**Unit 3 Outcome 1 – 2017**

**Outcome 1** On completion of this unit the student should be able to interpret designs and apply a range of functions and techniques using a programming language to develop working modules. To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 1.

**Key skills**

• interpret designs to develop working modules that meet these requirements

• use a range of data types and structures

• use appropriate processing features of a programming language

• select and use appropriate techniques to test the functionality of modules

• document the functioning of modules through the use of internal documentation.

**Task 1:**

**Create a Graphical User Interface that enables a user to enter the data in the following data dictionary, and clears the file.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data Type** | **Description** | **Identifier** | **Domain(range or values)** |
| Customer F Name | string | This is the customer’s first name. | strCustFirstName |  |
| Customer L Name | string | This is the customer’s last name. | strCustLastName |  |
| Customer house number | int | This is the house number. | intHouseNumber | Only a whole number. |
| Customer street name | string | This is the customer’s street name. | strStreetName |  |
| Customer suburb | string | This is the customer’s suburb. | strSuburb |  |
| Customer postcode | int | This is the customer’s postcode. | intPostcode | Between 3000-3999 |
| Customer phone number | string | This is the customer’s phone number | strPhoneNumber |  |

**Task 2**

**Create a module that will swap two objects, when two values are declared and the module swaps the contents of each text based object.**

Pseudocode

Begin (procedure Swap –ItemA, ItemB)

Temp temp A

ItemA ItemB

ItemB temp

End

**Task 3**

**Create a module for the following inputs and outputs:**

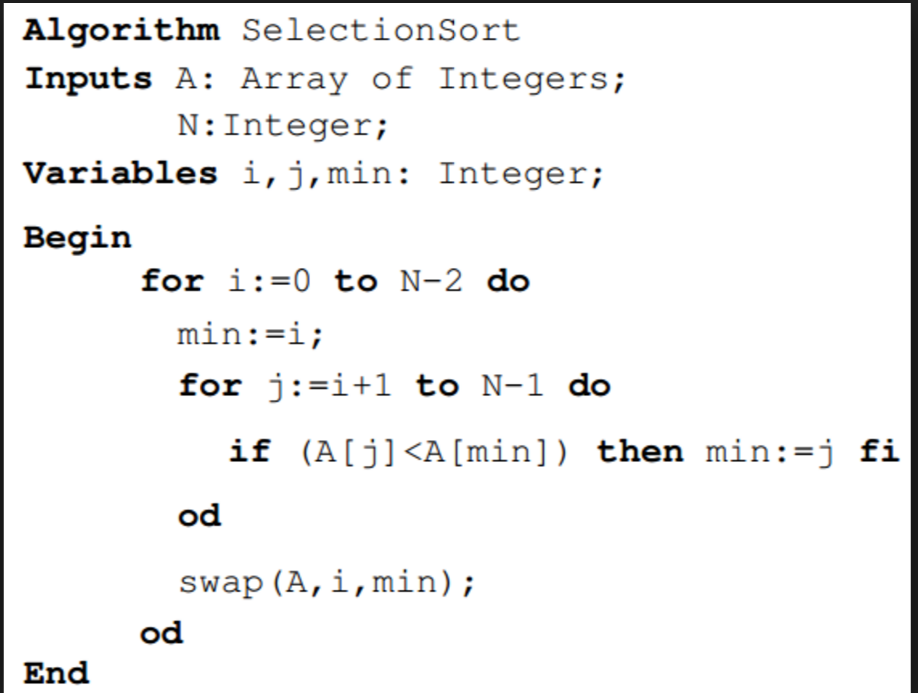
**Inputs:** Customer First Name, Customer last Name, Customer phone number, customer delivery address (including house number, street name, suburb and post code),

**Outputs:** All of the inputs and a welcome message to the customer for ordering from them.

**Task 4**

**Create a program for either of the following sort algorithms:**

1. Selection sort



or

selectionSort(array a)

     //Search for the minimum element and adds to the sorted sub array

    for i in 0 -> a.length - 2 do

        minIndex = i

    //Find minimum element in the remaining sub array and update the minIndex

        for j in (i + 1) -> (a.length - 1) do

            if a[j] < a[minIndex]

                minIndex = j

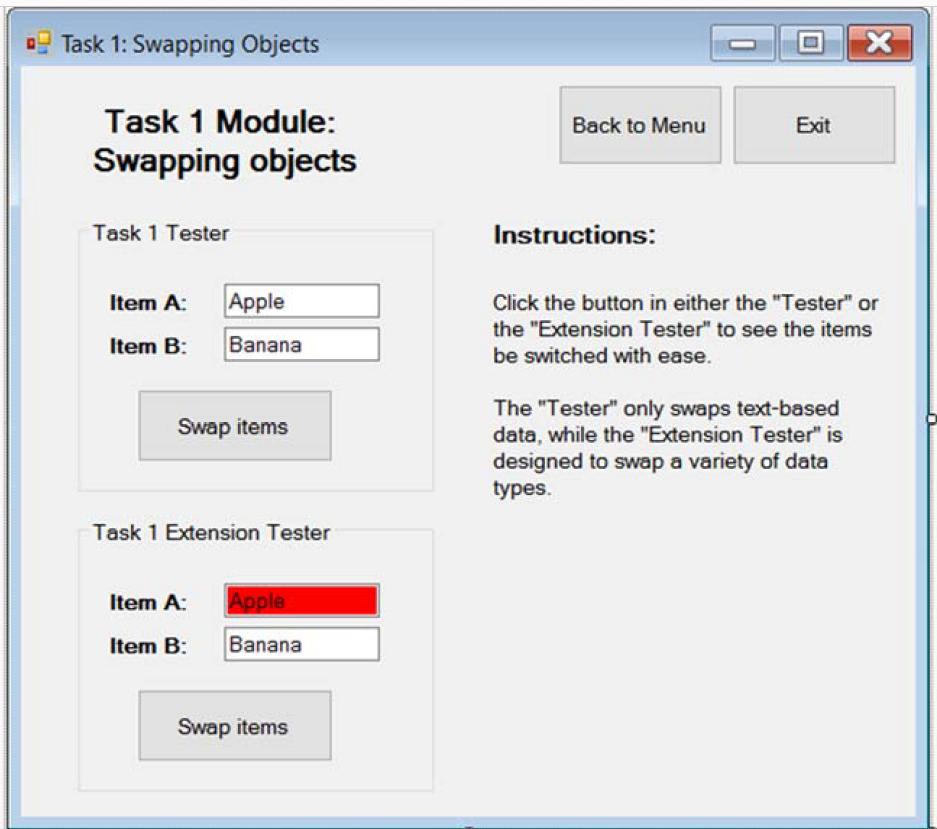
    //Swap the minimum value find with the first element of unsorted subarray

        swap(a[i], a[minIndex])

or

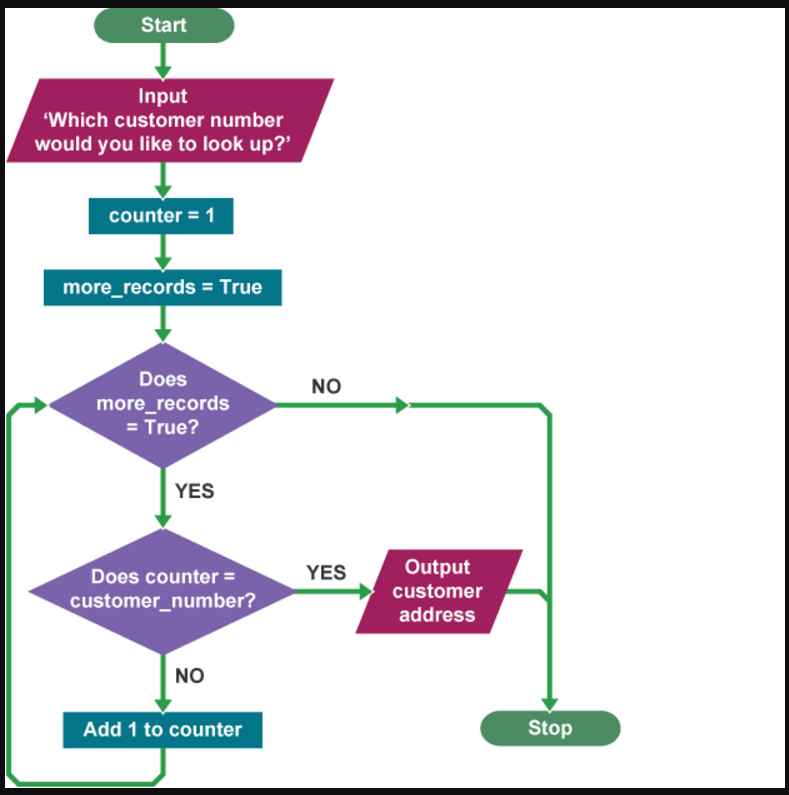
1. Quick Sort
2. procedure quickSort(left, right)
3. if right-left <= 0
4. return
5. else
6. pivot = A[right]
7. partition = partitionFunc(left, right, pivot)
8. quickSort(left,partition-1)
9. quickSort(partition+1,right)
10. end if
12. end procedure

**Task 5: Create the code for the following design. You can use code from other tasks to help.**

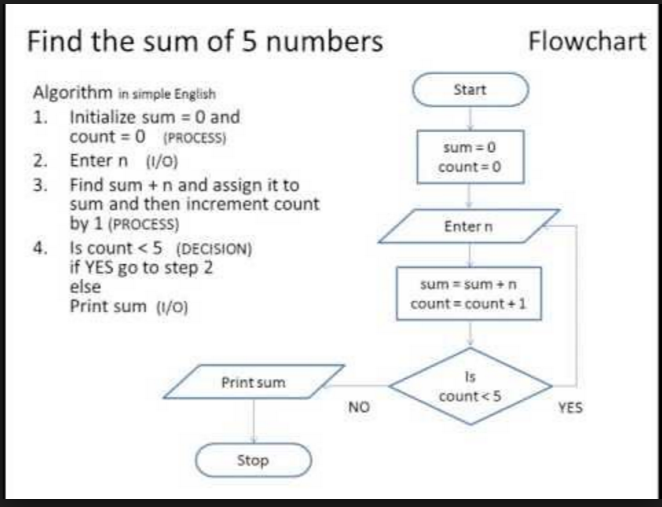


**Task 6:**

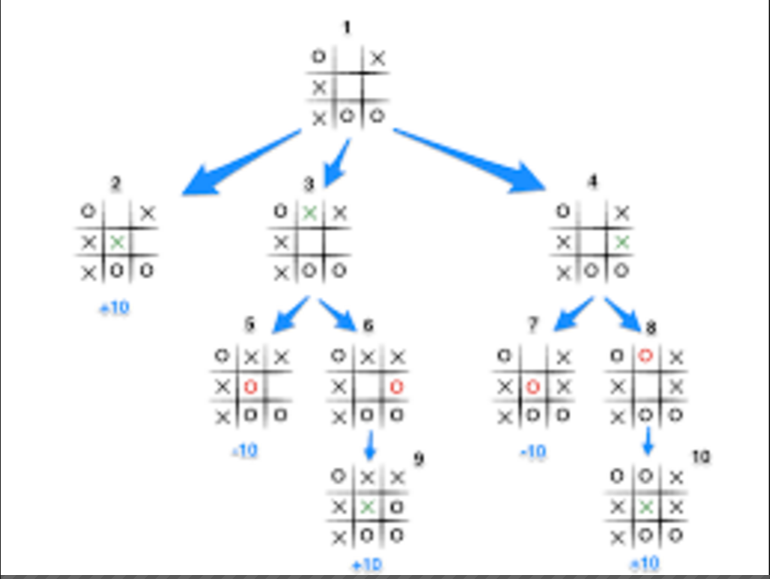
**Create a program for the following design:**



**Task 7:**



**Task 8:**



**Task 9:**

**Create a simple calculator using the following design, you do not need to add the advanced features but can as an extension.**

