Introduction to programming

Data structures and algorithms - Lab

Exercise 1 – filtered_exam_results.py

You have been given a list of student's exam results for a particular subject. There are several results which have fallen below the 50% pass rate. Write a program that creates a new list of exam results with any result below 50 being removed.

Exam results are as follows:

```
21, 11, 4, 96, 48, 5, 13, 64, 28, 33, 43, 20, 70, 24, 88, 57, 31, 9, 35, 47, 56, 45, 14,
74, 35, 6, 79, 62, 17, 83, 5, 8, 44, 56, 60, 47, 17, 23, 96, 66, 17, 43, 7, 21, 18, 100,
30, 8, 15, 15
```

There should be 15 exam results with a passing score overall. Save your program as **filtered_exam_results.py** in your Week 4 folder.

Exercise 2 – customer_numbers.py

You are working on part of a Customer Relationship Management application and your manager has asked you to create a program that uses a **dictionary** to store customer's names and telephone numbers. You will need to create a dictionary data structure and store some test data in it. The program should prompt users, asking them for the name of a person they need a phone number for. The program should then respond with the correct phone number for that user.

Here is some sample data you can use:

Ronnie Sullivan	01234 567890
Marilyn Tucker	01987 654321
Brad Bennett	01897 65320
Rodney White	020 765 6655
Brian Burton	01382 763999

Save your program and test that it returns the correct phone numbers for each person in the dictionary. Save your file as **customer_numbers.py** in your Week 4 folder.

Extension: Add the ability for users to enter new information into the dictionary which can then be accessed.

Exercise 3 – stack.py

Write a program that creates a stack as described in the reading material this week. Your program should show how to add (push) values on to the stack and also to remove them (pop). Your program should display the contents of the stack after changes have been made in a top down, vertical view. For example after pushing 3 values on to the stack you should print out your stack like this:

Value 3
Value 2
Value 1

There is some flexibility in this exercise and it is up to you with regards of what values to add to the stack. Your program should just needs to demonstrate a working stack. Save your program as **stack.py** in your Week 4 folder.

Extension: Enhance your stack program by creating a menu that users can use to add and remove items from the stack and also a third option for displaying the stack.

Extension task

This week you learnt about several different types of sorting and searching algorithms. Take one of these algorithms and implement it in Python. Save the file with an appropriate name in your Week 4 folder.