WEEK 5 :

/\*

NAME : NANDINI MAURYA

UNIVERSITY ROLL NUMBER : 2021320

SECTION : B

PROBLEM 1 : Given an unsorted array of alphabets containing duplicate elements. Design an algorithm and implement it using a program to find which alphabet has maximum number of occurrences and print it. (Time Complexity = O(n)) (Hint: Use counting sort) Input Format: The first line contains number of test cases, T. For each test case, there will be two input lines. First line contains n (the size of array). Second line contains space-separated integers describing array. Output: The output will have T number of lines. For each test case, output will be the array element which has maximum occurrences and its total number of occurrences. If no duplicates are present (i.e. all the elements occur only once), output should be “No Duplicates Present”.\*/

#include <bits/stdc++.h>

#include <vector>

#include<algorithm>

#include <fstream>

using namespace std;

void countingsort(vector<char>&arr)

{

int maxi=0;

for(int i=0;i<arr.size();i++)

{

maxi=max(maxi,arr[i]-'a');

}

vector<int>count(maxi+1,0);

for(int i=0;i<arr.size();i++)

{

count[arr[i]-'a']++;

}

int f=0,max\_count=0,ind;

for(int i=0;i<maxi;i++)

{

if(count[i]>max\_count && count[i]>1)

{

f=1;

max\_count=count[i];

ind=i;

}

}

if(f==0)

cout<<"No duplicates present"<<endl;

else

{

cout<<(char)(ind+97)<<" - "<<max\_count<<endl;

}

}

int main()

{

// Open input file

ifstream inputFile("C:\\Users\\user\\Desktop\\Lab questions\\codes\\C++\\input.txt");

// Check if the file is successfully opened

if (!inputFile.is\_open())

{

cerr << "Error opening the file!" << endl;

return 1;

}

int test;

inputFile >> test;

// Loop through each test case

while (test--)

{

int n, c = 0,s=0, key;

// Read the size of the array

inputFile >> n;

// Read the elements of the array directly into the vector

vector<char> nums(n);

for (int i = 0; i < n; i++)

{

inputFile >> nums[i];

}

countingsort(nums);

}

return 0;

}

\*\*\*\*\*\*\*\*\*INPUT \*\*\*\*\*\*\*\*\*

3

10

a e d w a d q a f p

15

r k p g v y u m q a d j c z e

20

g t l l t c w a w g l c w d s a a v c l

\*\*\*\*\*\*\*\*\*OUTPUT \*\*\*\*\*\*\*\*\*

A screenshot of a computer

Description automatically generated

/\*

NAME : NANDINI MAURYA

UNIVERSITY ROLL NUMBER : 2021320

SECTION : B

PROBLEM 2 : Given an unsorted array of integers, design an algorithm and implement it using a program to find whether two elements exist such that their sum is equal to the given key element. (Time Complexity = O(n log n)) Input Format: The first line contains number of test cases, T. For each test case, there will be two input lines. First line contains n (the size of array). Second line contains space-separated integers describing array. Third line contains key Output Format: The output will have T number of lines. For each test case, output will be the elements arr[i] and arr[j] such that arr[i]+arr[j] = key if exist otherwise print 'No Such Elements Exist”.\*/

#include <iostream>

#include <vector>

#include <algorithm>

#include <fstream>

using namespace std;

// Function to find two elements whose sum is equal to the given key

void findElements(vector<int>& arr, int n, int key) {

*// Sort the array*

    sort(arr.begin(), arr.end());

    // Initialize two pointers, one pointing to the beginning and the other pointing to the end of the array

    int left = 0, right = n - 1;

    // Traverse the array and find the elements whose sum is equal to the key

    while (left < right) {

        int sum = arr[left] + arr[right];

        if (sum == key) {

            cout << arr[left] << " " << arr[right] << endl;

            return;

        } else if (sum < key) {

            left++;

        } else {

            right--;

        }

    }

    // If no such elements exist, print "No Such Elements Exist"

    cout << "No Such Elements Exist" << endl;

}

int main() {

    // Open input file

    ifstream inputFile("C:\\Users\\user\\Desktop\\Lab questions\\codes\\C++\\input.txt");

    // Check if the file is successfully opened

    if (!inputFile.is\_open())

    {

        cerr << "Error opening the file!" << endl;

        return 1;

    }

    int test,key;

    inputFile >> test;

    // Loop through each test case

    while (test--)

    {

        int n, c = 0,s=0, key;

        // Read the size of the array

        inputFile >> n;

        // Read the elements of the array directly into the vector

        vector<int> nums(n);

        for (int i = 0; i < n; i++)

        {

            inputFile >> nums[i];

        }

        inputFile >> key;

        findElements(nums,nums.size(),key);

    }

    return 0;

}

\*\*\*\*\*\*\*\*\*\*\*INPUT\*\*\*\*\*\*\*\*\*\*\*\*

2

10

64 28 97 40 12 72 84 24 38 10

50

15

56 10 72 91 29 3 41 45 61 20 11 39 9 12 94

302

\*\*\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*

A computer screen shot

Description automatically generated

/\*

NAME : NANDINI MAURYA

UNIVERSITY ROLL NUMBER : 2021320

SECTION : B

PROBLEM 3 :You have been given two sorted integer arrays of size m and n. Design an algorithm and implement it using a program to find list of elements which are common to both. (Time Complexity = O(m+n))

Input Format: First line contains m (the size of first array). Second line contains m space-separated integers describing first array. Third line contains n (the size of second array). Fourth line contains n space-separated integers describing second array.

Output Format: Output will be the list of elements which are common to both\*/

#include <iostream>

#include <vector>

#include <fstream>

using namespace std;

// Function to find common elements in two sorted arrays

void findCommonElements(vector<int> arr1, int m, vector<int> arr2, int n) {

vector<int> commonElements;

int i = 0, j = 0;

// Traverse both arrays and compare elements

while (i < m && j < n) {

if (arr1[i] == arr2[j]) {

commonElements.push\_back(arr1[i]);

i++;

j++;

} else if (arr1[i] < arr2[j]) {

i++;

} else {

j++;

}

}

// Print the common elements

for (int elem : commonElements) {

cout << elem << " ";

}

}

int main() {

// Open input file

ifstream inputFile("C:\\Users\\devv2\\Desktop\\abba jabba\\weeks\\week5\\input.txt");

// Check if the file is successfully opened

if (!inputFile.is\_open())

{

cerr << "Error opening the file!" << endl;

return 1;

}

int test, key;

inputFile >> test;

int testCaseNumber = test - 1; // Initialize the test case number

// Loop through each test case

while (test--)

{

int n, m;

// Read the size of the array

inputFile >> n;

// Read the elements of the array directly into the vector

vector<int> nums1(n);

for (int i = 0; i < n; i++)

{

inputFile >> nums1[i];

}

inputFile >> m;

// Read the elements of the array directly into the vector

vector<int> nums2(m);

for (int i = 0; i < m; i++)

{

inputFile >> nums2[i];

}

cout << "Common elements in test case " << testCaseNumber - test << ": ";

findCommonElements(nums1, n, nums2, m);

cout << endl;

}

return 0;

}

\*\*\*\*\*\*\*\*\*\*\*\*INPUT \*\*\*\*\*\*\*\*\*\*\*\*

3

6

2 3 4 6 8 9

5

8 9 10 11 12

5

2 4 6 7 9

5

1 6 7 8 9

8

23 31 45 51 63 65 78 99

7

33 44 45 63 78 99 100

\*\*\*\*\*\*\*\*\*\*\*\*OUTPUT \*\*\*\*\*\*\*\*\*\*\*\*

A computer screen shot

Description automatically generated