DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

WEEK 5

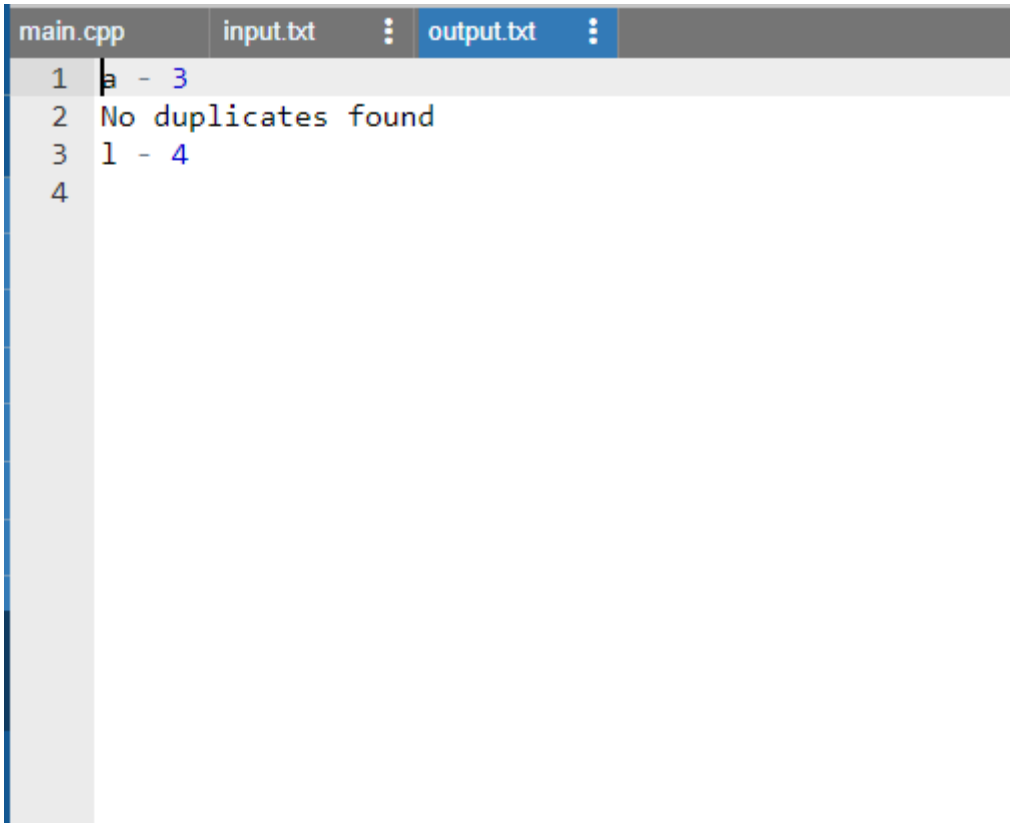
1. Given an unsorted array of alphabets containing duplicate elements. Design an algorithm and implement it using a program to find which alphabet has maximum number of occurrences andprint it. (Time Complexity = O(n)) (Hint: Use counting sort)

```
#include <iostream>
#include <vector>
#include <fstream>
#include<queue>
using namespace std;



                                                    //function to find duplicates
void largestfreq(vector<char>& arr,int n,ofstream &fout) {
   vector<int> res(26, 0);
   for(int i = 0; i < n; i++) {
      res[arr[i] - 'a']++;
   }
   int max = -1;
   char c = '\0';
   for(int i = 0; i < 26; i++) {
      if(res[i] > max) {
         max = res[i];
         c = i + 'a';
      }
   }
   if(max != 1) {
     fout<<c<<" - "<<max<<endl;
   }
   else {
      fout<<"No duplicates found"<<endl;
   }

}
```

```
int main() {
                                                        // Open input and output files
    ifstream fin("input.txt");
    ofstream fout("output.txt");
    // Check if files are opened successfully
    if (!fin.is_open() || !fout.is_open()) {
        cout << "Error occurred while opening files.\n";
        return 0;
    }
    int t;
                                                        // Number of test cases
    fin >> t;
    while (t--) {

        int n;
        fin >> n;
        vector<char> arr(n);

                                                        // Input array elements
        for (int i = 0; i < n; i++) {
            fin >> arr[i];
        }

                                                        // finding duplicates frequency
        largestfreq(arr,n,fout);
    }
    // Close input and output files
    fin.close();
    fout.close();
}
```

main.cpp | input.txt | output.txt

```
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
```


main.cpp
input.txt
output.txt
a - 3
No duplicates found
1 - 4

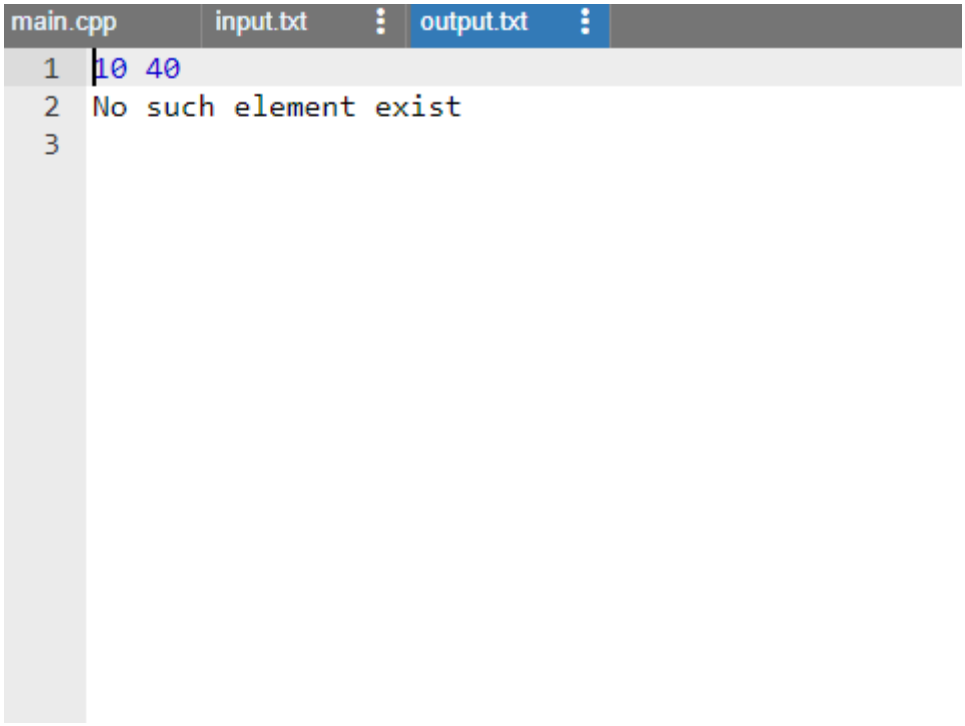
2 . Given an unsorted array of integers, design an algorithm and implement it using a program to find whether two elements exist such that their sum is equal to the given key element. (Time Complexity = O(n log n)).

```
#include <iostream>
#include <vector>
#include <fstream>
#include<queue>
using namespace std;



                                                    //function to find pairs
void pairSum(vector<int>& arr,int n,int k,ofstream &fout) {
   sort(arr.begin(), arr.end());
   vector<int> sol;
   int i = 0, j = n - 1;
   while(i < j) {
      if(arr[i] + arr[j] == k) {
         sol.push_back(arr[i]);
         sol.push_back(arr[j]);
         break;
      }
      if(arr[i] + arr[j] > k) j--;
      else i++;
   }
   if(sol.empty())
     fout<<"No such element exist";
   else {
      for(auto it : sol)
        fout<<it<<" ";
   }
   fout<<endl;
}
```

```
int main() {
                                                        // Open input and output files
    ifstream fin("input.txt");
    ofstream fout("output.txt");
    // Check if files are opened successfully
    if (!fin.is_open() || !fout.is_open()) {
        cout << "Error occurred while opening files.\n";
        return 0;
    }
    int t;
                                                        // Number of test cases
    fin >> t;
    while (t--) {

        int n;
        fin >> n;
        vector<int> arr(n);

                                                        // Input array elements
        for (int i = 0; i < n; i++) {
            fin >> arr[i];
        }
        int k;
        fin>>k;
                                                        // finding duplicates frequency
        pairSum(arr,n,k,fout);
    }
    // Close input and output files
    fin.close();
    fout.close();
}
```

main.cpp | input.txt | output.txt

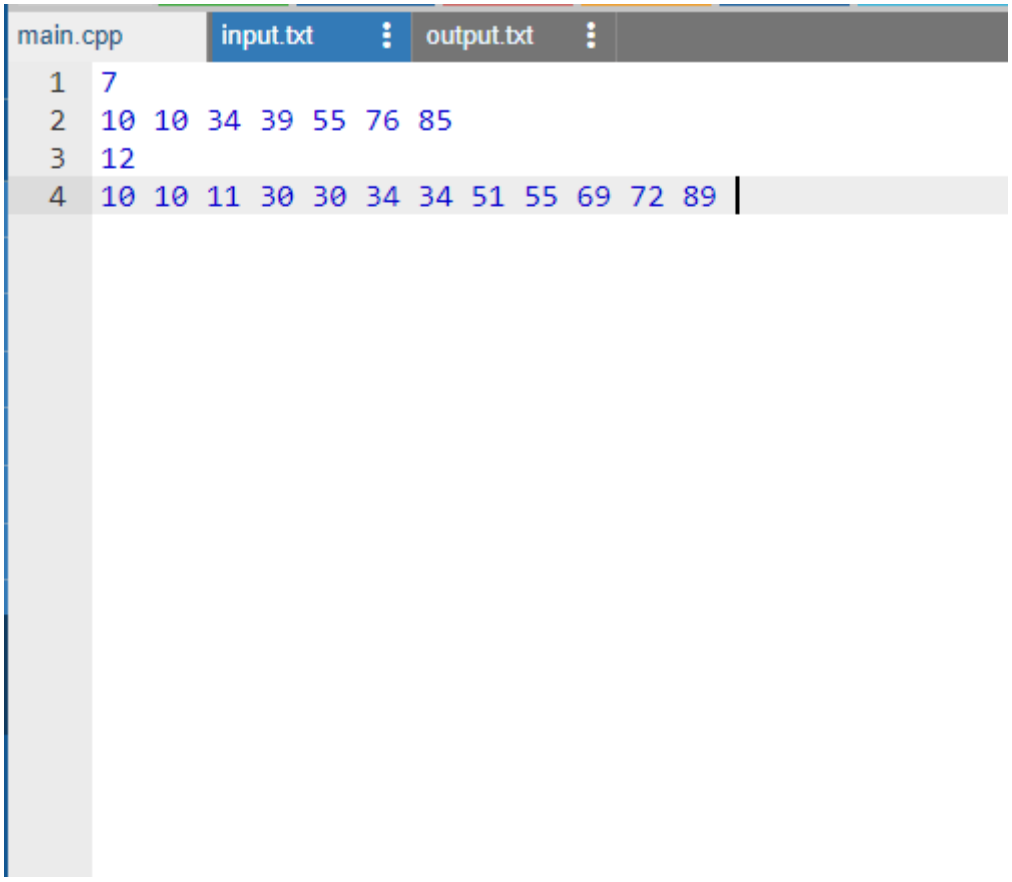
```
2
10
64 28 97 40 12 72 84 24 38 10
50
15
56 10 72 91 29 3 41 45 61 20 11 39 9 12 94
302
```


main.cpp
input.txt
output.txt
1 10 40
2 No such element exist
3

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))

```
#include <iostream>
#include <vector>
#include <fstream>
#include<queue>
using namespace std;



                                                    //function to find common element
void common(vector<int>& arr,int n,vector<int>&brr,int m,ofstream &fout) {
   int i = 0, j = 0;
   while(i < n && j < m) {
      if(arr[i] == brr[j]) {
         fout<<arr[i]<<" ";
         i++; j++;
      }
      else if(arr[i] < brr[j]) i++;
      else j++;
   }
}
```

```
int main() {
    // Open input and output files
    ifstream fin("input.txt");
    ofstream fout("output.txt");
    // Check if files are opened successfully
    if (!fin.is_open() || !fout.is_open()) {
        cout << "Error occurred while opening files.\n";
        return 0;
    }
    int n;
    fin >> n;
    vector<int> arr(n);
                                                        // Input array elements
    for (int i = 0; i < n; i++) {
        fin >> arr[i];
    }
    int m;
    fin >> m;
    vector<int> brr(m);

                                                        // Input array elements
    for (int i = 0; i < m; i++) {
        fin >> brr[i];
    }
                                                    // finding common elements
    common(arr,n,brr,m,fout);

                                                   // Close input and output files
    fin.close();
    fout.close();
}
```


main.cpp
input.txt
output.txt
1 7
2 10 10 34 39 55 76 85
3 12
4 10 10 11 30 30 34 34 51 55 69 72 89


main.cpp
input.txt
output.txt
1
10 10 34 55