

# CHAPTER 6. STRUCTURES

# KHAI BÁO CẤU TRÚC - KHAI BÁO BIẾN CẤU TRÚC

*Định nghĩa một kiểu dữ liệu có cấu trúc mới*

```
struct tên_kiểu_cấu_trúc
{ //khai báo các thành phần của cấu trúc
    <kiểu dữ liệu> <tên thuộc tính 1>
    <kiểu dữ liệu> <tên thuộc tính 2>
    <kiểu dữ liệu> <tên thuộc tính 3>
    ...
};
```

Chẳng hạn cấu trúc của bảng lương có thể được mô tả như sau:

Thành phần của cấu trúc có thể có kiểu dữ liệu là một cấu trúc đã được định nghĩa trước đó:

```
struct ngaythang
{
    int ngay;
    int thang;
    int nam;
};
```

```
struct hoso
{
    char hoten [30];
    struct ngaythang ns;
    long luongcoban;
    long thuong;
    long thucclanh;
};
```

## Khai báo biến cấu trúc

<kiểu cấu trúc> <tên biến cấu trúc>

Chẳng hạn:

```
hoso nv[1000];
```

```
hoso *nv;
```

**Lưu ý:** Sử dụng lệnh `gets` để nhập chuỗi; trước mỗi lệnh nhập chuỗi cần sử dụng lệnh xóa vùng đệm bàn phím bằng lệnh *`flushall()`* hoặc *`fflush(stdin)`*.

## Truy xuất đến từng thành phần của cấu trúc

### *Biến cấu trúc được khai báo tĩnh*

Với cấu trúc đơn thì theo cú pháp sau:

tên biến cấu trúc.tên thành phần

Ví dụ:

nv[i].hoten;      // chỉ *họ tên* của nhân viên thứ i

nv[i].thuong;    // chỉ *thưởng* của nhân viên thứ i

Với cấu trúc phức thì theo cú pháp sau:

tên biến cấu trúc.tên biến cấu trúc.tên thành phần

Ví dụ:

nv[i]. ns.ngay;

nv[i]. ns.thang;

nv[i]. ns.nam;

***Biến cấu trúc được khai báo động***

tên biến cấu trúc -> tên thành phần.

## Ví dụ 1.

*Viết chương trình cộng hai phân số  $a/b$  và  $c/d$  trong đó  $a, b, c, d$  là các số nguyên. Yêu cầu phân số kết quả phải ở dạng tối giản.*

```
1. #include <iostream.h>
2. struct phanso
3. {
4.     int tu;
5.     int mau;
6. };
7. void      nhapps(phanso &ps);
8. void      xuatps(phanso ps);
9. phanso congps( phanso ps1, phanso ps2);
10.int uscln(int a, int b);
```



```
11.int main()
12.{
13. phanso ps1,ps2;
14. nhapps(ps1);
15. nhapps(ps2);
16. xuatps(congps(ps1,ps2));
17.}

18.void nhapps(phanso &ps)
19.{
20.cout<<"Nhap vao tu so : ";cin>>ps.tu;
21.cout<<"Nhap vao mau so : ";cin>>ps.mau;
22.}
```

```
23. void xuatps (phanso ps)
24. {
25.     int uc=uscln(ps.tu,ps.mau);
26.     ps.tu=ps.tu/uc;
27.     ps.mau=ps.mau/uc;
28.     cout<<"Ketqua: "<<ps.tu<<"/"<<ps.mau<<endl;
29. }
```

```
30.phanso congps( phanso ps1, phanso ps2)
31.{
32.  phanso ps;
33.  ps.tu=ps1.tu*ps2.mau+ps2.tu*ps1.mau;
34.  ps.mau=ps1.mau*ps2.mau;
35.  return ps;
36.}
```

```
37.int uscln(int a, int b)
38.{
39.  int r=a%b;
40.  while (r!=0)
41.  {
42.      a=b;
43.      b=r;
44.      r=a%b;
45.  }
46.  return b;
47.}
```

## Ví dụ 2.

Viết chương trình thực hiện các công việc sau:

- a.** Nhập vào hồ sơ của  $n$  nhân viên với các thông tin: họ và tên, ngày sinh, lương cơ bản, thưởng, thực lãnh; trong đó thực lãnh = lương cơ bản + thưởng.
- b.** In danh sách nhân viên theo bậc lương giảm dần.

```
1. #include<iostream.h>
2. #include<stdlib.h>
3. #include<iomanip.h>
4. struct ngaysinh
5. {
6.     int ngay;
7.     int thang;
8.     int nam;
9. };

```

```
10.struct hoso  
11.{ char hoten [30];  
12.  struct ngaysinh ns;  
13.  long luongcoban;  
14.  long thuong;  
15.  long thucclanh;  
16.};
```

```
17. void nhap(hoso nv[], int &n);
18. void xuat(hoso nv[], int n);
19. void sapxep(hoso nv[], int n);
20. int main()
21. {
22.     int n;
23.     hoso nv[1000];
24.     nhap(nv,n);
25.     sapxep(nv,n);
26.     xuat(nv,n);
27. }
```



```

28. void nhap(hoso nv[], int &n)
29. {
30.     cout<<"Ho so co bao nhieu nhan vien:";cin>>n;
31.     for(int i=0;i<n;i++) //bắt đầu từ 1 để dễ theo dõi
32.     {
33.         cout<<"\Nhap nhan vien thu: "<<i;
34.         cout<<"\nHo                               sten:";
           flushall();gets(nv[i].hoten);
35.         cout<<"Ngay:";cin>>nv[i]. ns.ngay;
36.         cout<<"Thang:";cin>>nv[i]. ns.thang;
37.         cout<<"Nam:";cin>>nv[i]. ns.nam;
38.         cout<<"Luong co ban:";cin>> nv[i].luongcoban;
39.         cout<<"Thuong:") ;cin>>nv[i].thuong;
40.         nv[i].thucclanh          =          nv[i].luongcoban+
           nv[i].thuong;
41.     }
42. }

```

```

43. void xuat(hoso nv[], int n)
44. {
45.     cout<<"      Ho   ten      Ngay   sinh      LCB      Thuong      Thuc
        lanh"<<endl;
46.         for(int i=0;i<n;i++)
47.             cout<<nv[i].hoten<<setw(5)<<nv[i].
                ns.ngay<<setw(5)<<nv[i].      ns.thang<<setw(5)<<nv[i].      ns.nam
                <<setw(5)<<nv[i].luongcoban<<setw(5)<<nv[i].thuong<<setw(5)<
                <nv[i].thuc lanh;
48.     }

```

```
49. void sapxep(hoso nv[], int n)
50. {
51.     for(int i=0;i<n-1;i++)
52.         for(int j=i+1;j<n;j++)
53.             if(nv[i].thuclanh > nv[j].thuclanh)
54.             {
55.                 hoso temp = nv[i];
56.                 nv[i] = nv[j];
57.                 nv[j] = temp;
58.             }
59. }
```

### BT1.

Nhập vào dãy  $n$  phân số; trong đó tử số và mẫu số là các số nguyên dương.

- a. Đếm xem có bao nhiêu phân số có giá trị lớn hơn 0 và nhỏ hơn 1 ?
  - b. Tìm phân số có giá trị lớn nhất của mảng.
  - c. Hãy tính tổng các phân số của dãy; kết quả là một phân số.
- Yêu cầu phân số kết quả của 2 câu trên ở dạng tối giản.

Ví dụ:

$n=4$

$2/5, 3/7, 16/21, 17/6$  thì kết quả là  $929/210$

## BT2.

Cho mảng hai chiều  $m$  dòng  $n$  cột; mỗi phần tử là một phân số; các phân số có tử số và mẫu số là các số nguyên dương.

- Tìm phân số có giá trị lớn nhất của mảng.
- Tính tổng các phân số của mảng; kết quả là phân số ở dạng tối giản.

Ví dụ:

2 3

$\frac{1}{2}$   $\frac{3}{2}$   $\frac{4}{7}$

$\frac{3}{7}$   $\frac{3}{4}$   $\frac{5}{3}$

Thì kết quả câu a là  $\frac{5}{3}$ ; kết quả câu b là  $\frac{65}{12}$

### BT3.

Trong kỳ thi tốt nghiệp đại học có  $n$  thí sinh dự thi tham dự. Mỗi thí sinh cần quản lý các thông tin được mô tả như sau:

- Mã số sinh viên
- Họ và tên
- Điểm môn cơ bản
- Điểm môn chuyên ngành 1
- Điểm môn chuyên ngành 2

- Hãy khai báo kiểu dữ liệu cho bài toán như mô tả trên. Nhập thông tin cho  $n$  thí sinh.
- Xuất danh sách với các thông tin gồm 5 thông tin trên; điểm tổng 3 môn.
- Tìm những thí sinh có ít nhất một môn thi có điểm nhỏ hơn 5.
- Tìm những thí sinh có tổng điểm ba môn thi lớn nhất.

#### BT4.

Cho dãy  $n$  khối lập phương. Trên mỗi mặt của mỗi khối lập phương ghi một giá trị nguyên từ 1 đến 9.

- Tổng giá trị của mỗi khối lập phương bằng tổng các giá trị trên các mặt của khối lập phương đó. Hãy tìm tổng lớn nhất.
- Hãy cho biết có bao nhiêu khối lập phương có giá trị của các mặt là các số khác nhau.
- Hãy cho biết giá trị nào xuất hiện nhiều lần nhất. Bao nhiêu lần ?

### BT5.

Cho  $n$  hình chữ nhật có các cạnh song song với các trục tọa độ, mỗi hình chữ nhật biết tọa độ dưới trái, trên phải là  $x_1[i], y_1[i], x_2[i], y_2[i]$  ( $i=1..n$ ). Hãy tính diện tích của hình được phủ bởi  $N$  hình trên.



Hết