(Ghi chú)

Ngôn ngữ lập trình cpp

**Thực hiện: Nguyễn Văn Hào**

**Email:** [hao2205tb@gmail.com](mailto:hao2205tb@gmail.com)

(Tài liệu được biên xoạn lại từ nhiều nguồn và không dùng trong mục đích kinh doanh kiếm tiền)

# Mục lục

[Mục lục 1](#_Toc126692043)

[Tham khảo 3](#_Toc126692044)

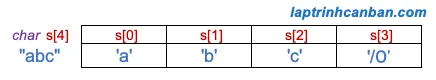
# Chuỗi

## Khái niêm

Chuỗi là tập hợp của nhiều kí tự và được viết trong dấu ngặc kép “”.

VD biểu diễn của chuỗi “abc”

"abc" = 'a' + 'b' + 'c' + '\0'



## Khai báo và Khởi tạo chuỗi

VD

char s[4];

char s[5] = "love";

char s[] = "love";

## in chuỗi

VD

#include <iostream>

using namespace std;

int main(){

  char str1[] = "C";

  char str2[] = "C#";

  char str3[] = "C++";

  cout << str1 <<' ' << str2 <<' ' << str3;

  return 0;

}

// C C# C++

## Strlen(): tính độ dài chuỗi

Hàm strlen() sẽ trả về độ dài chuỗi ký tự được gán trong chuỗi. Lưu ý rằng đây là độ dài (số ký tự) của chuỗi giá trị gán trong chuỗi, do đó nó sẽ không bao gồm cả ký tự kết thúc chuỗi \0 khi tính độ dài.

Lưu ý là chúng ta cần phải viết thêm dòng #include <cstring> để load header file cstring để có thể sử dụng được hàm strlen() trong chuơng trình.

VD:

#include <iostream>

#include <cstring>

using namespace std;

int main()

{

    char str[] = "I Love C/C++";

    int len;

    len = strlen(str);

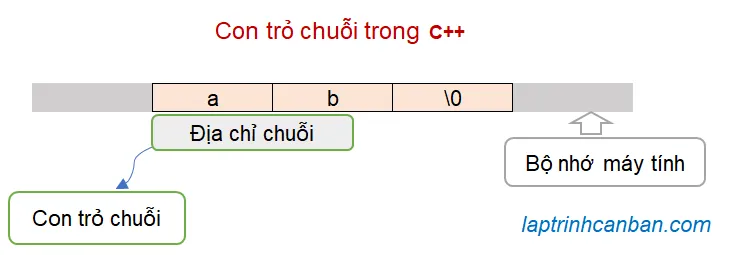
    cout << "Length of "<< str << " is " << len;

    return 0;

}

//Length of "I Love C/C++" is 12.

## Con trỏ chuỗi



### Khai báo

char a[] = "Hello", \*p1;

p1 = a;

char b[100], \*p2;

p2 = b;

### Truy suất các ký tự bằng con trỏ

Cú pháp:

\*(p + index);

| **index** | **Truy xuất bằng chuỗi** | **Truy xuất bằng con trỏ** |
| --- | --- | --- |
| 0 | a[0] | \*p |
| 1 | a[1] | \*(p + 1) |
| 2 | a[2] | \*(p + 2) |
| 3 | a[3] | \*(p + 3) |
| … | … | … |
| n | a[n] | \*(p + n) |

VD

#include <iostream>

using namespace std;

int main(){

    char a[] = "Hello", \*p;

    p = a;

    cout <<  a[2] <<endl;

    cout <<  \*(p +2)<<endl ;

    cout << a[4]<<endl;

    cout <<  \*(p +4)<< endl;

    return 0;

}

l

l

o

o

### Nhập chuỗi bằng con trỏ

char str[100], \*p;

p = str;

cin >> p;

### Xuất chuỗi bằng con trỏ

char str[100], \*p;

p = str;

while(\*p != '\0'){

    cout << \*p;

    ++p;

}

## Chèn chuỗi

Ví dụ như chèn "xyz" vào giữa chuỗi "abcdef" ở vị trí thứ 2

### Cách 1: dịch chuyển phần sau vị trí chèn ra sau và chép chuỗi vào vị trí chèn

* Bắt đầu từ vị trí cần chèn trong chuỗi ban đầu, chúng ta dịch chuyển dần về phía sau lượng ký tự bằng đúng với số ký tự của chuỗi cần chèn và tạo ra các ô trống.
* Sao chép chuỗi ký tự cần chèn để chèn vào các ô trống vừa tạo.

VD

#include <iostream>

#include <cassert>

#include <cstring>

using namespace std;

#define SIZE\_OF\_ARRAY(array)    (sizeof(array)/sizeof(array[0]))

/\*

    Chèn chuỗi trong C++

    Tham số

        str1:      Chuỗi ban đầu (không chấp nhận Null)

        str1\_size: Độ dài chuỗi str1

        pos:       Vị trí chèn (tính từ đầu chuỗi str1)

        str2:      Chuỗi chèn vào (không chấp nhận Null pointer)

    Giá trị trả về

        str1

\*/

char\* str\_insert(char\* str1, size\_t str1\_size, size\_t pos, const char\* str2)

{

    assert(str1 != NULL);

    assert(str2 != NULL);

    size\_t len1 = strlen(str1);

    size\_t len2 = strlen(str2);

    // Kiểm tra str1 có đủ độ dài để chèn str2 không

    assert(len1 + len2 < str1\_size);

    // Kiểm tra vị trí chèn có nằm trong chuỗi str1 không

    assert(len1 >= pos);

    // Làm trống một phạm vi dài với len2 ký tự, từ  str1[pos])

    // Dịch chuyển chuỗi ban đầu về sau, bao gồm cả ký tự `\0'

    memmove(&str1[pos + len2], &str1[pos], len1 - pos + 1);

    // Copy str2 và dán vào khoảng trống mới tạo

    memcpy(&str1[pos], str2, len2);

    return str1;

}

int main()

{

    char str[10] = "abcde";

    cout <<"before: "<< str <<endl;

    cout<< "after: "<<str\_insert(str, SIZE\_OF\_ARRAY(str), 2, "xyz");

    return 0;

}

### Cách 2: cấp phát bộ nhớ động và chép từng phần của chuỗi đích vào bộ nhớ.

#include <iostream>

#include <cassert>

#include <cstdlib>

#include <cstring>

using namespace std;

#define SIZE\_OF\_ARRAY(array)    (sizeof(array)/sizeof(array[0]))

/\*

    Chèn chuỗi trong C++

    Tham số

        str1:      Chuỗi ban đầu (không chấp nhận Null)

        pos:       Vị trí chèn (tính từ đầu chuỗi str1)

        str2:      Chuỗi chèn vào (không chấp nhận Null pointer)

    Giá trị trả về

        Con trỏ mảng chứa kết quả. Nếu thực thi thất bại thì trả về Null

\*/

char\* str\_alloc\_and\_insert(const char\* str1, size\_t pos, const char\* str2)

{

    assert(str1 != NULL);

    assert(str2 != NULL);

    size\_t len1 = strlen(str1);

    size\_t len2 = strlen(str2);

    // Kiểm tra vị trí chèn có nằm trong chuỗi str1 không

    assert(len1 >= pos);

    size\_t result\_size = len1 + len2 + 1;

    char\* result = (char\*) malloc(sizeof(char) \*result\_size);

    if (result == NULL) {

        return NULL;

    }

    // Copy phạm vi từ đầu chuỗi ban đầu tới trước vị trí cần chèn vào chuỗi kết quả

    memcpy(result, str1, pos);

    // Copy chuỗi cần chèn vào cuối kết quả

    memcpy(&result[pos], str2, len2);

    // Copy phạm vi từ vị trí cần chèn tới cuối chuỗi ban đầu vào chuỗi kết quả

    // Thêm ký tự kết thúc chuỗi `\0` vào kết quả

    strcpy(&result[pos + len2], &str1[pos]);

    return result;

}

int main()

{

    char str[] = "abcde";

    cout <<"before: "<< str <<endl;

    char\* result = str\_alloc\_and\_insert(str, 2, "xyz");

    if (result != NULL) {

        cout <<"after: "<< result;

        free(result);

    }

    return 0;

}

## Copy chuỗi

### Strcpy(): copy toàn bộ chuỗi

Cú pháp:

strcpy(target, source);

Trong đó:

* targe là chuỗi đích
* source là chuỗi nguồn.

Hàm strcpy() sẽ copy toàn bộ nội dung của chuỗi source và phục chế nó vào chuỗi target. Lưu ý là hai chuỗi này cần phải tồn tại trước đó, thông qua việc khởi tạo hay khai báo chuỗi trong C++.

VD

#include <iostream>

#include <cstring>

using namespace std;

int main(){

    char str1[30];

    char str2[30] = "I love C/C++";

    cout << "str2: " << str2 <<endl;

    //Copy toàn bộ chuỗi str2 vào chuỗi str1

    strcpy(str1, str2);

    cout << "str1: " << str1;

}

### Strcpy\_c(): copy toàn bộ chuỗi thay thế cho strcpy() ở một số trình biên dịch.

Khi tiến hành compile chương trình chứa hàm strcpy() thì lỗi compile sẽ xảy ra, và có thêm màn hình hướng dẫn chúng ta hãy sử dụng tới hàm strcpy\_c() để thay thế.

Lý do cũng dễ hiểu, bởi việc copy chuỗi bằng hàm strcpy() có một nhược điểm là nếu kích thước chuỗi cần copy lớn hơn kích thước của chuỗi đích chứa nó, thì hiện tượng tràn bộ nhớ sẽ bị xảy ra.

Và để khắc phục điểm yếu này của hàm strcpy() mà hàm strcpy\_c() đã được ra đời. Trong hàm strcpy\_c() có thêm chỉ định kích thước chuỗi đích, giúp chúng ta quản lý và không để xảy ra hiện tượng tràn bộ nhớ trong quá trình copy chuỗi trong C++.

Cú pháp

strcpy\_c(target, rsize\_t, source);

Trong đó:

* target là chuỗi đích
* rsize\_t là kích thước chuỗi đích
* source là chuỗi nguồn.

Lưu ý là chúng ta cũng cần phải viết thêm dòng #include <cstring> để load header file cstring để có thể sử dụng được hàm strcpy\_c() trong chuơng trình.

VD

#include <iostream>

#include <cstring>

using namespace std;

int main() {

    char str1[16] = "Hello World!";

    char str2[16];

    cout << "str2: " << str2 <<endl;

    //Copy toàn bộ chuỗi str2 vào chuỗi str1

    strcpy\_s(str2, 16, str1);

    cout << "str1: " << str1;

    return 0;

}

//Hello World!

### Strncpy(): copy n ký tự

Cú pháp

strncpy(target, source, n);

Trong đó:

* target là chuỗi đích
* source là chuỗi nguồn.
* n là số ký tự lớn nhất có thể copy

Hàm strncpy() sẽ copy từ chuỗi source một số lượng n ký tự, và dán nó vào chuỗi target. Hai chuỗi này cần phải tồn tại trước đó, thông qua việc khởi tạo hay khai báo chuỗi trong C++.

Cần chú ý là nếu chỉ định giá trị của n nhỏ hơn độ dài của chuỗi nguồn khi sử dụng hàm strncpy() thì ký tự kết thúc chuỗi \0 sẽ không được copy vào chuỗi đích, do đó chúng ta cần phải thêm xử lý nối ký tự này vào chuỗi đích target.

Và chúng ta cần phải viết thêm dòng #include <cstring> để load header file cstring để có thể sử dụng được hàm strlen() trong chuơng trình.

VD

#include <iostream>

#include <cstring>

using namespace std;

int main(){

    char str1[30];

    char str2[30] = "I love C++/C++";

    cout << "str2: " << str2 <<endl;

    //Copy nhiều nhất 6 ký tự từ chuỗi str2 vào chuỗi str1

    strncpy(str1, str2,6);

    str1[6] = '\0'; //Gán ký tự kết thúc chuỗi \0

    cout << "str1: " << str1;

}

## Nối chuỗi

## Strcat() nối 2 chuỗi

Cú pháp

strcat(target, source);

Trong đó:

* target là chuỗi đích
* source là chuỗi dùng để nối vào chuỗi target

Hàm strcat() sẽ copy toàn bộ nội dung của chuỗi source, sau đó gán vào vị trí cuối cùng của chuỗi target và trả về kết quả là con trỏ của chuỗi kết quả.

Và chúng ta cần phải viết thêm dòng #include <cstring> để load header file cstring để có thể sử dụng được hàm strcat() trong chuơng trình.

VD

#include <iostream>

#include <cstring>

using namespace std;

int main(){

    char str1[20] = "I Lov";

    char str2[10] = "e You";

    strcat(str1, str2);

    cout <<  str1;

}

//I Love You

## Strncat(): nỗi 2 chuỗi với n ký tự

Cú pháp:

strncat(target, source, n);

Trong đó:

* target là chuỗi đích
* source là chuỗi dùng để nối
* n là số ký tự tối đa được lấy ra từ đầu chuỗi source để nối vào chuỗi target

Hàm strncat() sẽ copy n ký tự tính từ đầu chuỗi source, sau đó gán chúng vào vị trí cuối cùng của chuỗi target và trả về kết quả là con trỏ của chuỗi kết quả.

Và chúng ta cần phải viết thêm dòng #include <cstring> để load header file cstring để có thể sử dụng được hàm strncat() trong chuơng trình.

VD:

#include <iostream>

#include <cstring>

using namespace std;

int main(){

    char str1[20] = "I Lov";

    char str2[10] = "e You";

    strncat(str1, str2,3);

    cout <<  str1;

}

// I Love Y

### Sprintf() nối nhiều chuỗi

Cú pháp:

sprintf (target, \*format, srt1 , srt2 , ...);

Trong đó

* target là chuỗi đích
* srt là các chuỗi dùng để nối vào chuỗi target
* \*format là chuỗi định dạng, là tập hợp các định dạng %s của các chuỗi str. Có bao nhiêu chuỗi str được chỉ định thì có bấy nhiêu định dạng tương ứng của chúng được viết bên trong chuỗi định dạng. Xem thêm: Định dạng nhập xuất trong C++.

VD:

#include <iostream>

#include <cstring>

using namespace std;

int main() {

    char str1[10] = "Hello";

    char str2[10] = "C++";

    char str3[10] = "World!";

    char target[100];

    sprintf(target, "%s %s %s\n", str1, str2,str3);

    cout << target;

    return 0;

}

//Hello C++ World!

## Tách chuỗi

### Strtok()

Hàm strtok() trong C++ là một hàm có sẵn trong header file cstring, giúp chúng ta tách chuỗi trong chuỗi C++ bằng ký tự chỉ định.

Cú pháp:

strtok(str, sep);

Trong đó:

* str là chuỗi cần tách
* sep là ký tự phân tách. (sep viết tắt separator)

Hàm strtok() sẽ trả về vị trí đầu chuỗi của chuỗi con đầu tiên được phân tách bằng dấu phân cách từ chuỗi ban đầu dưới dạng con trỏ chuỗi trong C++. Trong trường hợp không tìm thấy ký tự phân tách , giá trị NULL sẽ được trả về.

Cơ chế của hàm strtok() là điền ký tự kết thúc chuỗi \0 vào chuỗi ban đầu mỗi khi tìm thấy ký tự phân tách, do đó cần lưu ý là hàm strtok() sẽ làm biến đổi chuỗi ban đầu.

VD:

#include <iostream>

#include <cstring>

using namespace std;

int main(){

    char str[50] = "Ha Noi,Ninh Binh,Nam Dinh,Thanh Hoa";

    //Khai báo con trỏ chuỗi để chứa kết quả

    char \* p;

    p = strtok(str, ",");

    cout <<  p;

}

//Ha Noi

Ở ví dụ này, hàm strtok() sẽ hoạt động với các bước như sau:

* Hàm strtok() bắt đầu tìm vị trí ký tự phân tách là dấu , ở bên trong chuỗi str.
* Sau khi tìm thấy vị trí dấu phẩy đầu tiên, hàm strtok() sẽ điền ký tự kết thúc chuỗi \0 vào vị trí tìm thấy. Khi đó, chuỗi ban đầu sẽ chuyển thành dạng Ha Noi\0Ninh Binh,Nam Dinh,Thanh Hoa.
* Hàm strtok() quay trở về đầu chuỗi ban đầu và trả về vị trí đầu chuỗi này dưới dạng con trỏ chuỗi trong C++. Cuối cùng khi in con trỏ chuỗi này, chuỗi sẽ được in ra từ vị trí đầu chuỗi là ký tự H cho đến ký tự thúc chuỗi \0 (Ha Noi\0), do đó chuỗi kết quả Ha Noi sẽ được in ra màn hình.

### Tiếp tục sử dụng strtok() để tách toàn bộ chuỗi

Ở phần trên chúng ta đã biết hàm strtok() trong C++ chỉ có thể giúp chúng ta tách một chuỗi con bằng dấu phân tách từ chuỗi ban đầu trong mỗi lần sử dụng mà thôi.

Do đó, để có thể tách toàn bộ chuỗi con từ chuỗi ban đầu trong C++ thì sau lần tách đầu tiên, chúng ta cần phải tạo ra một vòng lặp để lần lượt tách các chuỗi con con lại từ chuỗi ban đầu bằng hàm strtok().

Tham số NULL được truyền vào để báo hàm tiếp tục tách từ vị trí hoàn thành trước.

VD

#include <iostream>

#include <cstring>

using namespace std;

int main(){

    char str[50] = "Ha Noi,Ninh Binh,Nam Dinh,Thanh Hoa";

    char \* p;

    //Tách chuỗi con lần đầu tiên

    p = strtok(str, ",");

    cout << p <<endl;

    //Tách chuỗi con từ lần thứ 2 trở đi

    //Bằng cách sử dụng hàm strok cho tới khi kết quả NULL được trả về.

    while(p != NULL) {

        //Chỉ dịnh đối số NULL trong hàm strtok để tiếp tục tách chuỗi ban đầu

        p = strtok(NULL, ",");

        if(p != NULL) {

            cout<< p<<endl;

        }

    }

    return 0;

}

Ha Noi

Ninh Binh

Nam Dinh

Thanh Hoa

## Tách số trong chuỗi

Sử dụng isdigit()

VD

#include <iostream>

#include <cctype>

#include <cstring>

using namespace std;

int main(){

    char str[] = "A100B20C3000D40";

    char result[100];

    int i=0;

    int n=0;

    for(i=0; i<=strlen(str); ++i){

        /\* Kiểm tra ký tự thứ i trong chuỗi đã cho có phải là số hay không\*/

        /\* Nếu là số thì lưu vào chuỗi result\*/

        if(isdigit(str[i])){

            result[n] = str[i];

            n++;

            result[n] = '\0';

        }else if(n){ /\* Nếu chuỗi result chứa phần tử thì in chuỗi result\*/

            cout << result<<endl;

            n=0; /\* Reset và làm rỗng chuỗi result\*/

        }

    }

    return 0;

}

100

20

3000

40

## Cắt chuỗi

Phương pháp sử dụng strncpy()

strncpy(target, source + begin, n);

Trong đó:

* target là chuỗi đích để chứa kết quả cắt
* source là chuỗi nguồn.
* begin là vị trí bắt đầu cắt trong chuỗi nguồn
* n là số ký tự sẽ cắt từ vị trí begin.

Hiểu đơn giản thì cú pháp trên sẽ cắt n ký tự từ vị trí begin trong chuỗi source và trả về kết quả là chuỗi target.

Cần chú ý là nếu chỉ định giá trị của n nhỏ hơn độ dài của chuỗi nguồn khi sử dụng hàm strncpy() thì ký tự kết thúc chuỗi \0 sẽ không được copy vào chuỗi đích, do đó chúng ta cần phải thêm xử lý nối ký tự này vào chuỗi đích target.

VD

#include <iostream>

#include <cstring>

using namespace std;

int main() {

    char str[] = "01234567890";

    char t[64];

    strncpy( t, str+3, 5 ); /\*Cắt 5 ký tự từ vị trí thứ 3 \*/

    t[5] = '\0';            /\*Thêm ký tự kết thúc chuỗi vào kết quả\*/

    cout << t ;

}

//34567

## Xoá ký tự trong chuỗi

Phương pháp: lấy từng kí tự trong chuỗi ban đầu ra. Nếu không phải kí tự cần xoá thì lưu và chuỗi đích.

VD

#include<iostream>

using namespace std;

char\* pop\_str(char\* a, char chr, char\* b){

  int a\_cnt;

  int b\_cnt = 0;

  /\* So sánh từng ký tự trong chuỗi a với ký tự C++\*/

  for ( a\_cnt = 0; a[a\_cnt] != '\0'; ++a\_cnt ) {

    /\* Nếu ký tự khác C++ thì lấy ra và thêm vào kết quả \*/

    if ( a[a\_cnt] != chr ) {

      b[b\_cnt] = a[a\_cnt];

      ++b\_cnt;

    }

  }

  b[b\_cnt] = '\0'; /\* Gán ký tự kết thúc chuỗi '\0' \*/

  return b;

}

int main(){

  char a[50] = "abcdcde"; /\* Chuỗi ban đầu\*/

  cout<< "before: "<<a<<endl;

  char b[50]; /\* Chuỗi kết quả\*/

  char chr = 'a'; /\* Ký tự cần xóa\*/

  cout<< "after : "<<pop\_str(a,chr,b)<<endl;

  return 0;

}

## Xoá chuỗi con trong chuỗi

Phương pháp: Tìm kiếm vị trí của chuỗi con trong chuỗi mẹ bằng hàm strstr(), sau đó tách chuỗi mẹ ra từ vị trí tìm thấy, và nối các phần này lại với nhau bằng hàm strcat().

VD

#include <iostream>

#include <cstring>

using namespace std;

void pop\_child\_str(char\* str1,char\* str2){

  const int len = strlen(str2);

  char str3[128] = "";

  char \*p1 = str1, \*p2;

    /\* Tạo vòng lặp để xóa hết chuỗi con \*/

  while((p2 = strstr(p1,str2)) != NULL) { /\*Tìm vị trí chuỗi con bằng hàm strstr\*/

    strncat(str3,p1,p2 - p1);   /\* Nối các phần không chứa chuỗi con \*/

    p1 = p2 + len;      /\* Dịch chuyển con trỏ sang vị trí tìm kiếm tiếp theo \*/

  }

  strcat(str3,p1);

  cout << str3;

}

int main()

{

  char str1[128], str2[128];

  cout << "Nhap chuoi me: ";

  cin >>  str1;

  cout << "Nhap chuoi con can xoa: ";

  cin >>  str2;

  pop\_child\_str(str1,str2);

  return(0);

}

Nhap chuoi me: abcde

Nhap chuoi con can xoa: bc

ade

## Xắp xếp chuỗi

### Xắp xếp các ký tự của chuỗi theo bảng chữ cái

VD

#include <iostream>

#include <cstring>

using namespace std;

int main (void) {

  char str[256] ;

  /\* Nhập các chuỗi vào chương trình \*/

  cout << "Nhap chuoi: ";

  cin >>  str;

  /\* Tìm độ dài chuỗi\*/

  int length = strlen(str);

  /\* Tạo vòng lặp để tiến hành sắp xếp ký tự trong chuỗi\*/

  int i, j;

  char temp;

  for (i = 0; i < length-1; i++) {

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

         /\* So sánh lần lượt các ký tự và hoán đổi giá trị nhờ biến temp\*/

         if (str[i] > str[j]) {

            temp = str[i];

            str[i] = str[j];

            str[j] = temp;

         }

      }

  }

  cout << "Ket qua: "<< str;

  return 0;

}

Nhap chuoi: akjhu

Ket qua: ahjku

### Sắp xếp các chuỗi theo thứ tự chữ cái

#include<iostream>

#include<cstring>

using namespace std;

int main()

{

  int i, j, num;

  /\* Lệnh nhập số chuỗi cần nhập \*/

  cout << "So chuoi can nhap: ";

  cin >> num;

  /\* Khai báo chuỗi sử dụng để sắp xếp \*/

  char str[num][20], tmp[20];

  /\* Nhập các chuỗi vào chương trình \*/

  cout << "\nHay nhap "<<num<<" chuoi vao chuong trinh\n";

  for (i=0;i<num;i++) {

    cout << "Chuoi thu "<<i+1<<"= ";

    cin >>  str[i];

  }

  /\* Sắp xếp \*/

  for(i=1;i<num;i++){

    for(j=1;j<num;j++){

      if(strcmp(str[j-1], str[j])>0){

        strcpy(tmp, str[j-1]);

        strcpy(str[j-1], str[j]);

        strcpy(str[j], tmp);

      }

    }

  }

  /\* In kết quả sắp xếp\*/

  cout << endl;

  for (i=0;i<num;i++)

    cout <<str[i]<<endl;

  return 0;

}

So chuoi can nhap: 3

Hay nhap 3 chuoi vao chuong trinh

Chuoi thu 1= xyz

Chuoi thu 2= abcd

Chuoi thu 3= mnk

abcd

mnk

xyz

# Tham khảo

Lập trình căn bản:

<https://laptrinhcanban.com/cpp/lap-trinh-cpp-co-ban/>