Assignment 4

Introduction to programming in C

Question 1

Given an array of positive integers, write a C Program to output the sum of the elements that are above the mean.

Given an array $[a_1, a_2 \ldots a_n]$ the mean of the array is defined as $\mu = \frac{1}{n}\sum_{i=1}^{n} a_i$.

Input The first line contains the size of the array. The second line contains the contents of the array. Output Output the sum of elements in the array which are greater than or equal to the mean.

Note : The sizes of the arrays are smaller than 20.

Solution

```c
#include <stdio.h>

int main() {
    int size;
    int array[20];

    // Reading the input
    scanf(”%d”, &size);

    for(int i = 0; i < size; i++) {
        scanf(”%d”, &array[i]);
    }

    // Finding the sum

    int sum1 = 0;

    for(int i = 0; i < size; i++) {
        sum1 += array[i];
    }

  // Finding the mean

    float avg = (float) sum1/size;
    int sum2 = 0;

  // Finding the sum of elements above mean
    for(int i = 0; i < size; i++) {
        if(array[i] >= avg){
            sum2+= array[i];
        }
    }

    printf("%d",sum2);
}
```

Question 2

Given two arrays of positive integers, write a C Program to output the smallest number in the first array that is also present in the second one.

Input

The first line contains the size of the first array. The second line contains the contents of first array. The third line contains the size of the second array. The fourth line contains the contents of second array. Output

Output the smallest number occurring in the first array that also occurs in the second. In case there is no such number, output -1.

Note : The sizes of the arrays are smaller than 20.

Solution

```c
#include <stdio.h>

int main() {
    int size1, size2;
    scanf("%d", &size1);
    int arr1[20], arr2[20];

    // Reading first array
    for(int i = 0; i < size1; i++) {
        scanf("%d", &arr1[i]);
    }

    scanf("%d", &size2);
    // Reading second array
    for(int i = 0; i < size2; i++) {
        scanf("%d", &arr2[i]);
    }

    int found = -1; // to store the smallest number found in both arrays

    // Checking each element of the first array to see if it exists in the second array
    for(int i = 0; i < size1; i++) {
        int val = arr1[i];
        for(int j = 0; j < size2; j++) {
            if(arr2[j] == val) { // If the element from the first array is found in the second array
                if(found == -1 || val < found) { // If it's the first found or smaller than the previous found
                    found = val;
                }
            }
        }
    }

    printf("%d", found); // Printing the result
    return 0;
}
```

Question 3

Given two strings(character arrays), write a C Program to check if one of them is an anagram of the other.

An anagram is a word or phrase formed by rearranging the letters of a different word or phrase, using all the original letters exactly once.

Examples are LISTEN and SILENT , KNEE and KEEN.

Input

The first line contains the size of the character array. The second line contains the contents of first array. The third line contains the contents of the second array.

Note: The maximum size of the character array can be assumed to be 20. The contents of both arrays are only upper case alphabets.

Output

1 If the contents of the arrays are anagrams 0 If the contents of the arrays are not anagrams

Note : The sizes of the both the arrays can be assumed to be the same.

Solution

```c
#include <stdio.h>

int main() {
    int size;
    char s1[20],s2[20];
    int isanagram = 1;

    // Reading the input
    scanf("%d", &size);
    scanf("%s",s1);
    scanf("%s",s2);

  //flag variable to detect character is present in s2.
```

```
    int flag;

  // Loop through each element in s1
    for(int i = 0; i < size; i++) {
        flag = 0;                        // Set flag to 0
        for(int j = 0; j < size; j++){
            if(s2[j] == s1[i]){          // s1[i] found in s2.
                s2[j] = '0';             // set s2[j] to 0, so
    that its not counted again.
                flag = 1;
                break;
            }
        }
        if(flag == 0){                   // If match is not found,
     its not an anagram
            isanagram = 0;
        }
    }


    printf("%d", isanagram); // Printing the result
    return 0;
}
```