

CHƯƠNG 7

Cấu trúc

Cấu Trúc

- Cấu trúc **(struct)** là sự kết hợp của một hay nhiều biến cùng hoặc khác kiểu, được đặt dưới một tên chung để thuận tiện xử lý.
- Cấu trúc struct giúp tổ chức các kiểu dữ liệu phức tạp, nó cho phép quan điểm các biến có liên quan với nhau như một thực thể thay vì là các thực thể riêng rẽ.

Cấu Trúc

- Cấu trúc là phương tiện để trừu tượng hoá dữ liệu, là cơ sở cho lập trình cấu trúc.
- Cấu trúc thường được dùng để biểu diễn các khái niệm dữ liệu có trong thực tế như phân số, sinh viên, thời điểm...
- Các thành phần của cấu trúc được gọi là các trường hay vùng tin (fields).

Khai báo cấu Trúc

- Ta dùng từ khoá struct và các dấu {, } để khai báo kiểu cấu trúc:

```
struct PhanSo {  
    int tu, mau;  
};  
struct QuocGia {  
    char TenNuoc[MAX];  
    char ThuDo[MAX];  
    int DanSo, DienTich;  
};
```

Khai báo, định nghĩa biến cấu trúc

- Tương tự như khai báo, định nghĩa biến thuộc kiểu cơ bản.

PhanSo a, b, c;

QuocGia vn;

Truy xuất thành phần của struct

- Để truy xuất thành phần của biến kiểu cấu trúc, dùng phép toán `'.'` hoặc phép toán `'->'`

`ten_bien.ten_truong`

`ten_biencontro->ten_truong`

- Mọi thao tác trên cấu trúc được thực hiện thông qua các *trường* của biến.

Khởi động cấu trúc

- Ta khởi động cấu trúc bằng dấu { và } tương tự như khởi động mảng.

```
struct PhanSo
```

```
{
```

```
    int tu, mau;
```

```
};
```

```
PhanSo a = { 1,4 },
```

```
        b = { 1,2 }, c;
```

Khởi động cấu trúc

```
struct Diem  
{  
    double x, y;  
};
```

```
Diem A = { 2.1, 4.5 },  
        B = { 4.1, 8.3};
```

```
struct NTN  
{  
    int ngay, thang, nam;  
};
```

```
NTN NgaySinh = { 12, 2, 1992 },  
NgayCuoi = { 2, 2, 2042 };
```


Khởi động cấu trúc

```
struct SinhVien
{
    char MaSo[20];
    char HoTen[32];
    NTN NamSinh;
};
```

```
SinhVien s = { "07112138", "Nguyen Van Ich", { 15,6,2000 }
};
```

```
struct TamGiac
{
    Diem A, B, C;
} t = { {200, 200}, {400,200}, {200,400} },
u = { 100, 200, 300, 150, 250, 400 };
```

Truyền tham số cấu trúc và trả về giá trị

- Truyền tham số cấu trúc cho hàm được thực hiện tương tự như kiểu cơ bản.
- Ta cũng có thể trả về kết quả kiểu cấu trúc.

```
#include <stdio.h>
#include <stdlib.h>

struct PhanSo
{
    int tu, mau;
};

int uscln(int a, int b)
{
    if (b == 0) return abs(a);
    return uscln(b, a%b);
}
```

```
void PS_RutGon(PhanSo *a)
{
    int u = uscln(a->tu, a->mau);
    a->tu /= u;
    a->mau /= u;
    if (a->mau < 0)
    {
        a->tu = -a->tu;
        a->mau = -a->mau;
    }
}
```

Tham số con trỏ

```
void PS_RutGon(PhanSo *a)
{
    int u = uscln(a->tu, a->mau);
    a->tu /= u;
    a->mau /= u;
    if (a->mau < 0)
    {
        a->tu = -a->tu;
        a->mau = -a->mau;
    }
}
```

Tham số con trỏ

Trả về cấu trúc

Tham số cấu trúc

```
PhanSo PS_Cong(PhanSo a, PhanSo b)
{
    PhanSo c;
    c.tu = a.tu*b.mau + a.mau*b.tu;
    c.mau = a.mau*b.mau;
    PS_RutGon(&c);
    return c;
}
```

```
PhanSo PS_Tru(PhanSo a, PhanSo b)
{
    PhanSo c;
    c.tu = a.tu*b.mau - a.mau*b.tu;
    c.mau = a.mau*b.mau;
    PS_RutGon(&c);
    return c;
}
```

```
PhanSo PS_Nhan(PhanSo a, PhanSo b)
{
    PhanSo c;
    c.tu = a.tu*b.tu;
    c.mau = a.mau*b.mau;
    PS_RutGon(&c);
    return c;
}
```

Tham số cấu trúc

```
PhanSo PS_Chia(PhanSo a, PhanSo b)
{
    PhanSo c;
    c.tu = a.tu*b.mau;
    c.mau = a.mau*b.tu;
    PS_RutGon(&c);
    return c;
}
```

```
void main()
{
    PhanSo a = { 1,4 }, b = { 1,2 }, c;
    c = PS_Cong(a, b);
    PS_Xuat(c);
    puts("\n");
    c = PS_Tru(a, b);
    PS_Xuat(c);
    puts("\n");
    c = PS_Nhan(a, b);
    PS_Xuat(c);
    puts("\n");
    c = PS_Chia(a, b);
    PS_Xuat(c);
    puts("\n");
}
```

Khởi động cấu trúc

Truyền tham số cấu trúc

Mảng cấu Trúc

- Kiểu cấu trúc (**struct**) cũng là kiểu dữ liệu, vì vậy, ta có thể khai báo, định nghĩa, tạo và khởi động một mảng các phần tử thuộc kiểu cấu trúc.

Ví dụ:

1. Viết một ứng dụng cho phép nhập vào tên nước, xuất ra tên thủ đô tương ứng.
2. Viết một ứng dụng xuất ra một danh sách các quốc gia với các thông tin đi kèm: Tên quốc gia, tên thủ đô, dân số, diện tích.

Tương ứng tên nước, thủ
đô dùng mảng 3 chiều

```
#include <stdio.h>
#include <string.h>
• #include <stdio.h>
• #include <string.h>
```

```
using namespace std;
const int MAX = 100;
```

```
int Tim(char a[][2][MAX], int n, char *s)
{
    for (int i = 0; i < n; i++)
        if (_stricmp(a[i][0], s) == 0)
            return i;
    return -1;
}
const int MAX = 100;
```

```

void main()
{
    static char a_bodoi[][2][MAX] =
    {
        "Viet Nam", "Ha Noi",
        "Phap", "Paris",
        "Trung Quoc", "Bac Kinh",
        "Hoa Ky", "Washington",
        "Anh", "Luan don"
    };
    int N = sizeof(a_bodoi) / sizeof(a_bodoi[0]);
    char nuoc[MAX];
    printf("Nhap nuoc: ");
    gets_s(nuoc, MAX);
    int j = Tim(a_bodoi, N, nuoc);
    if (j >= 0)
        printf("%s\t%s\n", a_bodoi[j][0], a_bodoi[j][1]);
    else
        printf("Khong tim thay\n");
}

```

Khởi động mảng 3 chiều

```
#include <stdio.h>
#include <string.h>
```

```
const int MAX = 100;
```

```
struct QuocGia {
    char TenNuoc[MAX];
    char ThuDo[MAX];
};
```

```
int Tim(QuocGia a[], int n, char *s)
{
    for (int i = 0; i < n; i++)
        if (_stricmp(a[i].TenNuoc, s) == 0)
            return i;
    return -1;
}
```

Tương ứng tên nước,
thủ đô dùng cấu trúc

```

void main()
{
    static QuocGia aQ[] =
    {
        {"Viet Nam", "Ha Noi"},
        {"Phap", "Paris"},
        {"Trung Quoc", "Bac Kinh"},
        {"Hoa Ky", "Washington"},
        {"Anh", "Luan don"}
    };
    int N = sizeof(aQ) / sizeof(aQ[0]);
    char nuoc[MAX];
    printf("Nhap nuoc: ");
    gets_s(nuoc, MAX);
    int j = Tim(aQ, N, nuoc);
    if (j >= 0)
        printf("%s\t%s\n", aQ[j].TenNuoc, aQ[j].ThuDo);
    else
        printf("Khong tim thay\n");
}

```

Khởi động cấu trúc

```
const int MAX = 100;
struct QuocGia {
    char TenNuoc[MAX];
    char ThuDo[MAX];
    int DanSo, DienTich;
};
```

```
int Tim(QuocGia a[], int n, char *s)
{
    for (int i = 0; i < n; i++)
        if (_stricmp(a[i].TenNuoc, s) == 0)
            return i;
    return -1;
}
```

```
void XuatQuocGia(QuocGia q)
{
    printf("Quoc Gia: %s\nThu Do: %s\nDan So: %d nguoi\nDien  
Tich: %d km2", q.TenNuoc, q.ThuDo, q.DanSo, q.DienTich);
}
```

Tương ứng tên nước với
thủ đô, dân số, diện tích.

```
void main()
```

```
{
```

```
    static QuocGia aQ[] =
```

```
    {
```

```
        { "Viet Nam", "Ha Noi", 95261021, 331212 },
```

```
        { "Phap", "Paris", 66259012, 632834 },
```

```
        { "Trung Quoc", "Bac Kinh", 1373541278, 9572900 },
```

```
        { "Hoa Ky", "Washington", 323995528, 9526468 },
```

```
        { "Anh", "Luan don", 63742977, 242900 }
```

```
    };
```

```
    int N = sizeof(aQ) / sizeof(aQ[0]);
```

```
    char nuoc[MAX];
```

```
    printf("Nhap nuoc: ");
```

```
    gets_s(nuoc, MAX);
```

```
    int j = Tim(aQ, N, nuoc);
```

```
    if (j >= 0)
```

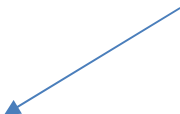
```
        XuatQuocGia(aQ[j]);
```

```
    else
```

```
        printf("Khong tim thay\n");
```

```
}
```

Khởi động mảng cấu trúc



Q&A