Dept. of CSE, Bennett University
ECSE379L – Programming Using C++

Lab Assignment – 7

In this lab, you will write a C++ program to solve the following problem.

(1) You need a picture frame of Amitabh Bacchan, so you walk down to the local photo store to examine their collection. They have all of their frames lined up against the wall. Apply the linear search algorithm to this problem, and describe how you would find the frame you wanted. Starting at the first frame, examine each frame along the wall (without skipping any) until you find the frame you want. Use the **pointer** for traversing the elements in char array.

Test Case 1:
Input:
Anil
Dev
Raj
Sanjeev
Amitabh
Dharmendra

Output:
Present at number 5

Test Case 2:
Input:
Ranvir
Arjun
Hritik
Abhishek
Akshay
Aditya

Output:
Not present

(2) Given two integers, find the described solution of the integers(use reference variable).

Write a function:
double * ProblemSolution :: solution(int N1, int N2)

that accepts two integers and return the **reference** to the variable holding:

a*20+34/b

Input:

5
10

Where,

- First line represents integer N1.
- Second line represents integer N2.

Output:
103.40

(3) We have three arrays:

a. Difficulty[i] is the difficulty of the ith problem
b. Marks[i] is the Marks of the ith problem.
c. Student[i] is the ability of the ith student.

The task is to figure out the maximum marks based on student's ability. Make use of pointers for traversing the values in array.

Note:
The student can only complete a problem with difficulty at most [i].

Every student can be assigned at most one problem, but one problem can be completed multiple times.

For example, if 5 students attempt the same question of 10 mark, then the total marks will be 30. If a student can not attempt any problem, his/her mark is 0.

Testcase 1
Input:

difficulty = [2,4,6,8,10], marks = [10,20,30,40,50], student = [4,5,6,7]

Output:
100

Explanation: Students are assigned problems of difficulty [4,4,6,6] and they get marks of [20,20,30,30] separately.

(4) Write a program to input an array of n integers (n<=100) and arrange them in ascending order. The subscripts of the elements vary from 0-(n-1).Make use of functions given below:

Functions/methords:
void read_array(int arr[],int l) //input array elements from the user
void arrange_array(int arr[],int l) //arrange the elements in ascending order
void display_array(int arr[],int l) //displays n integers **using pointers**
int binarysearch(int arr[],int l,int value) //searches for the value in the array using the binary search technique.

It returns the subscript of the array element if the value is found otherwise it returns -999.

Testcase
Input:
5
12 3 6 7 8
15

where:

First line represents the number of elements in the array.
Second line represents the elements in the array.
Third line represents the value of element to be searched.

Output:

-999

(Optional)

(5) Given an array of N numbers, partition the arrays into two subarrays such that none of the elements of the left subarray are in the right subarray. If there are multiple partitions possible, then partition the array in such a way that the absolute difference between the sum of left subarray and sum of the right subarray is minimum. Display the absolute difference between the sum of both subarrays.

Display -1 if no such partition is possible.

Input:

5
4 2 4 8 9

where:

First line represents the number of elements in the array.
Second line represents the elements in the array.

Output:

7

Explanation: Possible partitions are:{ 4, 2, 4 },{ 8, 9 } and { 4, 2, 4, 8 },{ 9 }

Out of the possible partitions { 4 ,2, 4 },{ 8, 9 } has least absolute difference i.e. |(4 + 2 + 4) - (8 + 9)| = 7, hence the output is 7.

Assumptions:

Array element can be in the range -1000 to 1000