**Assignment CPP**

**SET 1: - Question 1, Question 4, Question 6.**

**Q1.** Vectors: Write a C++ program, there you take two vectors of names. The program should return a vector that has all the names only once.

Example:

vector1 = [Virendra Sachin, Saurav, Rahul, Laxman]

vector2 = [Harbhajan, Ashish, Rahul, Yuvraj, Saurav]

output = [Virendra Sachin, Saurav, Rahul, Laxman, Harbhajan, Ashish, Yuvraj]

**code:**

#include <string>

#include <iostream>

#include <vector>

#include <algorithm> //added in order to use count and sort

using namespace std;

int main()

{

string str1;

vector<string> list1;

vector<string> list2;

vector<string> list3;

cout<<"\nEnter first list with 5 element's"<<endl;

for(int i=0;i<5;i++){

cin >> str1;

list1.push\_back(str1);

}

cout<<"\nEnter second list with 5 element's"<<endl;

for(int i=0;i<5;i++){

cin >> str1;

list2.push\_back(str1);

}

cout<<"\nshowing first list with 5 element's"<<endl;

for (auto i = list1.begin(); i != list1.end(); ++i){

cout << \*i << " ";}

cout<<endl;

cout<<"\nshowing second list with 5 element's"<<endl;

for (auto i = list2.begin(); i != list2.end(); ++i){

cout << \*i << " ";}

cout<<endl;

// adding 1st vector to list3

for (auto i = list1.begin(); i != list1.end(); ++i){

list3.push\_back(\*i);

}

// check if list2 elements are not in list3 then only add it to list3

for (auto j = list2.begin(); j != list2.end(); ++j){

if(!count(list3.begin(), list3.end(), \*j)){

list3.push\_back(\*j);

}

}

sort(list3.begin(), list3.end()); //sort the final union vector

cout<<"\nfinal merge vector without duplicate is"<<endl;

for (auto i = list3.begin(); i != list3.end(); ++i){

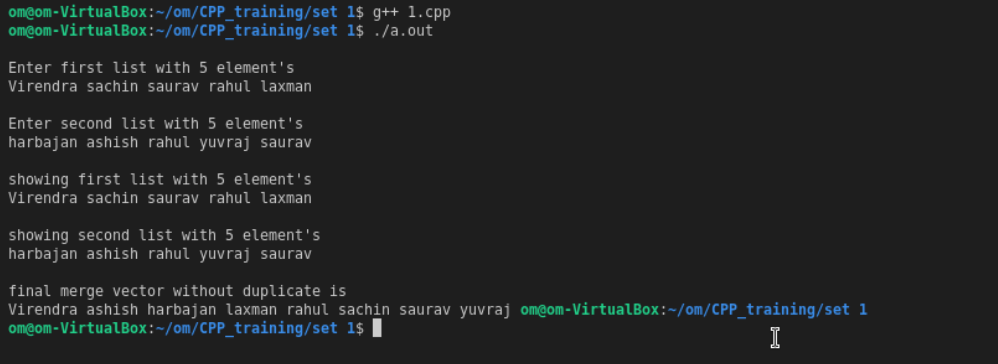
cout << \*i << " ";

}

return 0;

}

**Output:**

****

**Q4.** In this question, you are given a binary string of length T. Now you need to create two permutations of this string: S1 and S2 such that the ‘longest common subsequence’ between the two newly created strings is smallest.

**Code:**

#include <string.h>

#include <iostream>

using namespace std;

int j = 0, r = 0;

int arrstore[] = {};

string arr[] = {};

void get\_minimum(int arr[], int N)

{

int min = arr[0];

for (int i = 1; i < N; i++)

{

if (min > arr[i])

min = arr[i];

}

cout << "\nThe Final ans : The smallest LCS is: " << min;

}

bool swaper(char str[], int start, int curr)

{

for (int i = start; i < curr; i++)

if (str[i] == str[curr])

return 0;

return 1;

}

void printPermutations(char str[], int index, int n)

{

if (index >= n)

{

cout << str << "\t";

arr[j] = str;

j++;

return;

}

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

{

bool check = swaper(str, index, i);

if (check)

{

swap(str[index], str[i]);

printPermutations(str, index + 1, n);

swap(str[index], str[i]);

}

}

}

int lowest\_common\_substring(string X, string Y, int m, int n)

{

int lcsuff[m + 1][n + 1];

int result = 0; // To store length of the

// longest common substring

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

{

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

{

if (i == 0 || j == 0)

lcsuff[i][j] = 0;

else if (X[i - 1] == Y[j - 1])

{

lcsuff[i][j] = lcsuff[i - 1][j - 1] + 1;

result = max(result, lcsuff[i][j]);

}

else

lcsuff[i][j] = 0;

}

}

arrstore[r] = result;

r++;

return result;

}

int main()

{

char string[20];

int n, k, i, l, m;

cout << "\nEnter any string :: ";

cin >> string;

int s = strlen(string);

n = strlen(string);

printPermutations(string, 0, n);

for (i = 0; i < j; i++)

{

cout << arr[i];

}

if (j == 2)

{

cout << "\nThe result is 0";

}

else

{

for (i = 0; i < j; i++)

{

for (k = i + 1; k < j; k++)

{

cout << "\nThe length of the LCS is " << lowest\_common\_substring(arr[i], arr[k], s, s);

}

}

}

get\_minimum(arrstore, r);

return 0;

}

**Output:**

Text

Description automatically generated

**Q6.** : The grade of a student can be based on the following criteria: Attendance must be more than 50% Tests scores must be more than .70 Total score must be more than 5600 (just go with this number) The grades are awarded as follows:

Grade is 10: if all conditions are met

Grade is 9: if conditions 1 and 2 are met

Grade is 8: if conditions 3 and 3 are met

Grade is 7: if conditions 1 and 3 are met

Grade is 6: if only one condition is met

Grade is 5: if none of the conditions are met

Write a cpp program to display the grades, based on the scores students secured in individual criteria. Ex: input 53, 0.6, 5602 res: 10 Input 45, 0, 4500 res: 6

**Code:**

#include <string>

#include <iostream>

#include <vector>

#include<string.h>

#include <bits/stdc++.h>

using namespace std;

int main()

{

int atendance,total\_score;

float test\_score;

cout<<"atendance in precentage"<<endl;

cin>>atendance;

cout<<"test\_score"<<endl;

cin>>test\_score;

cout<<"total\_score"<<endl;

cin>>total\_score;

if(atendance>=50 && test\_score >= 0.70 && total\_score >= 5600){

cout<<"input = "<<atendance<<","<<test\_score<<","<<total\_score<<","<<endl;

cout<<"res = 10";

}

else if(atendance>=50 && test\_score >= 0.70 && total\_score <= 5600){

cout<<"input"<<atendance<<","<<test\_score<<","<<total\_score<<","<<endl;

cout<<"res = 9";

}

else if(atendance<=50 && test\_score <= 0.70 && total\_score >= 5600){

cout<<"input"<<atendance<<","<<test\_score<<","<<total\_score<<","<<endl;

cout<<"res = 8";

}

else if(atendance>=50 && test\_score <= 0.70 && total\_score >= 5600){

cout<<"input"<<atendance<<","<<test\_score<<","<<total\_score<<","<<endl;

cout<<"res = 7";

}

else if(atendance>=50 || test\_score >= 0.70 || total\_score >= 5600){

cout<<"input"<<atendance<<","<<test\_score<<","<<total\_score<<","<<endl;

cout<<"res = 6";

}

else{

cout<<"input"<<atendance<<","<<test\_score<<","<<total\_score<<","<<endl;

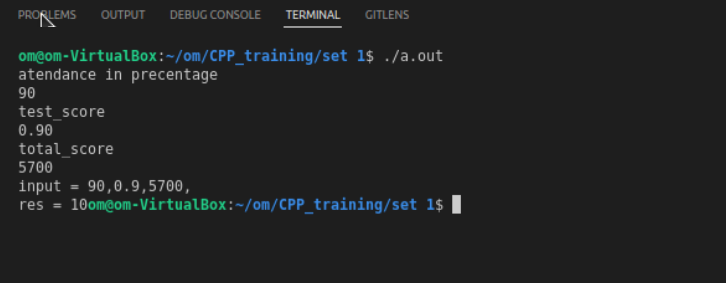
cout<<"res = 5";

}

return 0;

}

**Output:**

****