**National University of Computer and Emerging Sciences**



Lab Manual # 13 Programming Fundamentals (Section BDS-1A)

|  |  |
| --- | --- |
| Course Instructor | Dr. Mubashir Baig |
| Lab Instructor(s) | Mr.Fraz  Ms.Samia |
| Section | BDS-1A |
| Semester | Fall 2021 |

Department of Computer Science FAST-NU, Lahore, Pakistan

**Q1.** Implement a function sumOfN that takes as input parameters an integer array data, an index number Ind, an integer N and size of the array and calculates the sum of N elements in the array starting from index Ind.

Sample:

data: 3 4 1 5 7 11 3 5 4

Ind = 4 N=3

Return: 21

How: sum = 7+11+3 = 21

data: 3 4 1 5 7 11 3 5 4

Ind = 7 N=3

Return: 9

How: sum = 5+4 +end of array = 9

**Q2.** Using the function sumOfN implement the function hasSum that takes as input parameters an array data, an integer total and the size of the array. The function returns true if the sum of any consecutive numbers in the array equals to total.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sample: |  | | | | | | | |
| data: **3**  total = 10 Return: true | **4** | **1** | **5** | **7** | **11** | **5** | **3** | **4** |

How: sum of 3 consecutive numbers 4, 1 and 5 = 10

data: **3 4 1 5 7 11 5 3 4**

total = 31 Return: true

How: sum of 5 consecutive numbers 5,7,11,5 and 3 = 31

data: **3 4 1 5 7 11 5 3 4**

total = 11 Return: true

How: sum of one slot is 11

data: **3 4 1 5 7 11 5 3 4**

total = 9 Return: false

How: Sum of 9 cannot be generated by 1 or more consecutive slots

**Q3.** The file named **inventory.txt** contains three types of (space separated) information [**Name, Quantity, price per unit**] about items available in a shop for sale. Assume the maximum number of products that the shopkeeper manages is 10. You are required to read these details

from the file whose format is: **name** in an array of cstrings, **quantity** in an integer array and **price per unit** in a float array. Show all the data to the customer and ask him what he wants to buy.

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Name** | **Quantity** | **Price per unit** |
| 1 | Apple | 25 | 3.5 |
| 2 | Orange | 20 | 5.7 |
| 3 | Banana | 50 | 2.5 |
| 4 | Papaya | 23 | 10 |
| 5 | Lychee | 35 | 1 |
| 6 | Olive | 56 | 2 |
| 7 | Strawberry | 125 | 3.5 |
| 8 | Raspberry | 18 | 1 |
| 9 | Date | 90 | 1.2 |
| 10 | Mango | 40 | 15 |

Ask the customer the code of the item and then quantity. Then display “Do you want to buy more items”. If the customer presses 1 then continue shopping by asking the customer the code and quantity of another item. If 0 is pressed then show the bill to the customer, write the **updated inventory** to the file following the format as shown in example below, and exit program.

Sample example: Suppose the customer buys 10 apples, and 5 strawberries. Then the following updated inventory must be stored on file.

# inventory.txt

Apple **15** 3.5

Orange 20 5.7

Banana 50 2.5

Papaya 23 10

Lychee 35 1

Olive 56 2

Strawberry **120** 3.5

Raspberry 18 1

Date 90 1.2

Mango 4 15

**Q4.** The program begins with the menu:

1. Encrypt data
2. Decrypt data

If the user presses 1, read the entire data from file “input.txt” into a char array. The file cannot contain more than 100 characters (including digits, spaces, letters etc.). Encrypt the file by replacing each character with a character that has the next ascii letter. For example a with b, M with N etc. Print the updated array on screen. Also store the array in file “encrypt.txt”.

# Sample input.txt

Najamsheraz, the singer, is an intelligent boy.

# Output in encrypt.txt

Obkbn!tifsb{-!uif!tjohfs-!jt!bo!joufmmjhfou!cpz/

If the user presses 2, read the entire data from file “encrypt.txt” into a char array. The file cannot contain more than 100 characters (including digits, spaces, letters etc.). Decrypt the file by replacing each character with a character that has the preceding ascii letter. For example b with a, N with M etc. Print the updated array on screen. Also store the array in file “decrypt.txt”.

# Sample data:

Input in encrypt.txt

Obkbn!tifsb{-!uif!tjohfs-!jt!bo!joufmmjhfou!cpz/

# Output in decrypt.txt

Najamsheraz, the singer, is an intelligent boy.