BUỔI 23: TÌM HIỂU VỀ STRUCT

FULL HOUSE

1. Định nghĩa:

- Là một kiểu dữ liệu người dùng tự định nghĩa (user defined datatype) cho phép bạn lưu trữ các loại phần tử khác nhau.
- Mỗi phần tử của một cấu trúc được gọi là một thành viên (member).
- Bạn có thể khai báo các loại phần tử khác nhau trong đó.

2. Khai báo

a. Cách 1

```
struct structureName {
    dataType member1;
    dataType member2;
    dataType member3;
    ...
};
```

b. Cách 2

```
struct structureName {
    dataType member1;
    dataType member2;
    ...
}List_of_variables;
```

c. Cách 3

```
struct {
    dataType member1;
    dataType member2;
    ...
}List_of_variables;
```

3. Ví dụ khai báo:

```
struct SinhVien {
    int maSV;
    char ho[20];
    char ten[20];
    bool gioiTinh;
    char queQuan[100];
};
```

```
struct Books{
    char  tieude[50];
    char  tacgia[50];
    char  chude[100];
    int   book_id;
}b1,b2,b3;
```

4. Cách sử dụng kiểu dữ liệu struct.

```c
#include <stdio.h>

struct sinhvien{
    float toan, van;
    char ho_ten[100];
};

int main(){
    sinhvien sv1, sv2;

    struct sinhvien sv3, sv4;

    struct sinhvien sv5[100];
}
```

5. Truy xuất từng thuộc tính trong struct

a. Không dùng con trỏ (truy cập bằng (.)).

```
struct sinhvien{
    float toan;
    char ho_ten[100];
};
void nhap(struct sinhvien &sv){
    printf("Nhap diem mon toan: ");
    scanf("%f",&sv.toan);
}

void xuat(struct sinhvien sv){
    printf("Diem toan: %.2f\n",sv.toan);
}
int main(){
    struct sinhvien sv1;
    nhap(sv1);
    xuat(sv1);
}
```

b. Dùng con trỏ (Truy cập bằng (->)).

```c
struct sinhvien{
    float toan;
    char ho_ten[100];
};
void nhap(struct sinhvien *sv){
    printf("Nhap diem mon toan: ");
    scanf("%f",&sv->toan);
}

void xuat(struct sinhvien *sv){
    printf("Diem toan: %.2f\n",sv->toan);
}
int main(){
    struct sinhvien sv1;
    nhap(&sv1);
    xuat(&sv1);
}
```

6. Typedef

- sử dụng từ khóa **typedef** để tạo ra một tên thay thế cho kiểu dữ liệu đã có. Nó thường được sử dụng kiểu struct để đơn giản hóa cú pháp khai báo biến. Nhưng nó cũng có thể sử dụng với các kiểu dữ liệu nguyên thủy.

```
struct sinhvien{
    ...
};
typedef sinhvien sv;

int main(){
    sv sv1,sv2;

    ...
    ...
}
```

7. Struct lồng nhau

```c
struct ngay_sinh{
    int ngay, thang, nam;
};

struct sinhvien{
    float toan, van;
    char ho_ten[100];
    struct ngay_sinh ns;
};

int main(){
    struct sinhvien sv1;
    printf("Nhap ngay sinh: ");
    scanf("%d",&sv1.ns.ngay);
```

8. Ví dụ:

Nhập xuất thông tin của một lô hàng. Thông tin gồm có:

- **Tên lô hàng**
- **Số lượng**
- **Ngày nhận (Ngày, tháng, năm).**

```
struct ngay_sx{
    int ngay, thang, nam;
};
typedef ngay_sx nsx;

struct lo_hang{
    char ten[100];
    int so_luong;
    nsx sx;
};
typedef lo_hang lh;
```

```
void nhap(lh &x){
     printf("Ten lo hang: ");
     gets(x.ten);
     printf("So luong hang: ");
     scanf("%d",&x.so_luong);
     getchar();
     printf("Ngay thang nam san xuat: ");
     scanf("%d%d%d",&x.sx.ngay,&x.sx.thang,&x.sx.nam);
     getchar();
}

void xuat(lh x){
     printf("\nTen lo hang: ");
     puts(x.ten);
     printf("So luong hang: ");
     printf("%d",x.so_luong);
     printf("\nNgay thang nam san xuat: ");
     printf("%d/%d/%d",x.sx.ngay,x.sx.thang,x.sx.nam);
}
```

```
}

int main(){
    lh x;
    nhap(x);
    xuat(x);
}
```