



LẬP TRÌNH NÂNG CAO

Giảng viên: TS.GVC Bùi Thị Thanh Xuân
(ĐT: 0902001581/ Email: bttxuan@tlu.edu.vn)

Bộ môn: Tin học và KTTT

Năm học: 2020-2021



Giới thiệu môn học

- Số tín chỉ: 3
- Số buổi học: 15 buổi, 3 tiết/buổi
- Ngôn ngữ lập trình sử dụng: C++
- Điều kiện dự thi kết thúc HP: Nghỉ học không quá 20% số tiết (Nghỉ không quá 3 buổi)
- Trọng số điểm môn học: Chi tiết trong bảng sau



Giới thiệu môn học

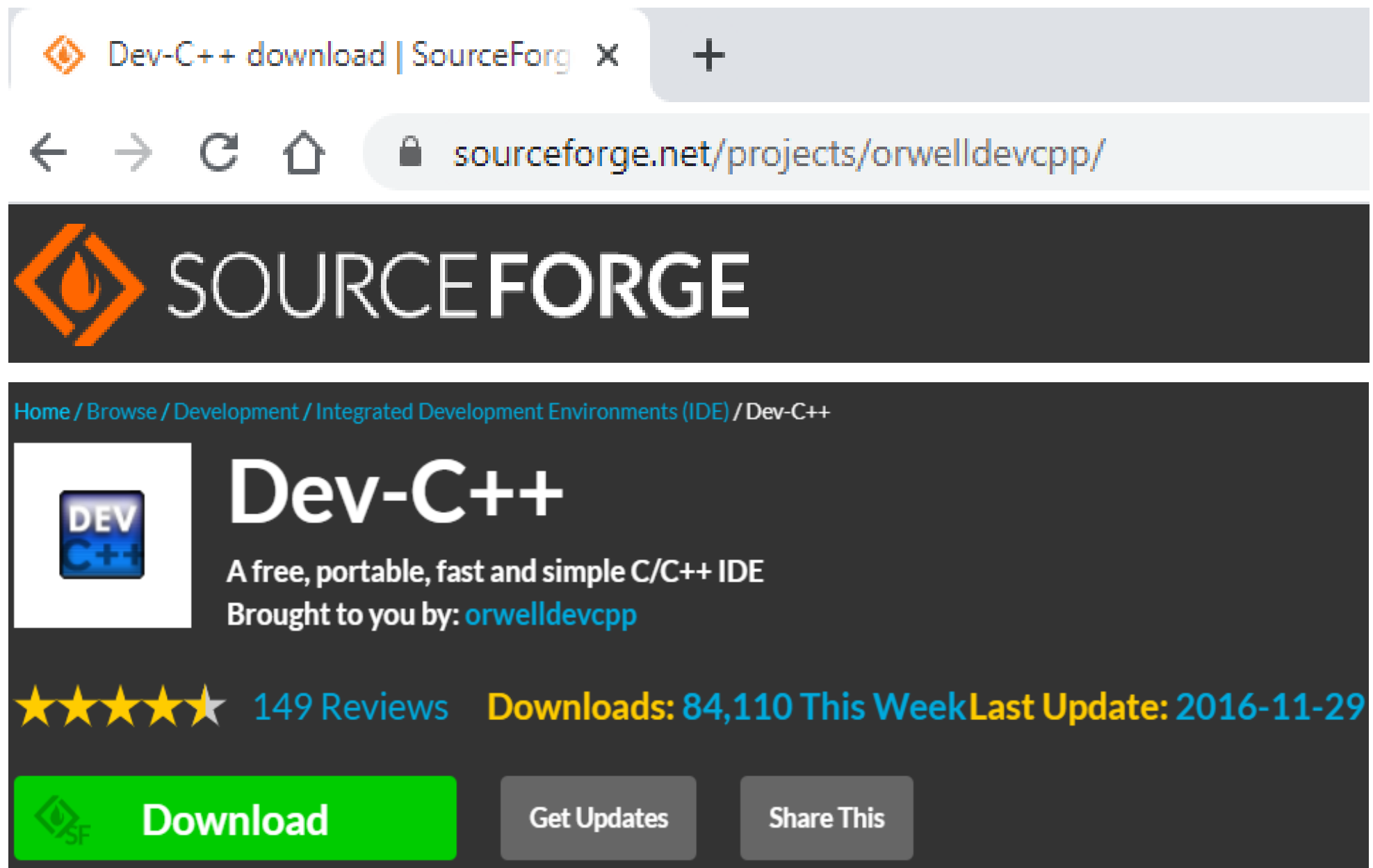
Hình thức	Số lần	Mô tả	Thời gian	Trọng số
Chăm điểm chuyên cần	1 lần lấy điểm	- Dựa vào số buổi dự học của sv/tổng 15 buổi và ý thức của SV trong giờ học	Tất cả các buổi học	10%
Bài kiểm tra	2 lần lấy điểm	- 60 phút	- Kết thúc Chương 2	20%
		- 1 bài kiểm tra - 1 bài kiểm tra	- Kết thúc Chương 4	20%
Tổng điểm quá trình				50%
Thi cuối kỳ	1 lần lấy điểm	- 60-90 phút - Thi trên máy	1-2 tuần sau khi kết thúc môn học	50%



Nội dung chính

- Chương 1: Hàm
- Chương 2: Mảng và chuỗi ký tự
- Chương 3: Con trỏ và bộ nhớ
- Chương 4: Kiểu cấu trúc và hợp
- Chương 5: Vào ra dữ liệu

Ngôn ngữ C++



The screenshot shows a web browser window with a single tab titled "Dev-C++ download | SourceForge". The address bar displays the URL "sourceforge.net/projects/orwelldvcpp/". The page header features the SourceForge logo and name. The main content area includes a breadcrumb trail: "Home / Browse / Development / Integrated Development Environments (IDE) / Dev-C++". Below this is the Dev-C++ logo, the title "Dev-C++", and the description "A free, portable, fast and simple C/C++ IDE Brought to you by: [orwelldvcpp](#)". The page also shows a rating of four stars, "149 Reviews", and download statistics: "Downloads: 84,110 This Week" and "Last Update: 2016-11-29". At the bottom, there are three buttons: a green "Download" button with the SourceForge logo, a grey "Get Updates" button, and a grey "Share This" button.

Dev-C++ download | SourceForge

sourceforge.net/projects/orwelldvcpp/


SOURCEFORGE

Home / Browse / Development / Integrated Development Environments (IDE) / Dev-C++

Dev-C++

A free, portable, fast and simple C/C++ IDE
Brought to you by: [orwelldvcpp](#)

★★★★☆ 149 Reviews Downloads: 84,110 This Week Last Update: 2016-11-29

 **Download** Get Updates Share This



Chương 1: Hàm

- 1.1. Cấu trúc chung của hàm
- 1.2. Cách sử dụng hàm
- 1.3. Các hàm có sẵn
- 1.4. Biến toàn cục, biến cục bộ
- 1.5. Truyền tham số
- 1.6. Nạp chồng hàm
- 1.7. Hàm đệ quy
 - 1.7.1. Khái niệm đệ quy
 - 1.7.2. Cấu trúc hàm đệ quy
 - 1.7.3. Một số bài toán giải được bằng đệ quy



Khái niệm chương trình con

- Chương trình con
 - Là đoạn chương trình thực hiện một nhiệm vụ cụ thể và được đóng gói theo cấu trúc xác định
 - Được định nghĩa một lần và sử dụng lại nhiều lần.
- Vai trò
 - Tổ chức một chương trình thành nhiều phần nhỏ để tăng tính cấu trúc.
 - Tiết kiệm thời gian bằng cách sử dụng chương trình con đã có thay vì viết lại.
 - Mở rộng tính năng cho chương trình bằng cách sử dụng các thư viện, ...
 - Gỡ lỗi, bảo trì chương trình dễ dàng hơn



Hàm trong ngôn ngữ C++

- ❑ Các ngôn ngữ lập trình nói chung: 2 loại chương trình con
 - Hàm: thực hiện công việc và trả lại kết quả.
 - Thủ tục: thực hiện công việc, không trả lại kết quả.
- ❑ Ngôn ngữ C++
 - Chỉ có 1 loại chương trình con là hàm (function).
 - Hàm trong C++ tương đương cả hàm và thủ tục trong các ngôn ngữ khác.
 - Sử dụng kiểu void (kiểu dữ liệu không định kiểu) khi hàm không trả về dữ liệu.

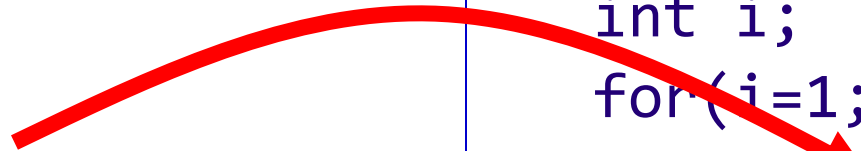
Ví dụ về hàm

Khai báo
chương trình
con



```
int bp(int x){  
    int y;  
    y = x*x;  
    return y;  
}
```

Gọi chương trình
con ra thực hiện



```
int main(){  
    int i;  
    for(i=1; i< 20; i+=2)  
        cout<<bp(i)<<endl;  
    return 0;  
}
```

Chương 1: Hàm

■ Khai báo và sử dụng

```
File Edit Search View Project Execute Tools AStyle
(globals)
ham_vd1.cpp
1  #include <iostream>
2  using namespace std;
3  int bp(int x){
4      int y;
5      y = x * x;
6      return y;
7  }
8  int main(){
9      for(int i=1; i< 20; i+=2)
10         cout<<bp(i)<<endl;
11         return 0;
12 }
```

```
1
9
25
49
81
121
169
225
289
361
-----
Process exited after 0.4231 seconds with return value 0
Press any key to continue . . .
```

Cấu trúc chương trình khi viết hàm

```
//Phần khai báo thư viện
```

```
#include <iostream>
```

```
using namespace std;
```

```
//Phần định nghĩa hàm
```

Viết định nghĩa
hàm tại đây

```
//Hàm chính
```

```
int main () {
```

```
//Lời gọi hàm
```

Gọi hàm

```
    return 0;
```

```
}
```

Cấu trúc chương trình khi viết hàm

//Phần khai báo thư viện

#include <iostream>

using namespace std;

//Phần khai báo nguyên mẫu hàm

//Hàm chính

int main () {

//Lời gọi hàm

return 0;

}

//Phần định nghĩa hàm

Viết nguyên
mẫu hàm tại
đây

Gọi hàm

Viết định
nghĩa hàm
tại đây

Định nghĩa hàm

Cú pháp

Dòng đầu hàm

Kiểu_hàm Tên_hàm (DS khai báo tham số)

{

[<Các khai báo cục bộ>]

[<Các câu lệnh>]

}

Thân hàm

Dòng đầu hàm

Kiểu_hàm Tên_hàm (DS khai báo tham số)

- Mô tả các thông tin được trao đổi giữa bên trong và bên ngoài hàm.
 - Tên của hàm,
 - Các tham số đầu vào
 - Hàm cần những thông tin gì để hoạt động
 - Tham số đầu ra và giá trị trả về
 - Hàm cung cấp những thông tin gì cho môi trường
- Dùng phân biệt các hàm với nhau,
 - Không tồn tại 2 hàm có dòng đầu hàm giống nhau.

Dòng đầu hàm → Tên hàm

Là tên do người sử dụng tự định nghĩa

- Tuân theo quy tắc đặt tên đối tượng
- Nên mang ý nghĩa gợi ý về chức năng của hàm

- Không trùng từ khóa,
- Không chứa các ký tự đặc biệt,
- Không chứa dấu cách (khoảng trắng)
- Không bắt đầu bằng chữ số
- Phân biệt chữ hoa/ thường

Dòng đầu hàm → Khai báo các tham số hình thức

- Khai báo các thông tin cần cho hoạt động của hàm và các thông tin, kết quả tính toán được hàm trả lại.
 - Tham số chứa dữ liệu vào cung cấp cho hàm
 - Tham số chứa dữ liệu ra mà hàm tính toán được.
- Các tham số sử dụng trong khai báo hàm là tham số hình thức.
 - Nguyên tắc khai báo tham số hình thức như giống như khai báo một biến

kiểu_dữ_liệu_của_tham_số tên_của_tham_số



Dòng đầu hàm→Khai báo các tham số hình thức

- Các tham số cung cấp cho hàm trong quá trình thực hiện hàm là tham số thực sự
 - Kiểu dữ liệu của tham số thực phải giống kiểu dữ liệu của tham số hình thức tương ứng với tham số thực sự đó.
- Một hàm có thể có một, nhiều hoặc không có tham số nào cả
 - Nếu có nhiều tham số, phải được phân cách với nhau bằng dấu phẩy.
 - không có tham số vẫn phải có cặp dấu ngoặc đơn sau tên hàm



Dòng đầu hàm → Kiểu dữ liệu trả về

- Thông thường hàm sau khi được thực hiện sẽ trả về một giá trị kết quả tính toán nào đó.
- Để sử dụng được giá trị đó cần phải biết nó thuộc kiểu dữ liệu gì.
 - Kiểu dữ liệu của đối tượng tính toán được hàm trả về được gọi là kiểu dữ liệu trả về của hàm.



Dòng đầu hàm→Kiểu dữ liệu trả về

- Trong C++, kiểu dữ liệu trả về của hàm có thể là kiểu dữ liệu bất kì (kiểu dữ liệu có sẵn hoặc kiểu dữ liệu do người dùng tự định nghĩa) nhưng không được là kiểu dữ liệu mảng.
- Nếu kiểu dữ liệu trả về là kiểu void thì hàm không trả về giá trị nào cả.
- Nếu không khai báo kiểu dữ liệu trả về thì chương trình dịch của C++ sẽ ngầm hiểu rằng kiểu dữ liệu trả về của hàm là kiểu int.



Thân hàm

- Danh sách các câu lệnh
- Thường có ít nhất một lệnh return

Hoạt động của hàm

- Thực hiện lần lượt các lệnh cho đến khi
 - Thực hiện xong tất cả các câu lệnh có trong thân hàm
 - Gặp lệnh return
 - Cú pháp chung

return [biểu_thức];



Thân hàm (tiếp)

Khi gặp lệnh *return biểu_thức*;

- Tính toán giá trị của *biểu_thức*,
- Lấy kết quả tính toán được làm giá trị trả về cho lời gọi hàm
- Kết thúc việc thực hiện hàm, trở về chương trình đã gọi nó.

Nếu *return không có phần biểu_thức*,

– Kết thúc thực hiện hàm mà không trả về giá trị nào cả.

- Dùng khi hàm được khai báo có kiểu trả về là void



Sử dụng hàm

Tên_hàm (DS_tham_số_thực_sự);

Ví dụ:

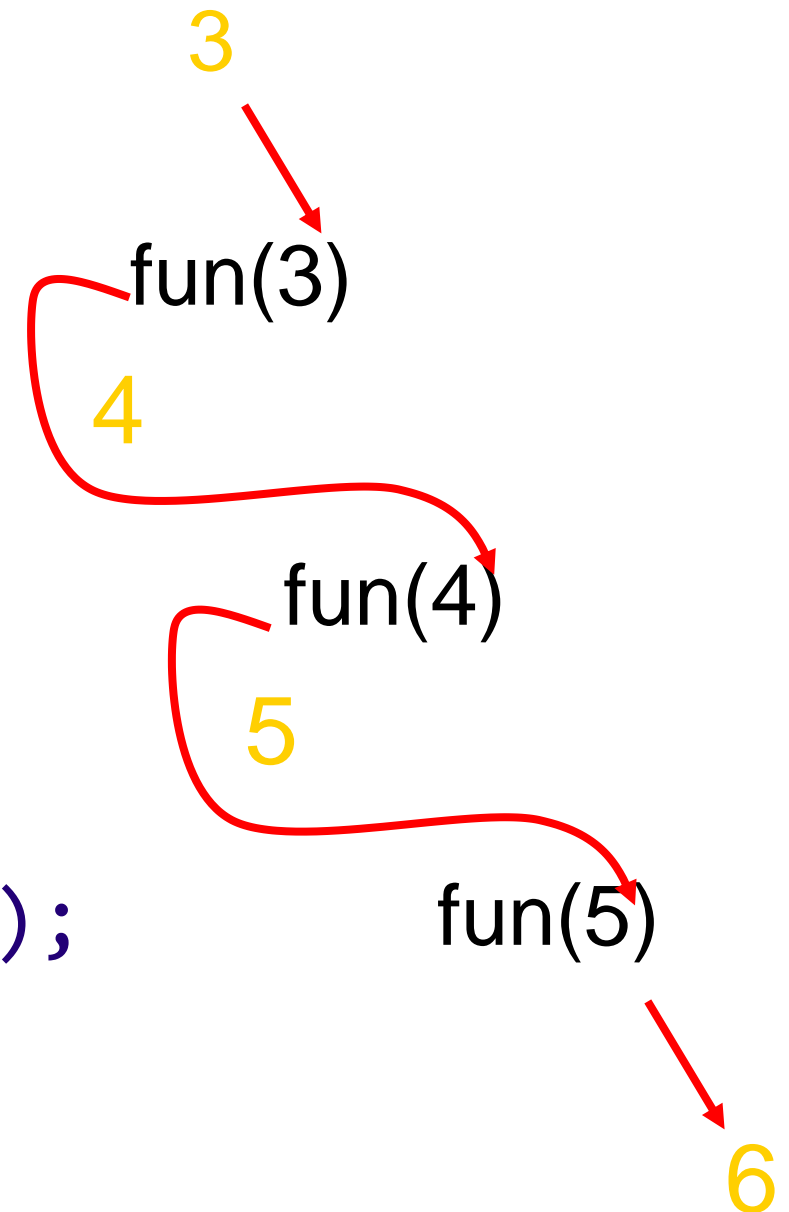
`N = bp(1); N = bp(3);...`

Lưu ý:

- Gọi hàm thông qua tên hàm và các tham số được cung cấp thực sự cho hàm (*tham số thực sự*).
- Nếu hàm nhận nhiều tham số thì các tham số ngăn cách nhau bởi dấu phẩy
- Các tham số hình thức của hàm sẽ nhận các giá trị từ tham số truyền vào
- Sau khi thực hiện xong, trở về điểm mà hàm được gọi

Ví dụ 1: Cho biết kết quả thực hiện chương trình

```
#include <iostream>
using namespace std;
int fun(int a){
    a++;
    return a;
}
int main(){
    cout<<fun(fun(fun(3)));
    return 0;
}
```



Ví dụ 2: Tính TBC $f(a)$, $f(b)$, $f(c)$ nếu

$$f(x) = x^5 + x^2$$

```
#include <iostream>
#include <cmath>
using namespace std;

float f(float x) {
    return pow(x,5)+pow(x,2);
}

int main() {
    float a, b, c;
    cout<<"So 3 so thuc: "; cin >>a>>b>>c;
    cout<<"Ket qua "<<(f(a) + f(b) +
        f(c))/3<<endl;
    return 0;
}
```


Ví dụ 2: Tính TBC $f(a)$, $f(b)$, $f(c)$ nếu

$$f(x) = x^5 + x^2$$

```
#include <iostream>
#include <cmath>
using namespace std;

float f(float x) {
    return pow(x,5)+pow(x,2);
}

int main() {
    float a,b,c;
    cout<<"So 3 so thuc: "; cin >>a>>b>>c;
    cout<<"Ket qua "<<(f(a) + f(b) + f(c))/3<<endl;
    return 0;
}
```

Ví dụ 2: Tính TBC $f(a)$, $f(b)$, $f(c)$ nếu $$f(x) = x^5 + x^2$$

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4  float f(float x) {
5      return pow(x,5) + pow(x,2);
6  }
7  int main() {
8      float a, b, c;
9      cout<<"So 3 so thuc: "; cin>>a>>b>>c;
10     cout<<"Ket qua "<< (f(a) + f(b) + f(c))/3<<endl;
11     return 0;
12 }
```

```
So 3 so thuc: 1 2 3
Ket qua 96.6667
```

```
-----
Process exited after 2.907 seconds with return value 0
Press any key to continue . . .
```

Ví dụ 3:

Tìm USCLN của hai số nguyên a và b

- Trong toán học, **ước số chung lớn nhất (USCLN)** hay **ước chung lớn nhất (UCLN)** của hai hay nhiều số nguyên là số nguyên lớn nhất trong tập hợp ước chung của các của các số đó.
- Trong trường hợp tất cả số nguyên đều bằng 0 thì chúng không có UCLN vì khi đó mọi số tự nhiên khác không đều là ước chung của các số đó. Nếu trong các số đó có ít nhất một số bằng 0 và ít nhất một số khác 0 thì UCLN của chúng bằng UCLN của các số khác 0.
- $\text{UCLN}(a, 0) = |a|$, với mọi $a \neq 0$, vì mọi số khác 0 bất kỳ là ước của 0, và ước lớn nhất của a là $|a|$. Đây là trường hợp cơ sở trong thuật toán Euclid.

Ví dụ 3: Tìm USCLN của hai số nguyên a và b

```
1.  #include <iostream>
2.  #include <cmath>
3.  using namespace std;
4.  int uscln(int a, int b) {
5.      while (a != b){
6.          if(a > b)
7.              a = a - b;
8.          else
9.              b = b - a;
10.     }
11.     return a;
12. }
13. int main() {
14.     int a, b;
15.     char c;
16.     do{
17.         cout<<"Cho so nguyen a = "; cin>>a;
18.         cout<<"Cho so nguyen b = "; cin>>b;
19.         cout<<"Ket qua UCLN (a,b) = "<<uscln(a,b);
20.         cout<<"\nBan co muon tim tiep khong(C/K)? "; cin>>c;
21.     }while (c == 'C' || c == 'c');
22.     return 0;
23. }
```

```
Cho so nguyen a = 15
Cho so nguyen b = 13
Ket qua UCLN (a,b) = 1
Ban co muon tim tiep khong(C/K)? c
Cho so nguyen a = 24
Cho so nguyen b = 35
Ket qua UCLN (a,b) = 1
Ban co muon tim tiep khong(C/K)? c
Cho so nguyen a = 72
Cho so nguyen b = 81
Ket qua UCLN (a,b) = 9
Ban co muon tim tiep khong(C/K)? k

-----
Process exited after 26.3 seconds with return value 0
Press any key to continue . . .
```

Ví dụ 3: Tìm USCLN của hai số nguyên a và b

```
1.  #include <iostream>
2.  #include <cmath>
3.  using namespace std;
4.  int uscln(int a, int b) {
5.      while (a != b){
6.          if(a > b)
7.              a = a - b;
8.          else
9.              b = b - a;
10.     }
11.     return a;
12. }
13. int main() {
14.     int a, b;
15.     char c;
16.     do{
17.         cout<<"Cho so nguyen a = "; cin>>a;
18.         cout<<"Cho so nguyen b = "; cin>>b;
19.         cout<<"Ket qua UCLN (a,b) = "<<uscln(a,b);
20.         cout<<"\nBan co muon tim tiep khong(C/K)? "; cin>>c;
21.     }while (c == 'C' || c == 'c');
22.     return 0;
23. }
```

Ví dụ 3: Tìm USCLN của hai số nguyên a và b

```
#include <iostream>
#include <cmath>
using namespace std;
int uscln(int a, int b) {
    while (a != b){
        if(a > b)
            a = a - b;
        else
            b = b - a;
    }
    return a;
}
```

```
int main() {
    int a,b;
    char c;
    do{
        cout<<"Cho so nguyen a | = "; cin>>a;
        cout<<"Cho so nguyen b = "; cin>>b;
        cout<<"Ket qua UCLN (a,b) = "<<uscln(a,b);
        cout<<"\nBan co muon tim tiep khong(C/K)? "; cin>>c;
    }while (c == 'C' || c == 'c');
    return 0;
}
```

```
Cho so nguyen a = 12
Cho so nguyen b = 24
Ket qua UCLN (a,b) = 12
Ban co muon tim tiep khong(C/K)? k
```

```
-----
Process exited after 132.3 seconds with return value 0
Press any key to continue . . .
```

Ví dụ 3: Tìm USCLN của hai số nguyên a và b

```
int uscln(int a, int b) {  
    while(b != 0)  
    {  
        int t = b;  
        b = a%b;  
        a = t;  
    }  
    return a;  
}
```

```
Cho so nguyen a = 15  
Cho so nguyen b = 13  
Ket qua UCLN (a,b) = 1  
Ban co muon tim tiep khong(C/K)? c  
Cho so nguyen a = 24  
Cho so nguyen b = 35  
Ket qua UCLN (a,b) = 1  
Ban co muon tim tiep khong(C/K)? c  
Cho so nguyen a = 72  
Cho so nguyen b = 81  
Ket qua UCLN (a,b) = 9  
Ban co muon tim tiep khong(C/K)? k
```

```
-----  
Process exited after 26.3 seconds with return value 0  
Press any key to continue . . .
```

Ví dụ 3

Tìm ƯSCLN của dãy số nguyên $A[0]$,

$A[1], \dots, A[n-1]$ gồm n phần tử

```
So phan tu n = 4
```

```
A[0] = 24
```

```
A[1] = 36
```

```
A[2] = 72
```

```
A[3] = 90
```

```
Ket qua USCLN cua day so: 6
```

```
-----
```

```
Process exited after 10.19 seconds with return value 0
```

```
Press any key to continue . . .
```


Ví dụ 3: Tìm ƯSCLN của dãy số

```
1. #include <iostream>
2. #include <cmath>
3. using namespace std;

4. int uscln(int a, int b) {
5.     while (a != b){
6.         if(a > b)
7.             a = a - b;
8.         else
9.             b = b - a;
10.    }
11.    return a;
12. }

13. int main() {
14.     int A[100], n;
15.     cout<<"So phan tu n = ";
16.     cin>>n;
```

```
17.     for(int i=0;i<n;i++)
18.     {
19.         cout<<"A["<<i<<" ] = ";
20.         cin>>A[i];
21.     }
22.     int r = A[0];
23.     for(int i = 1; i < n; i++)
24.         r = uscln(r,A[i]);
25.     cout<<"Ket qua USCLN cua
    day so: "<<r;
26.     return 0;
27. }
```

Ví dụ 3: Tìm ƯSCLN của dãy số

```
1. #include <iostream>
2. #include <cmath>
3. using namespace std;

4. int uscln(int a, int b) {
5.     while (a != b){
6.         if(a > b)
7.             a = a - b;
8.         else
9.             b = b - a;
10.    }
11.    return a;
12. }

13. int main() {
14.     int n, A[100];
15.     cout<<"So phan tu n = ";
16.     cin>>n;
```

```
17.     for(int i=0;i<n;i++)
18.     {
19.         cout<<"A["<<i<<" ] = ";
20.         cin>>A[i];
21.     }
22.     int r = A[0];
23.     for(int i = 1; i< n; i++)
24.         r = uscln(r, A[i]);
25.     cout<<"Ket qua USCLN cua
    day so: "<<r;
26.     return 0;
27. }
```

Ví dụ 3: Tìm ƯSCLN của dãy số

```
int uscln(int a, int b) {  
    while (a != b){  
        if(a > b)  
            a = a - b;  
        else  
            b = b - a;  
    }  
    return a;  
}
```

```
int main() {  
    int A[100], n;  
    cout<<"So phan tu n = "; cin>>n;  
    for(int i=0;i<n;i++)  
    {  
        cout<<"A["<<i<<" ] = ";cin>>A[i];  
    }  
    int r = A[0];  
    for(int i = 1; i < n; i++)  
        r = uscln(r,A[i]);  
    cout<<"Ket qua USCLN cua day so: "<<r;  
    return 0;  
}
```

So phan tu n = 4

A[0] = 24

A[1] = 36

A[2] = 72

A[3] = 90

Ket qua USCLN cua day so: 6

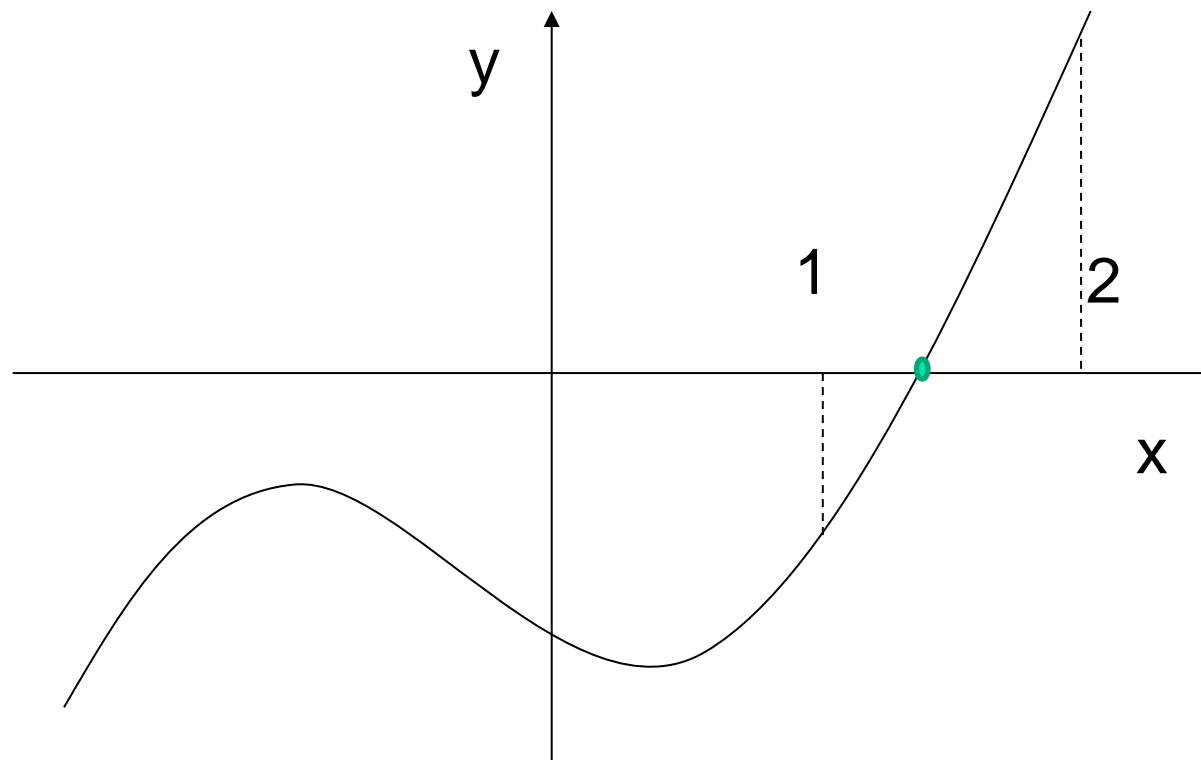
Process exited after 10.19 seconds with return value 0

Press any key to continue . . .

Ví dụ 4

Giải phương trình $f(x) = 0$ trên đoạn $[a,b]$

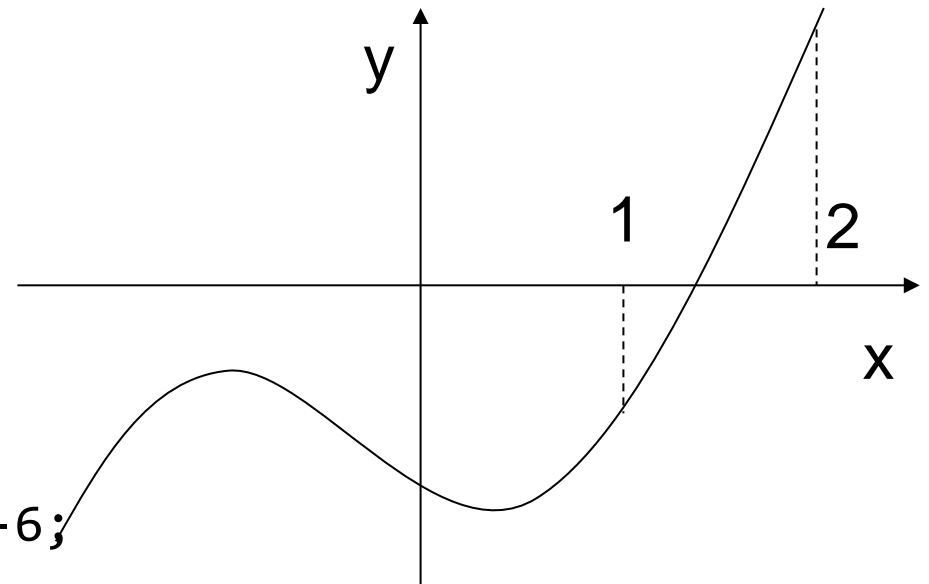
Ví dụ: Giải phương trình $x^3 - x - 1 = 0$



Ví dụ 4: Giải phương trình $f(x) = 0$ trên đoạn $[a,b]$

Giải phương trình $x^3 - x - 1 = 0$

```
1. #include <iostream>
2. #include <cmath>
3. using namespace std;
4. float f(float x) {
5.     return x*x*x-x-1;
6. }
7. int main() {
8.     float a = 1.0, b = 2.0, eps = 1.0e-6;
9.     do {
10.         float c = (a+b)/2;
11.         if (f(a) * f(c) < 0)
12.             b = c;
13.         else a = c;
14.     }while(fabs(b-a) > eps); //fabs: hàm trị tuyệt đối
15.     cout<<"Nghiem phuong trinh x = "<<(b+a)/2;
16.     return 0;
17. }
```

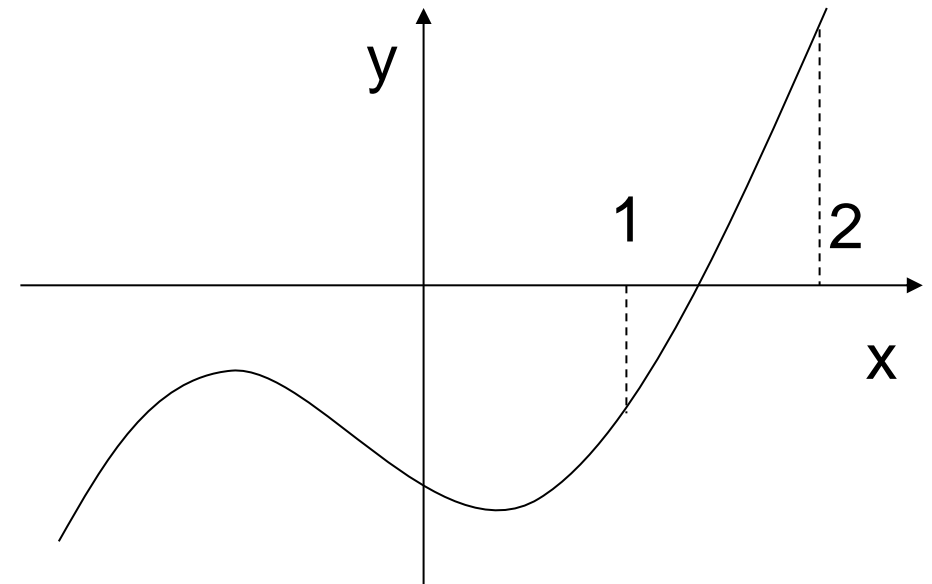


```
Nghiem phuong trinh x = 1.32472
-----
Process exited after 0.2791 seconds with return value 0
Press any key to continue . . .
```

Ví dụ 3: Giải phương trình $f(x) = 0$ trên đoạn $[a,b]$

Giải phương trình $x^3 - x - 1 = 0$

```
1. #include <iostream>
2. #include <cmath>
3. using namespace std;
4. float f(float x) {
5.     return x*x*x-x-1;
6. }
7. int main() {
8.     float a = 1.0, b = 2.0, eps = 1.0e-6;
9.     do {
10.         float c = (a+b)/2;
11.         if (f(a) * f(c) < 0)
12.             b = c;
13.         else a = c;
14.     }while(fabs(b-a) > eps); //fabs: hàm trị tuyệt đối
15.     cout<<"Nghiem phuong trinh x = "<<(b+a)/2;
16.     return 0;
17. }
```



```
Nghiem phuong trinh x = 1.32472
-----
Process exited after 0.2791 seconds with return value 0
Press any key to continue . . .
```



Ví dụ 4

Đọc tọa độ 3 điểm A, B, C và đưa ra diện tích $\triangle ABC$ nếu thỏa mãn.

```
Toa do A(x,y): 0 0
Toa do B(x,y): 2 0
Toa do C(x,y): 0 2
Dien tich tam giac ABC la S = 2
-----
Process exited after 9.095 seconds with return value 0
Press any key to continue . . .
```



Ví dụ 4

Đọc tọa độ 3 điểm A, B, C và đưa ra diện tích ΔABC

```
#include <iostream>
#include <cmath>
using namespace std;
float kc(float Ax, float Ay, float Bx, float By)
{
    return sqrt(pow(Ax - Bx, 2) + pow(Ay - By, 2));
}
```


Ví dụ 4 Đọc tọa độ 3 điểm A, B, C và đưa ra diện tích ΔABC

```
int main(){
    float Ax,Ay, Bx, By, Cx, Cy, p, S;
    cout<<"Toa do A(x,y): "; cin>>Ax>>Ay;
    cout<<"Toa do B(x,y): "; cin>>Bx>>By;
    cout<<"Toa do C(x,y): "; cin>>Cx>>Cy;
    float AB = kc(Ax,Ay,Bx,By), BC = kc(Bx,By,Cx,Cy), AC =
        kc(Cx,Cy,Ax,Ay);
    p = (AB + BC + AC)/2;
    S = sqrt(p*(p-AB)*(p-BC)*(p-AC));
    cout<<"Dien tich tam giac ABC la S = "<<S;
    return 0;
}
```

Ví dụ 4: Đọc tọa độ 3 điểm A, B, C và đưa ra diện tích ΔABC

```
int main(){
    float Ax,Ay, Bx, By, Cx, Cy, p, S;
    cout<<"Toa do A(x,y): "; cin>>Ax>>Ay;
    cout<<"Toa do B(x,y): "; cin>>Bx>>By;
    cout<<"Toa do C(x,y): "; cin>>Cx>>Cy;
    float AB = kc(Ax,Ay,Bx,By), BC = kc(Bx,By,Cx,Cy), AC = kc(Cx,Cy,Ax,Ay);
    if(AB+BC>AC && AB+AC>BC && AC+BC>AB)
    {
        p = (AB + BC + AC)/2;
        S = sqrt(p*(p-AB)*(p-BC)*(p-AC));
        cout<<"Dien tich tam giac ABC la S = "<<S;
    }
    else
        cout<<"3 diem A,B,C tao thanh 1 tam giac!";
    return 0;
}
```



Chương 1: Hàm Phạm vi của biến

- Phạm vi:
 - Khối lệnh, chương trình con, chương trình chính
- Biến chỉ có tác dụng trong phạm vi được khai báo
- Trong cùng một phạm vi các biến phải có tên khác nhau.

Tình huống

- Trong hai phạm vi khác nhau có hai biến cùng tên. Trong đó một phạm vi này nằm trong phạm vi kia?



Phạm vi của biến

```
#include <iostream>
#include <cmath>
using namespace std;
int i;
int binhphuong(int x){
    int y;
    y = x * x;
    return y;
}
int main(){
    int y;
    for (i = 0; i <= 10; i++)
    {
        y = binhphuong(i);
        cout<<"i: "<<i<<"\t i^2:"<< y <<endl;
    }
    return 0;
}
```

Phạm vi của biến

```
#include <iostream>
using namespace std;
int i;
int binhphuong(int x){
    int y;
    y = x * x;
    return y;
}
int main(){
    int y;
    for (i = 0; i <= 10; i++)
    {
        y = binhphuong(i);
        cout<<"i: "<<i<<"\t i^2: "<<y<<endl;
    }
    return 0;
}
```

```
i: 0      i^2: 0
i: 1      i^2: 1
i: 2      i^2: 4
i: 3      i^2: 9
i: 4      i^2: 16
i: 5      i^2: 25
i: 6      i^2: 36
i: 7      i^2: 49
i: 8      i^2: 64
i: 9      i^2: 81
i: 10     i^2: 100
```

Process exited after 0.1547 seconds
Press any key to continue . . .



Phân loại biến

- Biến toàn cục:
 - Biến được khai báo trong chương trình chính, được đặt sau khai báo tệp tiêu đề.
- Biến cục bộ:
 - Biến được khai báo trong lệnh khối hoặc chương trình con, được đặt trước các câu lệnh.

Ghi chú

- Hàm main() cũng là một chương trình con nhưng là nơi chương trình được bắt đầu
- Biến khai báo trong hàm main() cũng là biến cục bộ, chỉ có phạm vi trong hàm main().



Biến static

- Biến cục bộ ra khỏi phạm vi thì bộ nhớ dành cho biến được giải phóng
- Yêu cầu lưu trữ giá trị của biến cục bộ một cách lâu dài => sử dụng từ khóa *static*

Cú pháp:

static <kiểu_dữ_liệu> tên_biến;



Ví dụ → Kết quả

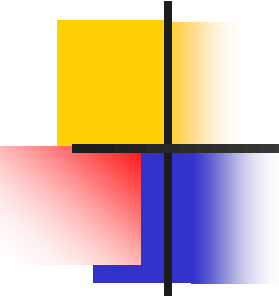
```
#include <iostream>
using namespace std;
void fct() {
    int count = 1;
    cout<<"Day la lan goi ham fct lan thu "<< count++ <<endl;
}
int main() {
    int i;
    for (i = 0; i < 10; i++) fct();
    return 0;
}
```


Ví dụ → Kết quả

```
#include <iostream>
using namespace std;
void fct() {
    int count = 1;
    cout<<"Day la lan goi ham fct lan thu "<< count++ <<endl;
}
int main() {
    int i;
    for (i = 0; i < 10; i++) fct();
    return 0;
}
```

```
Day la lan goi ham fct lan thu 1
Day la lan goi ham fct lan thu 1
Day la lan goi ham fct lan thu 1
Day la lan goi ham fct lan thu 1
Day la lan goi ham fct lan thu 1
Day la lan goi ham fct lan thu 1
Day la lan goi ham fct lan thu 1
Day la lan goi ham fct lan thu 1
Day la lan goi ham fct lan thu 1
Day la lan goi ham fct lan thu 1
```

```
-----
Process exited after 0.2459 seconds
Press any key to continue . . .
```



Ví dụ → Kết quả

```
#include <iostream>
using namespace std;
void fct() {
    static int count = 1;
    cout<<"Day la lan goi ham fct lan thu "<< count++ <<endl;
}
int main() {
    int i;
    for (i = 0; i < 10; i++)
        fct();
    return 0;
}
```

Ví dụ → Kết quả

```
#include <iostream>
using namespace std;
void fct() {
    static int count = 1;
    cout<<"Day la lan goi ham fct lan thu "<< count++ <<endl;
}
int main() {
    int i;
    for (i = 0; i < 10; i++)
        fct();
    return 0;
}
```

```
Day la lan goi ham fct lan thu 1
Day la lan goi ham fct lan thu 2
Day la lan goi ham fct lan thu 3
Day la lan goi ham fct lan thu 4
Day la lan goi ham fct lan thu 5
Day la lan goi ham fct lan thu 6
Day la lan goi ham fct lan thu 7
Day la lan goi ham fct lan thu 8
Day la lan goi ham fct lan thu 9
Day la lan goi ham fct lan thu 10
```

```
-----
Process exited after 0.2104 seconds
Press any key to continue . . .
```



Biến register

- Thanh ghi có tốc độ truy cập nhanh hơn RAM, bộ nhớ ngoài
- Lưu biến trong thanh ghi sẽ tăng tốc độ thực hiện chương trình

Cú pháp

register <kiểu_dữ_liệu> tên_biến;

Lưu ý:

- số lượng biến register không nhiều và thường chỉ với kiểu dữ liệu nhỏ như int, char



Thủ tục (hàm không trả về giá trị)

- Dùng từ khóa **void** để thay cho kiểu trả về của hàm
- Trong thân hàm không có lệnh **return giá trị**
- Không được gọi trong câu lệnh ghi ra màn hình **cout**, hay gán giá trị cho biến



Thủ tục (hàm không trả về giá trị)

```
#include <iostream>
using namespace std;

void trungBinh(double x, double y)
{
    double s = x + y;
    cout<<s/2;
}

int main () {
    double x, y;
    cout<<"Nhap gia tri x, y:"; cin>>x>>y;
    trungBinh(x,y);
    //Loi goi ham trungBinh
    return 0;
}
```



Trường hợp hàm không có tham số

- Không cần khai báo tham số trong định nghĩa hàm và khai báo nguyên mẫu hàm
- Ví dụ:

```
void hienThi()  
{  
    cout<<"Chao mung ban den voi mon LTNC!";  
}
```

Trường hợp hàm không có tham số

▪ Ví dụ:

```
#include <iostream>
using namespace std;

void hienThi()
{
    cout<<"Chao cac ban K62."<<endl;
    cout<<"Chuc mung cac ban den voi mon LTNC!";
}

int main () {
    hienThi();
    return 0;
}
```

```
Chao cac ban K62.
Chuc mung cac ban den voi mon LTNC!
-----
Process exited after 0.2048 seconds with return code 0
Press any key to continue . . .
```




Khai báo với giá trị mặc định

- Truyền giá trị mặc định khi định nghĩa hàm
 - Giá trị mặc định được sử dụng khi không truyền tham số trong lời gọi hàm
- Định nghĩa hàm:

```
kieutrave tenham (kieudulieu1 thamso1 = giatri,  
                  kieudulieu2 thamso2 = giatri,...)  
{  
    //các câu lệnh xử lý  
}
```



Khai báo với giá trị mặc định

▪ Ví dụ:

```
#include <iostream>
using namespace std;
int phepChia(int a, int b = 2)
{
    int r;
    r=a/b;
    return r;
}
int main ()
{
    cout << phepChia(12);//goi ham voi gia tri mac dinh
    cout << endl<<phepChia(20,4);
    return 0;
}
```



Truyền theo trị và truyền theo biến

- **Truyền theo trị**
 - Dựa trên nguyên tắc truyền những bản sao của biến được truyền
 - Những câu lệnh thay đổi giá trị tham số hình thức sẽ không ảnh hưởng tới biến được truyền
- **Truyền theo biến**
 - Tham số được truyền sẽ thực sự là biến và các thao tác sẽ thi hành trực tiếp với biến
 - Những câu lệnh thay đổi giá trị tham số hình thức sẽ ảnh hưởng tới biến được truyền



Truyền tham trị

- Tham số hình thức: là tham số khai báo trong phần khai báo hàm
- Tham số thực: là tham số được truyền vào trong lời gọi hàm
 - Truyền bằng tham trị
 - Truyền bằng tham chiếu
- **Truyền bằng tham trị:**
 - ✓ Tạo một bản sao giá trị của tham số gán cho tham số hình thức
 - ✓ Không làm thay đổi giá trị của biến truyền vào



Truyền tham trị

```
#include <iostream>
using namespace std;
void swap(int a, int b) {
    int x = a;
    a = b;
    b = x;
}
int main() {
    int a = 5, b = 100;
    cout<<"Truoc: a = "<<a<<"", b = "<<b<<endl;
    swap(a,b);
    cout<<"Sau: a = "<<a<<"", b = "<<b;
    return 0;
}
```

Ví dụ

```
#include <iostream>
using namespace std;

void swap(int a, int b) {
    int x = a;
    a = b;
    b = x;
}

int main() {
    int a = 5, b = 100;
    cout<<"Truoc: a = "<<a<<"", b = "<<b<<endl;
    swap(a, b);
    cout<<"Sau: a = "<<a<<"", b = "<<b;
    return 0;
```

```
Truoc: a = 5, b = 100
Sau: a = 5, b = 100
-----
Process exited after 0.1444 seconds
Press any key to continue . . .
```

Truyền tham chiếu

▪ Truyền bằng tham chiếu:

- ✓ Lời gọi hàm sẽ truyền trực tiếp tham số
- ✓ Làm thay đổi giá trị của biến truyền vào

▪ Nguyên mẫu hàm truyền tham chiếu:

```
kietrave tenham (kieudulieu &, kieudulieu &,....);
```

▪ Định nghĩa hàm truyền tham chiếu:

```
kietrave tenham (kieudulieu &thamso1, kieudulieu &thamso2,..)  
{  
    //các câu lệnh xử lý  
}
```



Địa chỉ của biến

```
#include <iostream>
using namespace std;

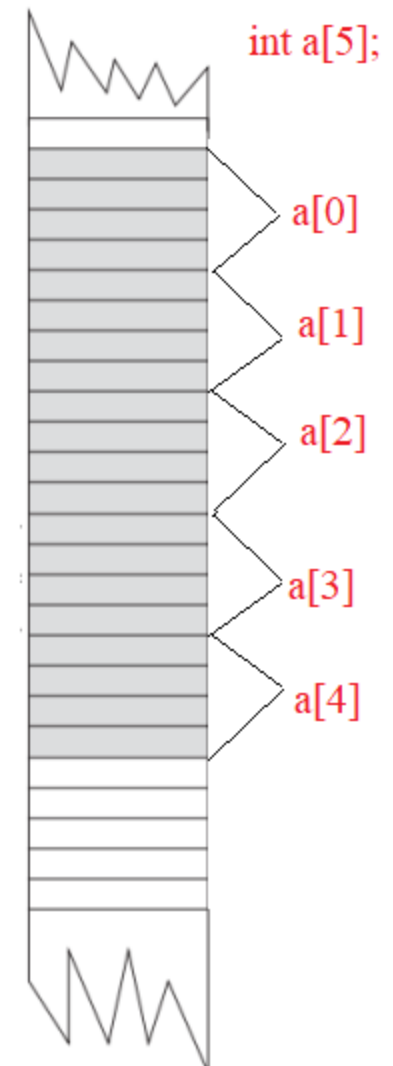
int main(){
    int x,y;
    int a[5] = {1,0,3,2,5};
    cout<<"Nhap so nguyen x = "; cin>>x;
    cout<<"Dia chi bo nho cua x: "<<&x;
    cout<<"\nNhap so nguyen y = "; cin>>y;
    cout<<"Dia chi bo nho cua y: "<<&y<<endl;
    for(int i =0; i<5;i++)
        cout<<"a["<<i<<"] = "<<a[i]<<"\t Dia chi bo nho
cua a["<<i<<"] = "<<&a[i]<<endl;
    return 0;
}
```


Địa chỉ của biến

```
Nhap so nguyen x = 12
Dia chi bo nho cua x: 0x6ffe08
Nhap so nguyen y = -12
Dia chi bo nho cua y: 0x6ffe04
a[0] = 1      Dia chi bo nho cua a[0] = 0x6ffdf0
a[1] = 0      Dia chi bo nho cua a[1] = 0x6ffdf4
a[2] = 3      Dia chi bo nho cua a[2] = 0x6ffdf8
a[3] = 2      Dia chi bo nho cua a[3] = 0x6ffdfc
a[4] = 5      Dia chi bo nho cua a[4] = 0x6ffe00
-----
Process exited after 6.389 seconds with return value 0
Press any key to continue . . .
```

Địa chỉ
của a[0]

0x6ffdf0



Truyền tham chiếu

▪ Ví dụ:

```
#include <iostream>
using namespace std;
void swap(int &a, int &b) {
    int x = a;
    a = b;
    b = x;
}
int main() {
    int a = 5, b = 100;
    cout<<"Truoc: a = "<<a<<", b = "<<b<<endl;
    swap(a,b);
    cout<<"Sau: a = "<<a<<", b = "<<b;
    return 0;
}
```

Truyền tham chiếu

▪ Ví dụ:

```
Truoc: a = 5, b = 100  
Sau: a = 100, b = 5
```

```
-----  
Process exited after 0.1392 second  
Press any key to continue . . .
```

```
#include <iostream>  
using namespace std;  
void swap(int &a, int &b) {  
    int x = a;  
    a = b;  
    b = x;  
}  
int main() {  
    int a = 5, b = 100;  
    cout<<"Truoc: a = "<<a<<", b = "<<b<<endl;  
    swap(a,b);  
    cout<<"Sau: a = "<<a<<", b = "<<b;  
    return 0;  
}
```

Dẫn trở

```
Truoc: a = 5, b = 100  
Sau: a = 100, b = 5
```

```
-----  
Process exited after 0.1842 seconds with return value 0  
Press any key to continue . . .
```

```
#include <iostream>  
using namespace std;  
  
void swap(int *pa, int *pb)  
{  
    int x = *pa;  
    *pa = *pb;  
    *pb = x;  
}  
  
int main() {  
    int a = 5, b = 100;  
    cout<<"Truoc: a = "<<a<<"", b = "<<b<<endl;  
    swap(a,b);  
    cout<<"Sau: a = "<<a<<"", b = "<<b;  
    return 0;  
}
```



Truyền theo trị và truyền theo biến

	Tham trị	Tham chiếu	Dẫn trở
Khai báo đối	<code>void swap(int x, int y)</code>	<code>void swap(int &x, int &y)</code>	<code>void swap(int *x, int *y)</code>
Câu lệnh	<code>t = x; x = y; y = t;</code>	<code>t = x; x = y; y = t;</code>	<code>t = *x; *x = *y; *y = t;</code>
Lời gọi	<code>swap(a, b);</code>	<code>swap(a, b);</code>	<code>swap(&a, &b);</code>
Tác dụng	a, b không thay đổi	a, b có thay đổi	a, b có thay đổi

Truyền tham trị, tham chiếu

Ví dụ: Viết chương trình tính diện tích của tam giác ABC khi biết 3 đỉnh A, B, C

Yêu cầu:

- Viết hàm tính khoảng cách giữa 2 điểm A và B

`double kc(double Ax, double Ay, double Bx, double By)`

- Viết hàm tính diện tích của tam giác ABC sử dụng biến diện tích S theo tham chiếu (Nếu ABC không thỏa mãn đk là tam giác thì diện tích $S = 0$)

`void dienTich(double Ax, double Ay, double Bx, double By, double Cx, double Cy, double &S)`

```
Toa do A(x,y): 1 1
Toa do B(x,y): 3 1
Toa do C(x,y): 3 2
Dien tich tam giac ABC la S = 1
-----
Process exited after 27.65 seconds with return value 0
Press any key to continue . . .
```

Truyền tham trị, tham chiếu

- Ví dụ: Viết chương trình tính diện tích của tam giác ABC khi biết 3 đỉnh A, B, C

```
#include <iostream>
#include <cmath>
using namespace std;
double kc(double Ax, double Ay, double Bx, double By)
{
    return sqrt(pow(Ax - Bx, 2) + pow(Ay - By, 2));
}
void dienTich(double Ax, double Ay, double Bx, double By,
double Cx, double Cy, double &S)
{
    double p, AB = kc(Ax,Ay,Bx,By), BC = kc(Bx,By,Cx,Cy), AC =
kc(Cx,Cy,Ax,Ay);
    if(AB+BC>AC && AB+AC>BC && AC+BC>AB)
    {
        p = (AB + BC + AC)/2;
        S = sqrt(p*(p-AB)*(p-BC)*(p-AC));
    }
    else
        S = 0;
}
```

Truyền tham trị, tham chiếu

- Ví dụ: Viết chương trình tính diện tích của tam giác ABC khi biết 3 đỉnh A,B,C

```
int main(){
    double Ax,Ay, Bx, By, Cx, Cy, dt;
    cout<<"Toa do A(x,y): "; cin>>Ax>>Ay;
    cout<<"Toa do B(x,y): "; cin>>Bx>>By;
    cout<<"Toa do C(x,y): "; cin>>Cx>>Cy;
    dienTich(Ax, Ay, Bx, By, Cx, Cy, dt);
    cout<<"Dien tich tam giac ABC la S = "<<dt;
    return 0;
}
```


Nạp chồng hàm

- Đây là một kỹ thuật cho phép sử dụng cùng một tên gọi cho các hàm "giống nhau" (cùng mục đích) nhưng xử lý trên các kiểu dữ liệu khác nhau hoặc trên số lượng dữ liệu khác nhau.
- Ví dụ hàm sau tìm số lớn nhất trong 2 số nguyên:

```
int max(int a, int b)
{
    return (a > b)? a:b ;
}
```

- Nếu đặt $c = \text{max}(3,5)$ ta sẽ có $c = 5$. Tuy nhiên cũng tương tự như vậy nếu đặt $c = \text{max}(3.0, 5.0)$ chương trình sẽ bị lỗi vì các giá trị (float) không phù hợp về kiểu (int) của đối trong hàm max.

→ Trong trường hợp như vậy chúng ta phải viết hàm mới để tính max của 2 số thực.



Nạp chồng hàm (tiếp)

- ❑ Trong C và các NNLT cổ điển khác chúng ta buộc phải sử dụng một tên mới cho hàm "mới" này.
- ❑ Ví dụ:
 - ❑ `float fmax(float a, float b) {return (a > b) ? a: b ; }`
 - ❑ `char cmax(char a, char b) {return (a > b) ? a: b ; }`
 - ❑ `long lmax(long a, long b) {return (a > b) ? a: b ; }`
 - ❑ `double dmax(double a, double b) {return (a > b) ? a: b ; }`
- ❑ Việc sử dụng tên như vậy sẽ gây bất lợi khi cần gọi hàm.
- ❑ Ngôn ngữ C++ cho phép ta có thể khai báo và định nghĩa cả 5 hàm trên với cùng 1 tên gọi ví dụ là max chẳng hạn.



Nạp chồng hàm (tiếp)

- ❑ Ngôn ngữ C++ cho phép ta có thể khai báo và định nghĩa cả 5 hàm trên với cùng 1 tên gọi ví dụ là max.
- ❑ Ví dụ:
 - ❑ `int max(int a, int b) {return (a > b)? a:b ; }`
 - ❑ `float max(float a, float b) {return (a > b) ? a: b ; }`
 - ❑ `char max(char a, char b) {return (a > b) ? a: b ; }`
 - ❑ `long max(long a, long b) {return (a > b) ? a: b ; }`
 - ❑ `double max(double a, double b) {return (a > b) ? a: b ; }`
- ❑ Lời gọi hàm bất kỳ dạng nào: `max(3,5)`, `max(3.0,5)`, `max('O', 'K')` đều được thực hiện.
- ❑ Chương trình sẽ dựa vào kiểu của các đối khi gọi để quyết định chạy hàm nào.



Nạp chồng hàm (tiếp)

- ❑ Nhiều hàm có thể được nạp chồng (với cùng tên gọi giống nhau) nếu chúng thoả các điều kiện sau:
 - ❑ Số lượng các tham đối trong hàm là khác nhau, hoặc
 - ❑ Kiểu của tham đối trong hàm là khác nhau.
- ❑ Ví dụ việc vẽ các hình: thẳng, tam giác, vuông, chữ nhật trên màn hình là giống nhau, chúng chỉ phụ thuộc vào số lượng các điểm nối và toạ độ của chúng. Do vậy ta có thể khai báo và định nghĩa 4 hàm vẽ nói trên với cùng chung tên gọi.
- ❑ Ví dụ:

```
void ve( Diem A, Diem B);           //Vẽ đường thẳng
void ve( Diem A, Diem B, Diem C);   // Vẽ tam giác
void ve( Diem A, Diem B, Diem C, Diem D); //Vẽ tứ giác
```
- ❑ Kỹ thuật nạp chồng hàm này còn áp dụng cả cho các toán tử. Trong phần lập trình hướng đối tượng, ta sẽ thấy NSD được phép định nghĩa các toán tử mới nhưng vẫn lấy tên cũ như +, -, *, / ...



Ví dụ: Nạp chồng hàm

```
#include <iostream>
#include <cmath>
using namespace std;
int max(int a, int b) {return (a > b)? a:b; }
float max(float a, float b) {return (a > b)? a:b; }
char max(char a, char b) {return (a > b)? a:b; }
long max(long a, long b) {return (a > b)? a:b; }
double max(double a, double b) {return (a > b)? a:b; }

int main() {
    cout<<"Max(3,5) = "<<max(3,5);
    cout<<"\nMax(3.5,5.7) = "<<max(3.5,5.7);
    cout<<"\nMax('B','E') = "<<max('B','E');
    return 0;
}
```

Ví dụ: Nạp chồng hàm

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4 int max(int a, int b) {return (a > b)? a:b; }
5 float max(float a, float b) {return (a > b)? a:b; }
6 char max(char a, char b) {return (a > b)? a:b; }
7 long max(long a, long b) {return (a > b)? a:b; }
8 double max(double a, double b) {return (a > b)? a:b; }
9
10 int main() {
11     cout<<"Max(3,5) = "<<max(3,5);
12     cout<<"\nMax(3.5,5.7) = "<<max(3.5,5.7);
13     cout<<"\nMax('B','E') = "<<max('B','E');
14     return 0;
15 }
```

```
Max(3,5) = 5
Max(3.5,5.7) = 5.7
Max('B','E') = E
-----
Process exited after 0.2073 seconds with return value 0
Press any key to continue . . .
```



Hàm đệ quy

- Khái niệm đệ quy
 - Trong thực tế, ta thường gặp những đối tượng bao gồm chính nó hoặc được định nghĩa dưới dạng của chính nó. Ta nói đối tượng đó được xác định một cách đệ quy.
 - Một khái niệm X gọi là định nghĩa theo đệ quy nếu trong định nghĩa X có sử dụng ngay chính khái niệm X.
- Hàm đệ quy: Các hàm đệ quy được xác định phụ thuộc vào biến nguyên không âm n theo sơ đồ sau:
 - Bước cơ sở: Xác định giá trị của hàm tại $n = 0$: $f(0)$
 - Bước đệ quy: Cho giá trị của $f(k)$, $k \leq n$, đưa ra quy tắc tính giá trị của $f(n+1)$

Ví dụ: $f(0) = 3$, $f(n+1) = 2f(n)+3$ với $n > 0$
- Hàm đệ quy gọi lại chính nó trong quá trình thực hiện
 - Đệ quy trực tiếp: Gọi lại chính nó ngay trong thân hàm
 - Đệ quy gián tiếp: gọi lại chính nó thực hiện trong các hàm con



Thuật toán đệ quy

- Một thuật toán được gọi là **đệ quy** nếu nó giải bài toán bằng cách rút gọn liên tiếp bài toán ban đầu tới bài toán cũng như vậy nhưng có dữ liệu đầu vào nhỏ hơn.
 - Ví dụ 1: Thuật toán tính giai thừa:
$$f(0) = 0! = 1, f(n) = n! = n * (n-1)! \text{ với } n > 0$$
 - Ví dụ 2: Thuật toán tìm dãy số Fibonacci:
$$0, 1, 1, 2, 3, 5, 8, \dots$$

$$F(0) = 0, F(1) = 1$$

$$F(n) = F(n - 2) + F(n - 1) \text{ với mọi } n \geq 2$$



Lớp các bài toán giải được = Đệ quy

- Phương pháp đệ quy thường được dùng để giải các bài toán có đặc điểm:
 - Giải quyết được dễ dàng trong các trường hợp riêng gọi là trường hợp *suy biến* hay *cơ sở*, trong trường hợp này hàm được tính bình thường mà không cần gọi lại chính nó,
 - Đối với trường hợp *tổng quát*, bài toán có thể giải được bằng bài toán cùng dạng nhưng với tham đối khác có kích thước nhỏ hơn tham đối ban đầu. Và sau một số bước hữu hạn biến đổi cùng dạng, bài toán đưa được về trường hợp suy biến.
- Một lớp rất rộng của bài toán dạng này là các bài toán có thể định nghĩa được dưới dạng đệ quy như các bài toán lặp với số bước hữu hạn biết trước, các bài toán UCLN, tháp Hà Nội, ...



Cấu trúc chung của hàm đệ quy

- Dạng thức chung của một chương trình đệ quy thường như sau: Dạng thức chung của một chương trình đệ quy thường như sau:

if (trường hợp suy biến)

{

trình bày cách giải //giả định đã có cách giải

}

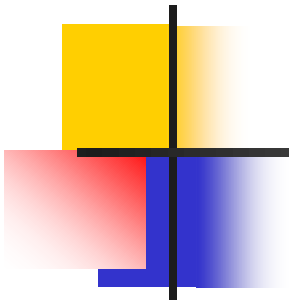
else //trường hợp tổng quát

{

gọi lại hàm với tham số "nhỏ" hơn

}

Ví dụ 1: Cho biết kết quả thực hiện chương trình



```
#include <iostream>
using namespace std;
int fun(int n){
    if(n==0)
        return 1;
    else
        return n*fun(n-1);
}
int main(){
    cout<<fun(5)<<endl;
    return 0;
}
```

fun(5)

5 * fun(4)

4 * fun(3)

3 * fun(2)

2 * fun(1)

1 * fun(0)

1

120

Ví dụ 1: Tìm $n!$ theo đệ quy:

$0! = 1,$

$n! = n * (n-1)!$

```
#include <iostream>
using namespace std;
int gt(int n){
    if(n==0)
        return 1;
    else
        return n*gt(n-1);
}
int main(){
    int n;
    cout<<"Cho so nguyen n = "; cin>>n;
    cout<<n<<"! = "<<gt(n);
    return 0;
}
```

Ví dụ 1: Tìm $n!$ theo đệ quy: $0! = 1, n! = n*(n-1)!$

```
#include <iostream>
using namespace std;
int gt(int n){
    if(n==0)
        return 1;
    else
        return n*gt(n-1);
}
int main(){
    int n;
    char c='C';
    while(toupper(c) == 'C')
    {
```



```
        cout<<"Cho so nguyen n =
"; cin>>n;
        cout<<n<<"!="<<gtt(n);
        cout<<"\nBan co tim tiep
khong? (C/K)"; cin>>c;
    }
    return 0;
}
```

```
Cho so nguyen n = 16
16! = 2004189184
Ban co tim tiep khong? (C/K) c
Cho so nguyen n = 17
17! = -288522240
Ban co tim tiep khong? (C/K)
```

Ví dụ 1: Tìm $n!$ theo đệ quy: $0! = 1, n! = n * (n-1)!$

```
#include <iostream>
using namespace std;
unsigned long int gt(int n){
    if(n==0)
        return 1;
    else
        return n*gt(n-1);
}
int main(){
    int n;
    char c='C';
    while(toupper(c) == 'C')
    {
```

```
        cout<<"Cho so nguyen n =
"; cin>>n;
        cout<<n<<"! = "<<gt(n);
        cout<<"\nBan co tim tiep
khong? (C/K) "; cin>>c;
    }
    return 0;
}
```

?

```
Cho so nguyen n = 30
30! = 1409286144
Ban co tim tiep khong? (C/K) c
Cho so nguyen n = 40
40! = 0
```

Ví dụ 2: Tính tổ hợp chập k của n theo công thức hồi quy: $C(k,n) = C(k-1,n-1) + C(k,n-1)$

```
#include <iostream>
using namespace std;
int C(int k, int n){
    if(k == n || k == 0)
        return 1;
    else
        return C(k-1,n-1) + C(k,n-1);
}
int main(){
    int n,k;
    cout<<"Cho n = ";cin>>n;
    cout<<"Cho k = ";cin>>k;
    cout<<"To hop C(k,n) = "<<C(k,n);
    return 0;
}
```

Ví dụ 2: Tính tổ hợp chập k của n theo công thức hồi quy: $C(k,n) = C(k-1,n-1) + C(k,n-1)$

```
#include <iostream>
using namespace std;
int C(int k, int n){
    if(k == n || k == 0)
        return 1;
    else
        return C(k-1,n-1) + C(k,n-1);
}
int main(){
    int n,k;
    cout<<"Cho n = ";cin>>n;
    cout<<"Cho k = ";cin>>k;
    cout<<"To hop C(k,n) = "<<C(k,n);
    return 0;
}
```

```
Cho n = 5
Cho k = 3
To hop C(k,n) = 10
-----
Process exited after 7.012 seconds with return value 0
Press any key to continue . . .
```


Ví dụ 3: Viết chương trình UCLN của 2 số nguyên a và b có sử dụng đệ quy

```
int ucln(int a, int b) {  
    if (a == 0 || b == 0)  
        return a+b;  
    else  
        if(a>b)  
            return ucln(b, a%b);  
        else  
            return ucln(a, b%a);  
}
```

Ví dụ 3: Viết chương trình UCLN của 2 số nguyên a và b có sử dụng đệ quy

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4 int uscln(int a, int b) {
5     if (a == 0 || b == 0)
6         return a+b;
7     else
8         if(a>b) return uscln(b,a%b);
9         else return uscln(a,b%a);
10 }
11 int main() {
12     int a,b;
13     char c;
14     do{
15         cout<<"Cho so nguyen duong a = "; cin>>a;
16         cout<<"Cho so nguyen duong b = "; cin>>b;
17         cout<<"Ket qua UCLN(a,b) = "<<uscln(a,b);
18         cout<<"\nBan co muon tim tiep khong(C/K)? "; cin>>c;
19     }while (c == 'C' || c == 'c');
20     return 0;
21 }
```

```
Cho so nguyen duong a = 0
Cho so nguyen duong b = 12
Ket qua UCLN(a,b) = 12
Ban co muon tim tiep khong(C/K)? C
Cho so nguyen duong a = 35
Cho so nguyen duong b = 0
Ket qua UCLN(a,b) = 35
Ban co muon tim tiep khong(C/K)? c
Cho so nguyen duong a = 45
Cho so nguyen duong b = 60
Ket qua UCLN(a,b) = 15
Ban co muon tim tiep khong(C/K)? k

-----
Process exited after 42.16 seconds with return value 0
Press any key to continue . . .
```

Ví dụ 4

Viết chương trình tìm số Fibonacci thứ n sử dụng đệ quy:

$$F(0) = 0, F(1) = 1$$

$$F(n) = F(n - 2) + F(n - 1) \text{ với mọi } n \geq 2$$

```
Cho so nguyen n = -12
Nhap sai! So n phai nguyen duong.
Ban co muon tim tiep khong?(C/K)C
Cho so nguyen n = 6
So Fibonacci thu 6 la: 8
Ban co muon tim tiep khong?(C/K)c
Cho so nguyen n = 10
So Fibonacci thu 10 la: 55
Ban co muon tim tiep khong?(C/K)k

-----
Process exited after 24.44 seconds with return value 0
Press any key to continue . . .
```

Ví dụ 4: Viết chương trình tìm số Fibonacci thứ n sử dụng đệ quy:

$$F(0) = 0, F(1) = 1$$

$$F(n) = F(n - 2) + F(n - 1) \text{ với mọi } n \geq 2$$

```
#include <iostream>
using namespace std;
int Fibo (int n){
    if(n==0) return 0;
    else if(n == 1) return 1;
    else
        return Fibo(n-1)+ Fibo(n-2);
}
```

Ví dụ 4: Viết chương trình tìm số Fibonacci thứ n sử dụng đệ quy:

$$F(0) = 0, F(1) = 1$$

$$F(n) = F(n - 2) + F(n - 1) \text{ với mọi } n \geq 2$$

```
int main(){
    int n;
    char c = 'C';
    while (c == 'C' || c == 'c')
    {
        cout<<"Cho so nguyen n = "; cin>>n;
        if(n>=0)
            cout<<"So Fibonacci thu "<<n<<" la: "<<Fibo(n);
        else
            cout<<"Nhap sai! So n phai nguyen duong.";
        cout<<"\nBan co muon tim tiep khong?(C/K)"; cin>>c;
    }
    return 0;
```

Ví dụ 4: Viết chương trình tìm số Fibonacci thứ n sử dụng đệ quy:

$$F(0) = 0, F(1) = 1$$

$$F(n) = F(n - 2) + F(n - 1) \text{ với mọi } n \geq 2$$

```
#include <iostream>
using namespace std;
int Fibo(int n){
    if(n==0) return 0;
    else if(n == 1) return 1;
    else
        return Fibo(n-1)+ Fibo(n-2);
}
int main(){
    int n;
    char c = 'C';
    while (c == 'C' || c == 'c'){
        cout<<"Cho so nguyen n = "; cin>>n;
        if(n>=0)
            cout<<"So Fibonacci thu "<<n<<" la: "<<Fibo(n);
        else
            cout<<"Nhap sai! So n phai nguyen duong.";
        cout<<"\nBan co muon tim tiep khong?(C/K)"; cin>>c;
    }
    return 0;
}
```

```
Cho so nguyen n = -12
Nhap sai! So n phai nguyen duong.
Ban co muon tim tiep khong?(C/K)C
Cho so nguyen n = 6
So Fibonacci thu 6 la: 8
Ban co muon tim tiep khong?(C/K)c
Cho so nguyen n = 10
So Fibonacci thu 10 la: 55
Ban co muon tim tiep khong?(C/K)k
```

```
-----
Process exited after 24.44 seconds with return value 0
Press any key to continue . . .
```



Khử đệ quy

- Ví dụ 1: Viết chương trình tính $n!$ không sử dụng đệ quy $n! = 1.2....n$

```
#include <iostream>
using namespace std;
int gt(int n){
    int s=1;
    for(int i=1; i<=n;i++)
        s=s*i;
    return s;
}
int main(){
    int n;
    cout<<"Cho so nguyen n = "; cin>>n;
    cout<<n<<"! = "<<gt(n);
    return 0;
}
```

Khử đệ quy: Viết chương trình tìm số Fibonacci thứ n không dùng đệ quy

```
Cho so nguyen n = -12
Nhap sai! So n phai nguyen duong.
Ban co muon tim tiep khong?(C/K)C
Cho so nguyen n = 5
So Fibonacci thu 5 la: 5
Ban co muon tim tiep khong?(C/K)c
Cho so nguyen n = 6
So Fibonacci thu 6 la: 8
Ban co muon tim tiep khong?(C/K)c
Cho so nguyen n = 10
So Fibonacci thu 10 la: 55
Ban co muon tim tiep khong?(C/K)k

-----
Process exited after 33.51 seconds with return value 0
Press any key to continue . . .
```




Khử đệ quy: Viết chương trình tìm số Fibonacci thứ n không dùng đệ quy

```
int Fibo(int n){  
    if(n==0) return 0;  
    else if(n == 1) return 1;  
    else  
    {  
        int f1 = 0, f2 = 1, f3;  
        for(int k = 2; k <= n; k++)  
        {  
            f3 = f1+f2;  
            f1 = f2;  
            f2 = f3;  
        }  
        return f3;  
    }  
}
```



Khử đệ quy: Viết chương trình tìm số Fibonacci thứ n không dùng đệ quy

```
int main(){
    int n;
    char c = 'C';
    while (c == 'C' || c == 'c'){
        cout<<"Cho so nguyen n = "; cin>>n;
        if(n>=0)
            cout<<"So Fibonacci thu "<<n<<" la: "<<Fibo(n);
        else
            cout<<"Nhap sai! So n phai nguyen duong.";
        cout<<"\nBan co muon tim tiep khong?(C/K)"; cin>>c;
    }
    return 0;
}
```

Khử đệ quy: Viết chương trình tìm số Fibonacci thứ n không dùng đệ quy

```
1 #include <iostream>
2 using namespace std;
3 int Fibo(int n){
4     if(n==0) return 0;
5     else if(n == 1) return 1;
6     else
7     {
8         int f1 = 0, f2 = 1, f3;
9         for(int k = 2; k<=n; k++)
10         {
11             f3 = f1+f2;
12             f1 = f2;
13             f2 = f3;
14         }
15         return f3;
16     }
17 }
```

```
18 int main(){
19     int n;
20     char c = 'C';
21     while (c == 'C' || c == 'c'){
22         cout<<"Cho so nguyen n = "; cin>>n;
23         if(n>=0)
24             cout<<"So Fibonacci thu "<<n<<" la: "<<Fibo(n);
25         else
26             cout<<"Nhap sai! So n phai nguyen duong.";
27         cout<<"\nBan co muon tim tiep khong?(C/K)"; cin>>c;
28     }
29     return 0;
30 }
```

```
Cho so nguyen n = -12
Nhap sai! So n phai nguyen duong.
Ban co muon tim tiep khong?(C/K)C
Cho so nguyen n = 5
So Fibonacci thu 5 la: 5
Ban co muon tim tiep khong?(C/K)c
Cho so nguyen n = 6
So Fibonacci thu 6 la: 8
Ban co muon tim tiep khong?(C/K)c
Cho so nguyen n = 10
So Fibonacci thu 10 la: 55
Ban co muon tim tiep khong?(C/K)k
```

```
-----
Process exited after 33.51 seconds with return value 0
Press any key to continue . . .
```

Câu hỏi 1: Kết quả đưa ra màn hình

```
#include <iostream>
using namespace std;
void fun(int n){
    if(n > 0) {
        fun(--n);
        cout<<n;
        fun(--n);
    }
}
int main() {
    fun(3);
    return 0;
}
```

a	0 2 1 0
b	0 1 0 2
c	1 1 2 0
d	0 1 2 0
e	0 2 0 1

Câu hỏi 1: Kết quả đưa ra màn hình

```
#include <iostream>
using namespace std;
void fun(int n){
    if(n > 0) {
        fun(--n);
        cout<<n;
        fun(--n);
    }
}
int main() {
    fun(3);
    return 0;
}
```

a	0 2 1 0
b	0 1 0 2
c	1 1 2 0
d	0 1 2 0
e	0 2 0 1

```
0120
```

```
-----
```

```
Process exited after 0.1028 seconds with return value 0
```

```
Press any key to continue . . .
```

Câu hỏi 2: Kết quả đưa ra màn hình

```
#include <iostream>
using namespace std;
void fun(int &i, int &j) {
    i = i*i;
    j = j*j;
}
int main() {
    int i = 5, j = 2;
    fun(i,j);
    cout<<i<<" , "<<j;
    return 0;
}
```

a	5, 2
b	2, 5
c	10, 4
d	4, 25
e	25, 4

Câu hỏi 2: Kết quả đưa ra màn hình

```
#include <iostream>
using namespace std;
void fun(int &i,int &j) {
    i = i*i;
    j = j*j;
}
int main() {
    int i = 5, j = 2;
    fun(i,j);
    cout<<i<<","<<j;
    return 0;
}
```

a	5, 2
b	2, 5
c	10, 4
d	4, 25
e	25, 4

Câu hỏi 3: Kết quả đưa ra màn hình

```
#include <iostream>
using namespace std;

void fun(char &);
int main() {
    char a[10]="ABCDEF";
    fun(a[0]);
    return 0;
}
void fun(char &a){
    cout<<++a;
    a++;
    cout<<a;
}
```

a	AB
b	AC
c	BC
d	BD
e	CD

Câu hỏi 3: Kết quả đưa ra màn hình

```
#include <iostream>
using namespace std;
void fun(char &);
int main() {
    char a[10]="ABCDEF";
    fun(a[0]);
    return 0;
}
void fun(char &a){
    cout<<++a;
    a++;
    cout<<a;
}
```

a	AB
b	AC
c	BC
d	BD
e	CD



Bài tập 1

Cho hàm $f(x)$ được định nghĩa như sau

$$f(x) = \begin{cases} \sqrt{e^{2x+1} + 1} + 7 & \text{khi } |x| \leq 2 \\ x^5 + 5x^3 + x + 1 & \text{khi } |x| > 2 \end{cases}$$

Hãy viết chương trình thực hiện các công việc sau

- Viết chương trình con tính hàm trên
- Nhập vào từ bàn phím 2 số thực x, y , tính và đưa ra màn hình $(f(x)+f(y))^2$
- Đưa ra màn hình theo dòng các cặp $\langle x, f(x) \rangle$ trong đó giá trị của x lần lượt là -5.0; -4.9; -4.8;.....2.8; 2.9; 3.0.



Bài tập 2

- Tiền điện được tính theo số điện tiêu thụ như sau
 - Dùng ít hơn 250 số: 2000đồng/số
 - Dùng từ 250 đến 400 số: 3000 đồng/số
 - Dùng từ 400 đến 500 số: 4000 đồng/số
 - Dùng từ 500 số trở lên: 5000đ/số
- Hãy viết hàm *TienDien*, trả về số tiền điện phải trả với tham số vào là số điện năng đã tiêu thụ
- Nhập vào dãy số điện sử dụng của n hộ gia đình ($n \leq 20$)
- Đưa ra màn hình số tiền điện ít nhất trong các hộ.