

NGÔN NGỮ LẬP TRÌNH C++

Hàm, con trỏ, mảng, chuỗi ký tự



Hàm

- Một hàm là một nhóm các câu lệnh cùng nhau thực hiện một nhiệm vụ. Mỗi chương trình C++ có ít nhất một hàm, là hàm `main()`
- Lợi thế của các hàm trong C++
 - Có khả năng sử dụng lại
 - Tối ưu hóa chương trình



Hàm

- **Khai báo và sử dụng hàm**

```
<Kiểu trả về> <Tên hàm>(Danh sách tham số) {  
    <Câu lệnh 1>;  
    <Câu lệnh 2>;  
}
```

- **Ví dụ**

```
int kiem_tra_nt(long long n) {  
    if (n < 2) return 0;  
    for (int i = 2; i <= sqrt(n); i++) {  
        if (n % i == 0) return 0;  
    }  
    return 1;  
}
```



Hàm

- Chồng hàm

- Chồng hàm là kỹ thuật sử dụng 2 hàm có cùng tên nhưng khác tham số hoặc khác kiểu trả về

- Ví dụ:

```
int tong(int a, int b) {  
    return a + b;  
}  
float tong(float a, float b) {  
    return a + b;  
}
```



Tham trị, tham chiếu

- Tham trị
 - Giá trị được truyền vào tham số của hàm, sau khi kết thúc hàm, giá trị của biến được truyền vào không thay đổi.
- Tham chiếu
 - Nội dung được truyền vào tham số của hàm, sau khi kết thúc hàm, giá trị của biến được truyền vào bị thay đổi.
 - Để truyền tham chiếu vào hàm, sử dụng dấu & trước tên biến.



Nguyên tắc xây dựng hàm

- Chức năng
 - Cố gắng mỗi hàm thực hiện giải quyết một vấn đề duy nhất
- Xác định kiểu trả về
 - Nếu hàm trả giá trị thì xác định kiểu dữ liệu tương ứng, return phù hợp
 - Nếu hàm chỉ thực hiện thủ tục, không mang giá trị thì kiểu trả về là void
 - Mỗi hàm chỉ trả về được 1 giá trị thông qua kiểu trả về, muốn trả về nhiều giá trị thì có thể sử dụng tham chiếu
- Đặt tên
 - Đặt tên hàm có ý nghĩa



Bài tập

1. Viết hàm đổi chỗ 2 số tự nhiên a và b
2. Viết hàm tìm số lớn nhất trong 3 số a, b, c
3. Viết hàm tính tổng, hiệu, tích, thương của 2 số a, b
4. Viết hàm tính diện tích, chu vi của hình chữ nhật
5. Viết chương trình đếm tất cả các số tự nhiên n không quá 10 chữ số thỏa mãn tất cả điều kiện dưới đây
 - n là số nguyên tố
 - Số viết ngược lại của n là 1 số nguyên tố
 - Tổng các chữ số của n cũng là số nguyên tố
 - Mỗi chữ số trong n cũng là số nguyên tố



Con trỏ

- Con trỏ là một biến đặc biệt, nó chứa địa chỉ của một biến khác.
Con trỏ có kiểu là kiểu của biến mà nó trỏ tới
 $\text{<Kiểu dữ liệu> *<tên con trỏ>;}$
- Lấy địa chỉ con trỏ
 $\text{<Tên con trỏ> = \&<tên biến>}$
- Lấy giá trị con trỏ
 *<Tên con trỏ>
- Phép gán con trỏ
 $\text{<Tên con trỏ1> = <Tên con trỏ2>}$



Con trỏ

- Đặc trưng

- Hiệu quả nhưng khó điều khiển
- Quan hệ chặt chẽ với mảng và chuỗi
- Có thể khai báo kiểu con trỏ cho bất kỳ kiểu dữ liệu nào
- Con trỏ khởi tạo bằng 0, NULL (Không trỏ vào gì cả), hoặc một địa chỉ.

Non-zero value



null



0



undefined



Con trỏ

- Lỗi thường gặp

```
int value, *p;
```

```
p = value; // sai, p cần địa chỉ, không phải giá trị
```

```
*p = &value; // sai, *p là giá trị nên phải được gán giá trị, không phải địa chỉ
```

```
p = &value; // đúng
```

```
*p = value; // đúng
```



Mảng

- Mảng là một loại cấu trúc dữ liệu trong ngôn ngữ lập trình C/C++, nó lưu trữ một tập hợp tuần tự các phần tử cùng kiểu với độ dài cố định.
- Mảng thường được sử dụng để lưu trữ tập hợp dữ liệu, nhưng nó cũng hữu dụng khi dùng để lưu trữ một tập hợp biến có cùng kiểu.

- Sử dụng mảng

<Kiểu dữ liệu> <tên biến mảng>[<Số lượng phần tử tối đa>];

```
int array[4] = { 5, 8, 2, 7 };
```

- Sử dụng mảng 2 chiều

```
int arr[3][5];
```

```
int arr[3][5] = {{5, 12, 17, 9, 3}, {13, 4, 8, 14, 1}, {9, 6, 3, 7, 21}};
```



Con trỏ và mảng

- Mảng và con trỏ có quan hệ mật thiết
 - Tên mảng coi như con trỏ hằng
 - Con trỏ có thể thao tác trên chỉ số của mảng
- Truy cập các phần tử mảng bằng con trỏ
 - Phần tử $b[n]$ có thể được truy cập bằng $*(bPtr+n)$
 - Địa chỉ, ví dụ: $\&b[3]$ tương đương $bPtr + 3$
 - Tên mảng có thể coi như con trỏ, ví dụ:
 $b[3]$ tương đương $*(b+3)$
 - Có thể đánh chỉ số cho con trỏ, ví dụ:
 $bPtr[3]$ tương đương $b[3]$



Con trỏ và mảng

- Khai báo mảng bằng con trỏ

```
int *a;
```

```
a = new int[SIZE];
```

```
int **a;
```

```
a = new int*[SIZE];
```

- Thu hồi bộ nhớ cấp phát
`delete[] a;`



Xâu kí tự

- Một xâu có thể được coi là một mảng một chiều, trong đó mỗi phần tử là một ký tự. Ký tự kết thúc là '\0'

`char s1[200];`

`fgets(s1,200,stdin):` nhập xâu

`strcpy(s1, s2):` copy xâu s2 vào s1

`strcat(s1, s2):` nối xâu s2 vào sau xâu s1

`strlen(s) :` tính độ dài xâu s

`strcmp(s1, s2) :` so sánh hai xâu s1 và s2 theo thứ tự từ điển.



Xâu kí tự

- Trong C++ có kiểu dữ liệu dành riêng cho xử lý chuỗi là string
- `string str;`
- `s1 = s2` sao chép xâu s2 vào s1
- `s1 = s1 + s2` Nối s2 vào sau s1
- `s1 == s2` So sánh xâu s1 có trùng xâu s2 hay không
- `s1 > s2` So sánh xâu s1 có đứng trước s2 hay không theo thứ tự từ điển
- `s1 < s2` So sánh xâu s1 có đứng sau s2 hay không theo thứ tự từ điển
- `s1 <= s2` So sánh xâu s1 có đứng trước hoặc trùng s2 hay không theo thứ tự từ điển
- `s1 >= s2` So sánh xâu s1 có đứng sau hoặc trùng s2 hay không theo thứ tự từ điển
- `s1 != s2` So sánh xâu khác nhau hay không



Xâu kí tự

- Một số hàm xử lý xâu

STT	Tên hàm (Phương thức)	Ý nghĩa
1	s.size()	Trả lại độ dài string s
2	s.length()	Trả lại độ dài string s
3	getline(cin, s)	Nhập một dòng từ bàn phím cho string s
4	s.erase(n, k)	Xóa k ký tự trong s kể từ vị trí thứ n
5	s.insert(n, s1)	Chèn s1 vào s kể từ vị trí thứ n.
6	s.insert(n, s1, k, m)	Chèn m ký tự kể từ ký tự thứ k trong s1 vào s kể từ vị trí thứ n.
7	s.replace(n, k, s1)	Thay thế k ký tự trong s kể từ vị trí thứ n bằng xâu s1.
8	s.find(s1)	Trả lại vị trí xuất hiện đầu tiên của s1 trong s.
9	s.rfind(s1)	Trả lại vị trí xuất hiện cuối cùng của s1 trong s.
10	s.at(int i)	Truy nhập đến phần tử thứ i trong string

Bài tập

Viết chương trình nhập vào một xâu ký tự S, thực hiện phân tích xâu đã nhập chứa bao nhiêu chữ cái, chữ số và các ký tự khác.

Input:

Chỉ có một dòng ghi xâu S.

Output:

Ghi ra kết quả theo mẫu trong ví dụ: số chữ cái, số chữ số, số các ký tự khác.

Ví dụ:

Input	Output
mon thcs mon THCS 2 ...	14 1 8

```
#include <iostream>
#include <string.h>
using namespace std;
int main() {
    string s;
    getline(cin, s);
    int k = s.length();
    int so=0, chu=0, khac = 0;
    for (int i = 0; i < k; i++) {
        if ((s[i]>=65 && s[i]<=90) || (s[i]>=97 && s[i]<=122)) chu++;
        else if ((s[i]>=48 && s[i]<=57)) so++;
        else khac++;
    }
    cout<< chu<<" "<< so<<" "<< khac;
    return 0;
}
```



Bài tập

ĐỊA CHỈ EMAIL - 1

Bài làm tốt nhất



Địa chỉ email của cán bộ PTIT được cấp theo nguyên tắc ghép tên với chữ cái đầu tiên của họ và tên đệm. Viết chương trình cho phép tạo các địa chỉ email theo tên cán bộ

Input

Chỉ có một xâu ký tự họ tên độ dài không quá 100.

Output

Ghi ra địa chỉ email theo mẫu trong ví dụ.

Ví dụ

Input	Output
Nguyen vAn nAM	namnv@ptit.edu.vn

```
string toEmail() {  
    string c="";  
    for (int i=0; i<a.length(); i++) {  
        if (a[i] >= 'A' && a[i] <= 'Z') {  
            a[i] = a[i] + 32;  
        }  
    }  
    a = ' ' + a;  
    while (a[a.length()-1] != ' ') {  
        c = a[a.length()-1] + c;  
        a.erase(a.length()-1,1);  
    }  
}
```

```
for (int i=1; i<a.length()-1; i++) {  
    if (a[i-1] == ' ') {  
        c = c + a[i];  
    }  
}  
c += "@ptit.edu.vn";  
return c;  
}
```

