**3. An Anagram of a string is another string that contains same characters, only the order of characters can be different.**

#include <bits/stdc++.h>

using namespace std;

int anagram(string s1,string s2){

int array1[26]={0},array2[26]={0};

if(s1.length()!=s2.length())

return 0;

for(int i=0;s1[i]!='\0';i++){

array1[s1[i]-'a']++;

}

for(int i=0;s2[i]!='\0';i++){

array2[s2[i]-'a']++;

}

for(int i=0;i<26;i++){

if(array1[i]!=array2[i])

return 0;

}

return 1;

}

int main()

{

int n;

string s1,s2;

cout<<"enter string1\n";

cin>>s1;

cout<<"enter string2\n";

cin>>s2;

if(anagram(s1,s2))

printf("strings are anagrams of each other\n");

else

printf("strings are not anagrams of each other\n");

return 0;

}

**OUTPUT :**

**C CODE :**

#include <stdio.h>

#include <string.h>

int main (void) {

char s1[100];

char s2[100];

char temp;

printf("enter String 1");

scanf("%s",s1);

printf("enter String 2");

scanf("%s",s2);

int i, j;

int n = strlen(s1);

int n1 = strlen(s2);

// If both strings are of different length, then they are not anagrams

if( n != n1) {

printf("%s and %s are not anagrams! \n", s1, s2);

return 0;

}

// lets sort both strings first −

for (i = 0; i < n-1; i++) {

for (j = i+1; j < n; j++) {

if (s1[i] > s1[j]) {

temp = s1[i];

s1[i] = s1[j];

s1[j] = temp;

}

if (s2[i] > s2[j]) {

temp = s2[i];

s2[i] = s2[j];

s2[j] = temp;

}

}

}

// Compare both strings character by character

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

if(s1[i] != s2[i]) {

printf("Strings are not anagrams! \n", s1, s2);

return 0;

}

}

printf("Strings are anagrams! \n");

return 0;

}