

Chương 1

Lập trình C++ cơ bản (phần 2)

Nội dung

- 1.4 Kiểu dữ liệu mảng
- 1.5 Kiểu dữ liệu cấu trúc
- 1.6 Hàm

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

1.4 Kiểu dữ liệu mảng

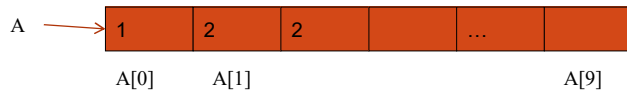
- **Mảng:** là dãy gồm nhiều **phần tử**
- Mỗi phần tử của mảng được định danh qua
- Cấu trúc của mảng xác định qua số chiều:

A) Mảng 1 chiều

- Khai báo gồm: kiểu dữ liệu của phần tử, tên mảng và số lượng phần tử
- Chú ý: số lượng phần tử phải là hằng hoặc biểu thức hằng số nguyên dương

Thao tác với phần tử mảng

- Truy xuất đến phần tử qua chỉ số và dấu []
- Chú ý: chỉ số mảng bắt đầu từ 0
- Chỉ số phải là số nguyên



•5

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

B) Mảng 2 chiều

- Khai báo gồm: kiểu dữ liệu, tên mảng, kích thước mảng theo từng chiều
- Mảng 2 chiều được sử dụng để lưu dữ liệu

•6

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

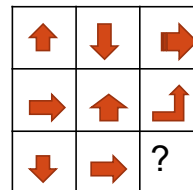
Ví dụ

1 2 4
5 6 8
2 3 4

Ma trận

	Thắng	Hòa	Thua	Điểm
AC Milan	1	1	1	4
MU	3	0	0	9
Ajax	1	1	1	4
Auxerre	0	0	3	0

Bảng



Lưới 2 chiều

•7

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

Cách nhập và in toàn bộ phần tử

- Sử dụng 2 cấu trúc for lồng nhau
- Nhập dữ liệu phần tử
- In giá trị phần tử

•8

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

Mảng ký tự

- Mảng ký tự là mảng 1 chiều với phần tử là ký tự
char ten[100] = "Tran Van Tuan";
char lop[40] = "63IT1";
- In mảng ký tự
- Nhập mảng ký tự

•9

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

Bài tập minh họa

Bài 1. Lập chương trình xử lý một dãy số nguyên n phần tử nhập từ bàn phím

- Tính và in ra tổng dãy số
- Tính và in ra trung bình của dãy số
- So sánh tổng các số có vị trí lẻ (1,3,5,...) và tổng các số có vị trí chẵn (2,4,6,...)
- Tìm và in ra vị trí xuất hiện đầu tiên trong dãy của phần tử x nhập từ bàn phím
- Tìm và in ra số lớn nhất của dãy
- Nhập số nguyên x và chỉ số $0 < i < n$. Chèn số x vào vị trí i trong dãy
- Nhập số nguyên i: $0 \leq i < n$. Xóa phần tử có chỉ số i khỏi dãy
- Sắp xếp dãy số theo thứ tự tăng và in ra màn hình

•10

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

- **Bài 2.** Lập chương trình xử lý một ma trận số nguyên m x n nhập từ bàn phím
 - Tính và in ra tổng các phần tử của ma trận
 - Tính và in ra tổng các phần tử chẵn và tổng các phần tử lẻ
 - Tìm và in ra số lớn nhất trên từng hàng của ma trận
 - Tìm và in ra số nhỏ nhất trên từng cột của ma trận
 - Tìm và in ra giá trị, chỉ số hàng và cột của phần tử nhỏ nhất

•11

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

1.5) Kiểu dữ liệu cấu trúc

- Kiểu struct: biểu diễn dữ liệu
- Kiểu struct dùng để mô tả

Ví dụ: mỗi 1 hồ sơ sinh viên gồm các thành phần:

•12

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

- Không dùng struct, vẫn có thể biểu diễn được dữ liệu nhưng sẽ rời rạc và không có tính cấu trúc
- Struct giúp

VD: Ma trận thường được biểu diễn với các dữ liệu sau

```
unsigned int n,m; // số hàng và cột
int A[10][10]; // các phần tử
```

•13

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

Khai báo (tạo) kiểu struct

- Sử dụng từ khóa struct, tên kiểu tự chọn và khai báo các thành phần dữ liệu
- Khai báo kiểu struct MaTran

•14

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

Khai báo biến dữ liệu struct

- Sử dụng tên kiểu struct và tên biến
- Có thể kết hợp vừa khai báo kiểu và khai báo biến struct

```
struct MaTran
{
    int hang; // số hàng
    int cot; // số cột
    int E[10][10]; // mảng chứa phần tử
}
```

•15

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

Thao tác với biến dữ liệu struct

- Nhập dữ liệu: sử dụng cú pháp **<biến struct>.<tên thành phần>**
- Tính toán và in ra màn hình cũng dùng cú pháp trên

•16

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

Mảng với phần tử là struct

- Khai báo với tên kiểu struct, tên mảng và số phần tử
- Truy nhập phần tử mảng và thành phần

Số hàng của ma trận chỉ số 0:

Số cột của ma trận chỉ số 3:

Phần tử i,j của ma trận chỉ số i :

•17

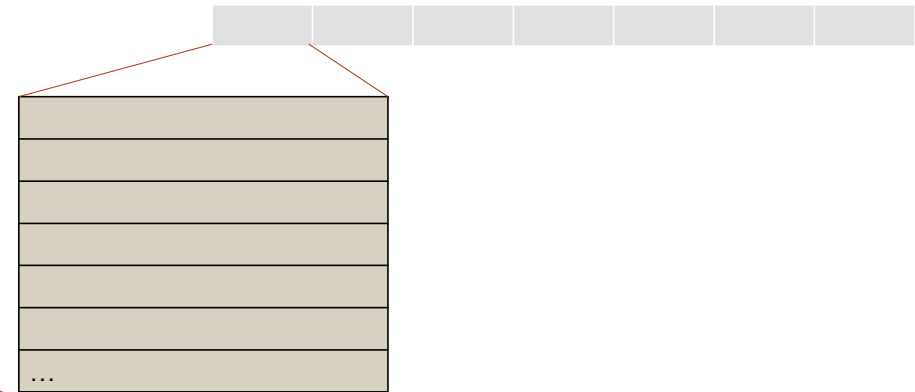
•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

Ví dụ minh họa

```
struct MaTran dayMaTran[6];
```

dayMaTran[5]



•18

Định nghĩa tên kiểu dữ liệu với từ khóa typedef

- Người lập trình có thể đặt tên mới cho một kiểu dữ liệu tùy ý
- Cú pháp

```
typedef <tên kiểu đã có> <tên kiểu mới>;
```

```
struct MaTran
{
    unsigned int n;
    unsigned int m;
    int E[10][10];
};
// khai báo với tên kiểu mới
```

•19

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

Bài tập minh họa

- Bài 1. Lập chương trình quản lý danh bạ thông tin đối tác của 1 công ty với mỗi mục bao gồm: tên đối tác, cơ quan công tác, chức vụ, số điện thoại di động, số điện thoại cơ quan, địa chỉ nhà riêng, địa chỉ cơ quan, địa chỉ email, ghi chú.
 - Nhập dữ liệu cho danh bạ (trong danh bạ lưu được tối đa 500 đối tác)
 - In danh bạ ra màn hình ở dạng bảng với các cột là các trường thông tin như STT, tên, cơ quan, chức vụ, số di động, email
 - Tìm kiếm đối tác trong danh bạ theo tên và cơ quan. Nếu tìm thấy thì in ra màn hình các trường thông tin như trong câu b)
 - Sắp xếp danh bạ theo tên với thứ tự ABC và in ra màn hình với các trường thông tin như trong câu b)

•20

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

1.6) Hàm

- Hàm là gì ?
- Hàm có nghĩa là
- Hàm là một đơn vị (mô đun) của chương trình,
- Một số hàm đã biết như các hàm toán học: sqrt, log ...

•21

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

1.6.2 Cách tạo HÀM C++

- Khai báo **nguyên mẫu hàm** (function prototype) , còn gọi là kiểu của hàm, bao gồm
- Lập trình các câu lệnh bên trong hàm (function definition)

•22

A) Khai báo nguyên mẫu

- Cú pháp:

```
kiểu_dl_trả_về    tên_hàm(danh_sách_tham_số) ;
```

Ví dụ: nguyên mẫu hàm giá trị max của 2 số nguyên x, y

Cách khai báo 1:

Cách khai báo 2:

Nguyên mẫu của hàm tính căn bậc 2:

•23

Tiếp

- Chú ý:
 - Danh sách tham số có thể trống (không tham số)
 - Kiểu trả về của hàm có thể là **void** (hàm không trả lại giá trị nào)

•24

B) Lập trình xử lý bên trong hàm (định nghĩa hàm)

- Cú pháp

```
<kiểu_dl> <tên_hàm>(<ds_tham_số_hình_thức>)  
{  
    // phần thân hàm (function body)  
    Khai_báo_biến_cục_bộ;  
    Dãy_câu_lệnh_xử_lý;  
}
```

- Biến cục bộ là

- Dãy câu lệnh thực hiện thao tác xử lý, tính toán của hàm

•25

Chương trình minh họa

- Hàm tính tổng bình phương của 2 số thực

Danh sách
tham số

Trả về kết quả

•26

Tiếp

- **Chú ý:** lệnh `return`
 - Khi thực hiện lệnh `return`,
- Nếu kiểu của hàm là `void`,

•27

Chương trình ví dụ

- Lập hàm tính giá trị lớn nhất của một dãy số nguyên đầu vào `A[]`
 - Xử lý: Duyệt dãy và tìm giá trị lớn nhất
 - Kiểu trả về: `int`
 - Tham số đầu vào
 - Mảng `A` phần tử nguyên
 - Số phần tử `n`
 - Giá trị trả về: phần tử lớn nhất trong mảng `A`

•28

1.6.3 Lời gọi hàm

- Sau khi tạo hàm, để sử dụng cần phải **gọi hàm (call function)**
- Lời gọi hàm (function call) bao gồm

- Ví dụ

```
int x, y, s;  
x = y = 5; // x và y là 2 tham số truyền cho hàm
```

•29

Tên hàm

Tham số thực sự

Tham số hình thức và tham số thực sự

Tham số hình thức

Tham số thực sự

```
// a và b là tham số hình thức  
float max(float a, float b)  
{  
    return a > b ? a : b ;  
}
```

```
// x và 5.5 là tham số thực  
sự  
int m, x = 7.7;  
m = max(x, 5.5);
```

•30

Chú ý

- Số lượng tham số thực sự = số lượng tham số hình thức
- Kiểu dữ liệu của tham số thực sự phải

•31

B) Truyền tham số

- Truyền tham số là truyền tham số thực sự vào bên trong phần thân hàm thông qua tham số hình thức

```
float sum_of_square(float a, float b)  
{  
    float sum; // biến cục bộ  
    tong = a*a + b*b;  
    return sum;  
}
```

•32

Các kiểu truyền tham số

Truyền giá trị

- Tham số hình thức là bản sao giá trị của tham số thực sự
- Kết quả:

Truyền tham chiếu

- Tham số hình thức là một định danh khác (bí danh) của chính tham số thực sự
- Kết quả:

•33

C) Tham số kiểu mảng

- Có thể đặt mảng 1 chiều, mảng 2 chiều làm tham số hình thức của hàm

```
void max(int A[100], int n)
{
    int max = A[0];
    int i;
    for(i=1; i<n; i++)
        if(max<A[i]) max = A[i];
    return max;
}
```

•34

Chú ý

- Tham số thực sự là

•35

Chương trình minh họa

- Hàm `void sort(int A[], int n)` thực hiện việc sắp xếp một dãy số (mảng 1 chiều) theo thứ tự tăng dần.
- Tham số:
 - A là mảng 1 chiều cần sắp xếp
 - n là số phần tử trong mảng
- Cũng có thể truyền tham số dạng con trỏ như sau
 - `void sort(int* A, int n)`
 - A là con trỏ, trỏ đến mảng 1 chiều cần sắp xếp

•36

1.6.4 Phạm vi sử dụng của biến

- **Phạm vi** (scope) của một biến dữ liệu là
- Tùy theo phạm vi, một biến sẽ có **thời gian tồn tại và vị trí bộ nhớ** khác nhau
- **Thời gian tồn tại** là từ thời điểm biến được cấp bộ nhớ cho đến khi giải phóng khỏi bộ nhớ chương trình
- **Vị trí bộ nhớ:**

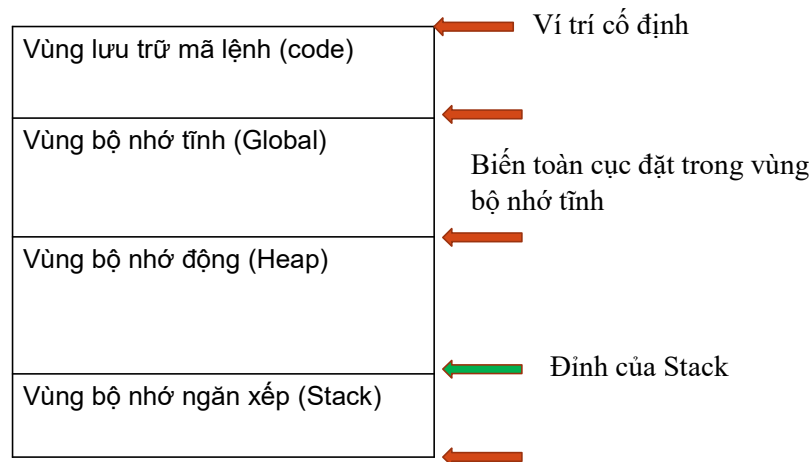
•37

Biến cục bộ và toàn cục

- Biến cục bộ là biến khai báo trong thân một hàm
- Biến toàn cục là biến khai báo bên ngoài, không thuộc bất cứ hàm nào

•38

Các vùng trong bộ nhớ của 1 chương trình máy tính



Biến cục bộ nằm trong vùng bộ nhớ Stack

•39

Biến cục bộ

- Nếu khai báo biến cục bộ nằm trong một khối lệnh thì
- Khi thực thi xong khối lệnh thì

```
int i; // biến i ngoài khối
for(i=0; i<10; i++)
{ // khối lệnh
    int x; // biến x nằm trong khối
    x = i*2;
    cout << x;
}
x = 3; // báo lỗi vì ra khỏi phạm vi, biến x không còn sử dụng
```

•40

Ví dụ biến toàn cục và biến cục bộ

```
#include <iostream>
using namespace std;
int r ;
float sum_of_square(float a, float b)
{
    float sum;
    sum = a*a + b*b;
    return sum;
}
int main()
{
    float x = 5.5, y = 6.6;
    r = sum_of_square (x, y);
    cout << "r = " << setw(6) << setprecision(2)<< r;
    return 0;
}
```

•41

Bài tập minh họa

- **Bài 1** Viết hàm tính giá trị của hàm e^x với x là tham số, theo công thức chuỗi sau

$$e^x = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} \text{ với điều kiện dừng là } \frac{x^n}{n!} \leq 10^{-8}$$

- Viết hàm tính giá trị của hàm $\sin(x)$, theo công thức

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} \text{ với đk dừng là } \left| \frac{x^{2n+1}}{(2n+1)!} \right| \leq 10^{-8}$$

•42

•B/m KTHT, khoa CNTT, ĐHXD

•Bài giảng C++. 8/10/2019

Bài 3

Lập chương trình thao tác trên ma trận, gồm các hàm sau

a) Hàm nhập một ma trận số nguyên từ bàn phím với tham số hình thức là 1 ma trận 2 chiều, số hàng và số cột của ma trận. Dữ liệu sẽ được nhập vào các phần tử của tham số ma trận

void nhap_matran(int A[][30], int m, int n);

b) Viết hàm chuyển vị một ma trận

void chuyenvi_matran(int A[][30], int m, int n);

c) Viết hàm để chèn vector V có n phần tử vào vị trí hàng p của một ma trận đầu vào A.

void chen_vecto_matran(int A[][30], int V[], int m, int n, int p);

d)Viết một hàm main() thực hiện các công việc sau (có sử dụng các hàm đã viết ở trên)

- Nhập m, n
- Nhập ma trận kích thước m x n (m hàng n cột) từ bàn phím vào mảng A
- Chuyển vị ma trận A và in ma trận ra màn hình
- Chèn thêm một hàng gồm toàn số 1 vào hàng 4 trong ma trận và in ma trận ra màn hình

Chú ý: Khai báo A[MAX][MAX] với hằng số MAX = 30. Kích thước ma trận nhập vào từ bàn phím: m, n < MAX

•43

B/m KTHT, khoa CNTT, ĐHXD

Bài giảng C++. 8/10/2019