

Introduction to Programming with Python

Weekly Programming Assignment – Week 11

All solution files must be submitted at CodeGrade/Assignment enabled Link for grading.

All solutions must be uploaded on or before 1st of December 2021 at 11:59 PM

Exercise 1

Write a procedure **LinearSearchNumber()** which takes a list that contain numbers only (float or int or both), and search value (float or int) as arguments and prints the index of the specified search value if exists in the list. Else, print “**Not exists in the list**”. If the user passed search value is not a number (int/float) then throw an appropriate message via exception handling and terminate the program. The sample run is here:

Use **list1 = [12,34,78,11,90,45,12.4,34.10,78.3,11.5,90.12,45.6]** to check your procedure in the main program.

```
Python 3.7.9 (bundled)
>>> %Run Q1_W11.py
Enter a search value:12abc
Search value must be int or float
>>> %Run Q1_W11.py
Enter a search value:78.3
It exists: 8
>>> %Run Q1_W11.py
Enter a search value:67
It does not exist
>>> %Run Q1_W11.py
Enter a search value:abc
Search value must be int or float
>>> %Run Q1_W11.py
Enter a search value:12
It exists: 0
...
```

Exercise 2

Write a function **binsearchList(l1, value)** that accepts list and search value as arguments and return the index of search value if exists else, “**not exists**” to the called program by using binary search algorithm. The expected output will be:

```
16 #Main program
17 list1 = [12,34,23,89,20]
18 print(binsearchList(list1, 89))
19 print(binsearchList(list1, 189))
20
```

Shell ×

```
Python 3.7.9 (bundled)
>>> %Run Q2_W11.py

4
not exists
```

Exercise 3

Write a procedure **InsertionSortTuple()** that accepts tuple as argument and print its values in **descending order**. Your procedure must use **insertion sort algorithm / technique** to sort the values of tuple. Save the file as “**InsertionModule.py**”. The sample run is here: Submit **only the module file**. **Do not with main program**

```
13 #Main program
14 tuple1 = (12,34,3,89,12,-46,10)
15 InsertionSortTuple(tuple1)
```

```
Shell x
Python 3.7.9 (bundled)
>>> %Run Q3_W11.py
(89, 34, 12, 12, 10, 3, -46)
```

Exercise 4 (self-study)

Bubble sort is a kind of sorting algorithm that works repeatedly by swapping adjacent elements if they are not in order. Write a function **BubbleSort(list/tuple, sort order)** that takes *list or tuple* and *sort order(int)* (1 for ascending / 2 for descending) as arguments and return in sorted form to the called program based on user choice (ascending or descending). Check your program with example test program saved separately given below. But your program will also be checked with tuple as well.

```
Q4_W11.py x
1 import BubbleSort as bs
2 list1 = [12,34,78,11,90,45,12.4,34.11,78.3,11.5,90.12,45.6]
3 print(bs.BubbleSort(list1,2))
4
```

```
Shell x
Python 3.7.9 (bundled)
>>> %cd 'Z:\Python 2021_Fall\Fall 2021_CT60A0203\Week 11\Weekly programming questions'
>>> %Run Q4_W11.py
[90.12, 90, 78.3, 78, 45.6, 45, 34.11, 34, 12.4, 12, 11.5, 11]
>>>
```

Your function must use **bubble sort technique** to arrange the elements in order as requested. Save the file that with the function only as “**BubbleSort.py**”. **Do not include your test program.**

Exercise / task Number	Codegrade link_Moodle for file solution files upload	Points / Marks
1	Exercise1_Week 11 with main/test program	15
2	Exercise2_Week 11 with main/test program	15
3	Exercise3_Week 11 as “ InsertionModule.py ”	20

4	Exercise 4_Week 11 as “BubbleSort.py”	20
---	---------------------------------------	----