

3.4. Mảng cấu trúc

Người lập trình có thể sử dụng các kiểu cấu trúc đã được định nghĩa để khai báo các cấu trúc và mảng các cấu trúc.

Ví dụ chương trình sau đây cho phép nhập vào một mảng các linh kiện:

```
#include<conio.h>
```

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

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

```
typedef struct
```

```
{
```

```
    unsigned char Ngay;
```

```
    unsigned char Thang;
```

```
    unsigned int Nam;
```

```
} NgaySX;
```

```
typedef struct
```

```
{
```

```
    char ma_lk[10];
```

```
    char ten_lk[40];
```

```
    NgaySX Ngaysx;
```

```
    char DiaChi_NSX[40];
```

```
} LinhKien;
```

```
// In danh sach linh kien ra man hinh
```

```
void InLK(LinhKien LK[100],int n)
```

```

{
    printf("Ma linh kien: |Ten linh kien| Ngay san xuat |Dia chi NSX\n");
    for(int i=0;i<n;i++)
        printf("%s      | %s      | %d-%d-%d      |%s\n",LK[i].ma_lk,LK[i].ten_lk,
            LK[i].Ngaysx.Ngay,LK[i].Ngaysx.Thang,LK[i].Ngaysx.Nam,LK[i].DiaChi_NSX);
}

```

// Nhap vao danh sach linh kien

```

void NhapLK(LinhKien LK[100], int n){
    for (int i=0;i<n;i++){
        printf("\nNhap ma linh kien: "); fflush(stdin);
        gets(LK[i].ma_lk);
        printf("\nNhap ten linh kien: ");gets(LK[i].ten_lk);
        printf("\nNgay san xuat: ");scanf("%d",&LK[i].Ngaysx.Ngay);
        printf("\nThang san xuat: ");scanf("%d",&LK[i].Ngaysx.Thang);
        printf("\nNam san xuat: ");scanf("%d",&LK[i].Ngaysx.Nam);
        fflush(stdin);
        printf("\nDia chi NSX: ");gets(LK[i].DiaChi_NSX);
    }
}

```

int **main**()

```

{

```

```

    LinhKien LK[100]; int n;

    clrscr();

    printf("Nhap vao so phan tu mang: "); scanf("%d",&n);

    NhapLK(LK,n);

    InLK(LK,n);

    getch();

    return 0;

}

```

3.5. Phép gán cấu trúc

Người lập trình có thể gán hai cấu trúc có cùng kiểu cho nhau. Ví dụ như chương trình sau:

```

#include<conio.h>

#include<stdio.h>

#include<string.h>

typedef struct

{

    unsigned char Ngay;

    unsigned char Thang;

    unsigned int Nam;

} NgaySX;

typedef struct

{

```

```

    char ma_lk[10];

    char ten_lk[40];

    NgaySX Ngaysx;

    char DiaChi_NSX[40];
} LinhKien;

void InLK(LinhKien s)
{
    printf("Ma linh kien: |Ten linh kien| Ngay san xuat |Dia chi NSX\n");
    printf("%s      | %s      | %d-%d-%d      |%s\n",s.ma_lk,s.ten_lk,
        s.Ngaysx.Ngay,s.Ngaysx.Thang,s.Ngaysx.Nam,s.DiaChi_NSX);
}

int main()
{
    LinhKien LK, LKMoi; //LK va LKMoi co cung kieu

    clrscr();

    printf("Nhap ma linh kien: ");

    gets(LK.ma_lk);

    printf("Nhap ten linh kien: ");gets(LK.ten_lk);

    printf("Ngay san xuat: ");scanf("%d",&LK.Ngaysx.Ngay);

    printf("Thang san xuat: ");scanf("%d",&LK.Ngaysx.Thang);

    printf("Nam san xuat: ");scanf("%d",&LK.Ngaysx.Nam);

```

```

fflush(stdin);

printf("Dia chi NSX: ");gets(LK.DiaChi_NSX);

InLK(LK);

LK Moi=LK; //Phep gan cau truc

printf("\n\nLinh kien moi duoc gan la:\n");

InLK(LK Moi); //In linh kien moi duoc gan

getch();

return 0;

}

```

3.6. Hàm trên các cấu trúc

a) Cấu trúc làm đối của hàm

Đối của hàm có thể là biến cấu trúc, hoặc mảng cấu trúc hình thức hoặc con trỏ cấu trúc.

- Khi đối số của hàm là một biến cấu trúc thì tham số thực của hàm tương ứng của hàm là một giá trị cấu trúc.
- Khi đối số của hàm là con trỏ cấu trúc thì tham số thực tương ứng của hàm là địa chỉ của các biến cấu trúc.
- Khi đối số của hàm là mảng cấu trúc hình thức hoặc con trỏ cấu trúc thì tham số thực tương ứng của hàm là tên mảng cấu trúc.

b) Hàm có kiểu cấu trúc

Các hàm có thể trả về giá trị là cấu trúc hoặc con trỏ cấu trúc

c) Ví dụ

Ví dụ sau đây sẽ minh họa việc sử dụng cấu trúc trong hàm:

Ví dụ 1: Xét kiểu cấu trúc DienTro gồm 3 thành phần :

+ Ten (Tên điện trở) kiểu mảng char

+ nsx (ngày sản xuất) kiểu struct date

+ giatri (giá trị)kiểu float

6 hàm sau đây được định nghĩa để thao tác trên kiểu DienTro

+ Hàm

```
DienTro *ptim(char *ten, DienTro ds[], int n);
```

có tác dụng tìm trong danh sách n điện trở lưu trong mảng ds điện trở có tên ten . Hàm trả về con trỏ tới điện trở tìm được hoặc trả về NULL nếu không tìm thấy .

+ Hàm

```
DienTro tim(char *Ten, DienTro ds[], int n);
```

Hàm này cũng có tác dụng tìm kiếm như hàm ptim, nhưng nó trả về một cấu trúc chứ không chứa thông tin điện trở tìm được . Các thông tin này sẽ bằng không nếu không tìm thấy .

+ Hàm

```
void DoiCho(DienTro *p1, DienTro *p2);
```

dùng để đổi chỗ hai cấu trúc .

+ Hàm

```
void sapxep(DienTro *p,int n);
```

có tác dụng sắp xếp n phần tử cấu trúc chứa trong p theo thứ tự tăng của ngày sản xuất. Trong hàm sapxep có dùng tới hàm DoiCho .

+ Hàm

```
void Nhap(DienTro *p);
```

dùng để nhập dữ liệu cho một đối tượng kiểu DienTro. Một chú ý quan trọng là : nếu trong hàm Nhap() ta dùng câu lệnh .

```
scanf(“%f%c”,&h.b1);
```

để nhập trực tiếp vào thành phần h.b1 thì bị treo máy.

+ Hàm

```
void in(DienTro p);
```

dùng để in thông tin 1 cấu trúc ra màn hình.

Thứ tự làm việc của chương trình như sau : Đầu tiên là phần nhập số liệu , rồi đến sắp xếp, sau đó sẽ in danh sách DienTro đã sắp xếp. Cuối cùng đến các mục tìm kiếm theo hàm ptim và hàm tim. Dưới đây là chương trình nguồn:

```
#include"stdio.h"
```

```
#include<conio.h>
```

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

```
struct date
```

```
{
```

```
    int ngay,thang,nam;
```

```
};
```

```
typedef struct
```

```
{
```

```
    char ten[25];
```

```
    struct date nsx;
```

```
    float giatri;
```

```

    } DienTro;

DienTro *ptim(char *ten,DienTro ds[],int n);

DienTro tim(char *ten,DienTro ds[],int n);

void DoiCho(DienTro *p1,DienTro *p2);

void SapXep(DienTro *ps,int n);

void Nhap(DienTro *p);

void in(DienTro p);

DienTro tim(char *ten,DienTro ds[],int n)
{
    int i;DienTro ps;

    ps.nsx.ngay=ps.nsx.thang=ps.nsx.nam=0; ps.giatri=0.0;

    ps.ten[0]=0;

    for(i=1;i<=n;i++)

        if (strcmp(ten,ds[i].ten)==0) return(ds[i]);

    return(ps);
}

DienTro *ptim(char *ten,DienTro ds[],int n)
{
    int i;

    for(i=1;i<=n;i++)

        if(strcmp(ten,ds[i].ten)==0) return(&ds[i]);

```



```

        return(NULL);

    }

void DoiCho(DienTro *p1,DienTro *p2)

{

    DienTro t;

    t=*p1;

    *p1=*p2;

    *p2=t;

}

void Nhap(DienTro *p)

{

    DienTro q; float gt;

    printf("\nNhap vao ten dien tro :");gets(q.ten);

    printf("\nNhap ngay ngay san xuat :");

    scanf("%d%d%d",&q.nsx.ngay,&q.nsx.thang,&q.nsx.nam);

    printf("\Nhap gia tri dien tro :");

    scanf("%f%c",&gt);q.giatri=gt;

    *p=q;

}

void in(DienTro p)

{

```

```
printf("\n\n%s%12d-%d-%d%12.1f",p.ten,p.nsx.ngay,p.nsx.thang,p.nsx.nam,p.giatri);

}
```

```
void SapXep(DienTro *p,int n)
```

```
{

    int i,j;

    for(i=1;i<=n;++i)

        for(j=i+1;j<=n;++j)

            if(p[i].nsx.nam>p[j].nsx.nam)

                DoiCho(&p[i],&p[j]);

}
```

```
void main()
```

```
{

    DienTro *p,ds[100];int n,i,j;char ten[40];

    /*vao so lieu */

    clrscr();

    printf("\n So dien tro n=");scanf("%d%c",&n);

    for(i=1;i<=n;i++)

        Nhap(&ds[i]);

    //sap xep theo chieu tang cua nam san xuat

    SapXep(ds,n);

    //in danh sach sau khi sap xep
```

```

printf("\n\nTen dien tro|Ngay san xuat|Gia tri");

for(i=1;i<=n;++i)

    in(ds[i]);

    //tim kiem theo ho ten

while(1)

{

    printf("\nNhap ten dien tro can tim\ (Bam Enter de ket thuc)");

    gets(ten);

    if(ten[0]==0) break;

    if((p=ptim(ten,ds,n))==NULL)

        printf("\n Khong tim thay");

    else in(*p);

}

    //tim kiem theo ho ten dung ham tim

while(1)

{

    printf("\nNhap ten dien tro can tim\ (bam Enter de ket thuc)");

    gets(ten);

    if(ten[0]==0) break;

    if(tim(ten,ds,n).ten[0]==0)

        printf("\nKhong tim thay");

```

```

        else in(tim(ten,ds,n));

    }          //ket thuc

}

```

Ví dụ 2 : Xét kiểu cấu trúc **sophuc**(số phức) gồm hai thành phần :

- + Biến **x** kiểu float biểu thị phần thực của số phức .
- + Biến **y** biểu thị phần ảo của số phức.

Hai hàm sau đây được sử dụng để thao tác trên kiểu **sophuc** .

+ Hàm

```

sophuc cong(sophuc u,sophuc v);

```

dùng để cộng các giá trị số phức **u,v**. Giá trị của hàm là cấu trúc chứa tổng **u+v** .

+ Hàm

```

void insp(sophuc u);

```

dùng để in cấu trúc **u**. Một điểm đáng lưu ý ở đây là: có thể dùng hàm này để in trực tiếp tổng của hai giá trị phức cho bởi hàm **cong()**. Dưới đây là nội dung chi tiết chương trình:

```

#include"stdio.h"

```

```

#include"conio.h"

```

```

typedef struct

```

```

{

```

```

    float x,y;

```

```

} sophuc;

```

```

sophuc cong(sophuc u ,sophuc v);

```

```

void insp(sophuc u);

```

```
sophuc cong(sophuc u,sophuc v)
```

```
{
```

```
    sophuc w;
```

```
    w.x=u.x+v.x;
```

```
    w.y=u.y+v.y;
```

```
    return w;
```

```
}
```

```
void insp(sophuc u)
```

```
{
```

```
    printf("(%.2f,%.2f)",u.x,u.y);
```

```
}
```

```
void main()
```

```
{
```

```
    clrscr;
```

```
    sophuc u,v;
```

```
    u.x=6.5;u.y=-3.6;
```

```
    v.x=2.8;v.y=12.1;
```

```
    printf("\nSo phuc u=");
```

```
    insp(u);
```

```
    printf("\n So phuc v=");
```

```
    insp(v);  
    printf("\n Tong=");  
    insp(cong(u,v));  
    getch();  
}
```