<https://www.sanfoundry.com/cpp-program-find-number-permutations-string/>

<https://www.techiedelight.com/find-lexicographic-permutations-string/>

**Mô tả vấn đề**

1. Thuật toán này in ra tổng số hoán vị có thể có cho một chuỗi nhất định.  
2. Độ phức tạp về thời gian của thuật toán này là O (n).

**Giải pháp vấn đề**

1. Thuật toán này lấy đầu vào của chuỗi.  
2. Sau đó, nó kiểm tra sự lặp lại của các ký tự.  
3. Tính toán hoán vị và in kết quả.  
4. Thoát.

**Chương trình / Mã nguồn**

C ++ Chương trình tìm số hoán vị của một chuỗi đã cho.  
Chương trình này được chạy thành công trên Dev-C ++ sử dụng trình biên dịch TDM-GCC 4.9.2 MinGW trên hệ thống Windows.

#include <iostream>

#include <string.h>

sử dụng không gian tên std ;

// Một hàm để tìm giai thừa.

int giai thừa ( int n )

{

int i ;

for ( i = n - 1 ; i > 1 ; i - )

n \* = i ;

trả về n ;

}

// Một hàm tính tổng số hoán vị có thể có cho tập hợp đã cho.

int CountPermutation ( char \* str )

{

int countoccur [ 26 ] = { 0 } , len, i, res ;

len = strlen ( str ) ;

// Đếm sự xuất hiện của mỗi ký tự.

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

{

countoccur [ str [ i ] - 'a' ] ++ ;

}

res = giai thừa ( len ) ;

// Chia giai thừa độ dài cho giai thừa của số lần xuất hiện của mỗi ký tự.

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

{

if ( countoccur [ i ] > 1 )

res = res / factorial ( countoccur [ i ] ) ;

}

trả lại res ;

}

int main ( )

{

int result ;

char str [ 100 ] ;

cout << "Chương trình tìm hoán vị của một chuỗi đã cho:" ;

cout << " **\ n \ n \ t** Nhập chuỗi:" ;

cin >> str ;

// Lấy kết quả bằng cách sử dụng CountPermutation ().

kết quả = CountPermutaion ( str ) ;

cout << " **\ n** Số hoán vị có thể có là:" << kết quả ;

trả về 0 ;

}

**Giải thích chương trình**

1. Lấy đầu vào của chuỗi.  
2. Gọi CountPermutation () với chuỗi làm đối số.  
3. Bên trong CountPermutation, đầu tiên là Đếm số lần xuất hiện của mỗi ký tự trong chuỗi.  
4. Tính giai thừa của độ dài của chuỗi và lưu trữ trong 'res'.  
5. Chia 'res' cho giai thừa của một số lần xuất hiện của mỗi ký tự nếu chúng đếm nhiều hơn 1.  
6. Quay lại hàm main và in kết quả.  
7. Thoát.

**Các trường hợp kiểm tra thời gian chạy**

Trường hợp 1:

Một chương trình để tìm một hoán vị của một chuỗi đã cho:

Nhập chuỗi: aabcddd

Số hoán vị có thể có là: 420

Trường hợp 2:

Một chương trình để tìm một hoán vị của một chuỗi đã cho:

Nhập chuỗi: vghhnhkkkvv

Số hoán vị có thể có là: 184800