|  |
| --- |
| /\*Write a C Program to check whether two strings are anagram or not. |
|  | Hint: An Anagram of a string is another string that contains same characters, only the order of characters can be different. |
|  | For example, "act" and "cat" are anagram of each other.\*/ |
|  |  |
|  | #include <stdio.h> |
|  | int check\_anagram(char [], char []); |
|  | int main() |
|  | { |
|  | char a[100], b[100]; |
|  | printf("Enter two strings : \n"); |
|  | gets(a); |
|  | gets(b); |
|  |  |
|  | if (check\_anagram(a, b) == 1) |
|  | printf("%s and %s strings are anagrams\n",a,b); |
|  | else |
|  | printf("%s and %s strings are not anagrams\n"); |
|  |  |
|  | return 0; |
|  | } |
|  | int check\_anagram(char a[], char b[]) |
|  | { |
|  | int first[26] = {0}, second[26] = {0}, c=0; |
|  | while (a[c] != '\0') |
|  | { |
|  | first[a[c]-'a']++; |
|  | c++; |
|  | } |
|  | c = 0; |
|  | while (b[c] != '\0') |
|  | { |
|  | second[b[c]-'a']++; |
|  | c++; |
|  | } |
|  | for (c = 0; c < 26; c++) |
|  | { |
|  | if (first[c] != second[c]) |
|  | return 0; |
|  | } |
|  | return 1; |
|  | } |