function with a number of appropriate test cases.
6. Create a function named **isSorted** accepts an array description and returns True or False depending the array is in descending order or not. Test the working of the function with a number of appropriate test cases.

### Tasks set 2 (25 each)

7. Make a **running** calculator with the following functionalities: plus, minus, multiply, divide, power, remainder using python operators. Also, provide, stop, initialize, last, continuing last. When calculator programs start, it displays 0. Next it inputs a command or functionality from the user, performs the functionality and displays the updated value on screen, it always performs the operation on the last value displayed on the screen. The special functionalities **stop** will stop the calculator program, **initialize** will display again 0 value, **last** will display the second last value and continuing **last** commands one by one will displays the previous values. An example execution of the calculator program is on the next page, you need to run several test cases.

Thanks, for your **patience**

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

**Note:** You may need too many *ifs*, functions, loop or loops, or variable or array to manage the running calculator.

<i>Action or command</i>	<i>Screen update</i>
Start of calculator	0
+	+
90	90
/	/
15	6
*	*
7	42
Initialize	0
-	-
5	-5
*	*
10	-50
+	+
72	22
Last	-50
/	/
-5	10
-	-
2	8
+	+
7	15
+	+
7	22
+	+
7	29
+	+
11	40
Last	29
Last	22
Last	15
Stop	Stop the calculator program