**1.  Given two strings s and t, return true if t is an anagram of s, and false otherwise.**

**CODE :**

#include <stdio.h>

#include <string.h>

int main() {

char s[100], t[100];

printf("Enter the first string: ");

scanf("%s", s);

printf("Enter the second string: ");

scanf("%s", t);

int count[256] = {0};

int len\_s = strlen(s);

int len\_t = strlen(t);

if (len\_s != len\_t) {

printf("Output: false\n");

return 0;

}

for (int i = 0; i < len\_s; i++) {

count[(int)s[i]]++;

count[(int)t[i]]--;

}

for (int i = 0; i < 256; i++) {

if (count[i] != 0) {

printf("Output: false\n");

return 0;

}

}

printf("Output: true\n");

return 0;

}

**2. Write a function to find the longest common prefix string amongst an array of strings. If there is no common prefix, return an empty string "".**

**CODE :**

#include <stdio.h>

#include <string.h>

int main() {

int n;

printf("Enter the number of strings: ");

scanf("%d", &n);

char strs[n][100];

printf("Enter the strings: \n");

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

scanf("%s", strs[i]);

}

if (n <= 0) {

printf("Output: \"\"\n");

return 0;

}

int len = strlen(strs[0]);

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

int j = 0;

while (j < len && strs[0][j] == strs[i][j]) {

j++;

}

len = j;

}

char result[100];

strncpy(result, strs[0], len);

result[len] = '\0';

printf("Output: %s\n", result);

return 0;

}