



CHƯƠNG 2

Kiểu Cấu Trúc

GIẢNG VIÊN

TS. Hà Ngọc Long

TIÊU CHUẨN ĐÁNH GIÁ SINH VIÊN

Hình thức Đánh giá	% Điểm
Điểm danh	15%
Kiểm tra trên máy tính 1	15%
Kiểm tra trên máy tính 2	20%
<i>Thi kết thúc học phần</i> <i>(Làm bài trên máy tính)</i>	50%
Tổng cộng:	100%

NỘI DUNG MÔN HỌC

1

- Kiểu Chuỗi

2

- Kiểu Cấu Trúc

3

- Kiểu Tập Tin



Căn bản về kiểu cấu trúc (struct)

Nội dung chính

1. Khái niệm cấu trúc

- Khái niệm

2. Khai báo cấu trúc

- Khai báo kiểu cấu trúc
- Khai báo biến cấu trúc
- Định nghĩa kiểu dữ liệu với **typedef**

3. Xử lý dữ liệu cấu trúc

- Truy cập các trường dữ liệu
- Phép gán giữa các biến cấu trúc
- Một số ví dụ

Ví dụ: Quản lý thí sinh thi Đại học (1/3)

- ❑ Để quản lý cần lưu trữ các thông tin
 - Số báo danh: Số nguyên không dấu
 - Họ tên sinh viên: Chuỗi ký tự không quá 30
 - Khối thi: Ký tự (A,B,C..)
 - Tổng điểm 3 môn thi: kiểu thực
- Do vậy với mỗi sinh viên cần các biến

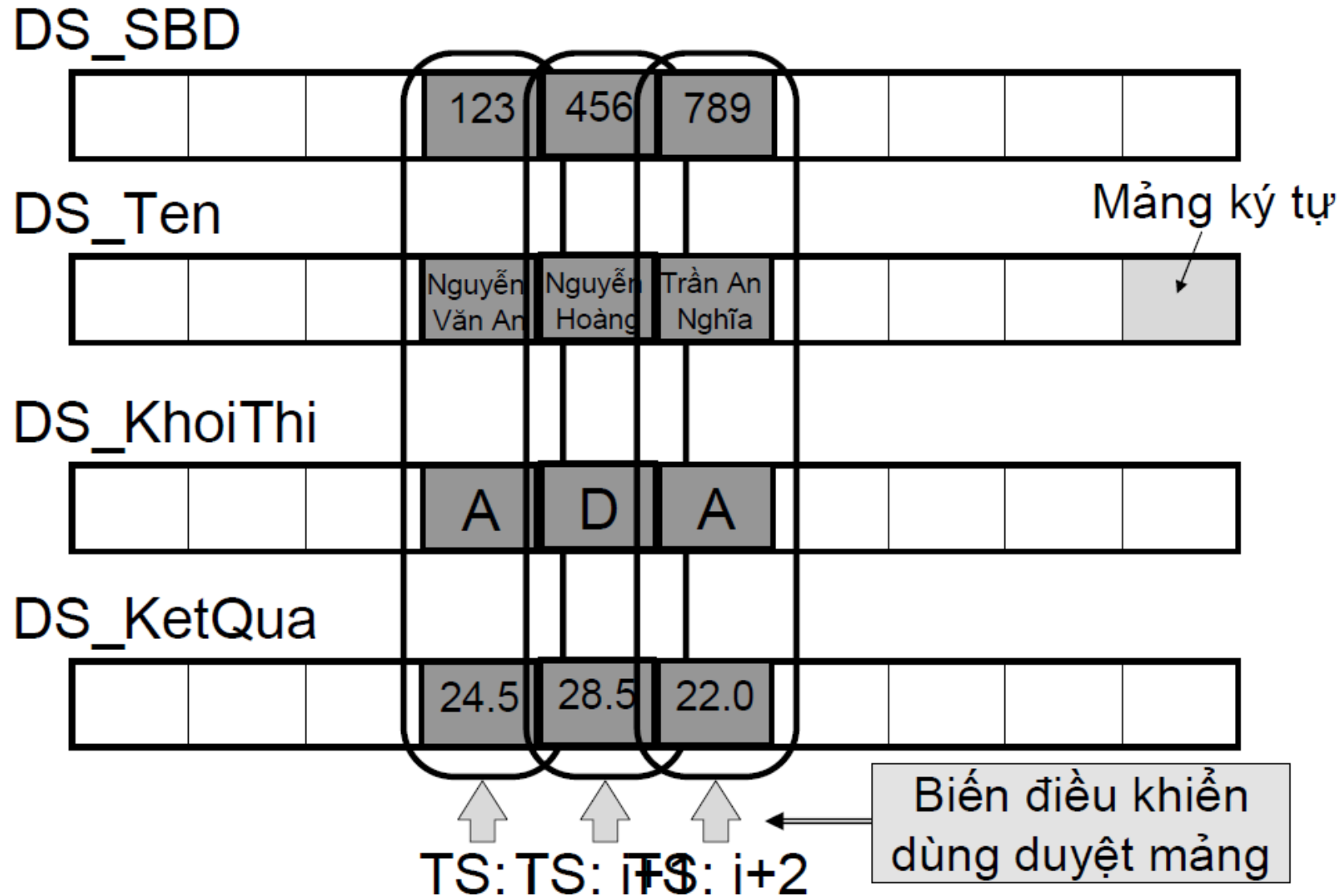
<code>unsigned</code>	<code>SBD;</code>
<code>char</code>	<code>Ten[30];</code>
<code>char</code>	<code>KhoiThi;</code>
<code>float</code>	<code>KetQua;</code>

Ví dụ: Quản lý thí sinh thi Đại học (2/3)

Để quản lý danh sách (dưới 1000) thí sinh dự thi, cần nhiều mảng rời rạc

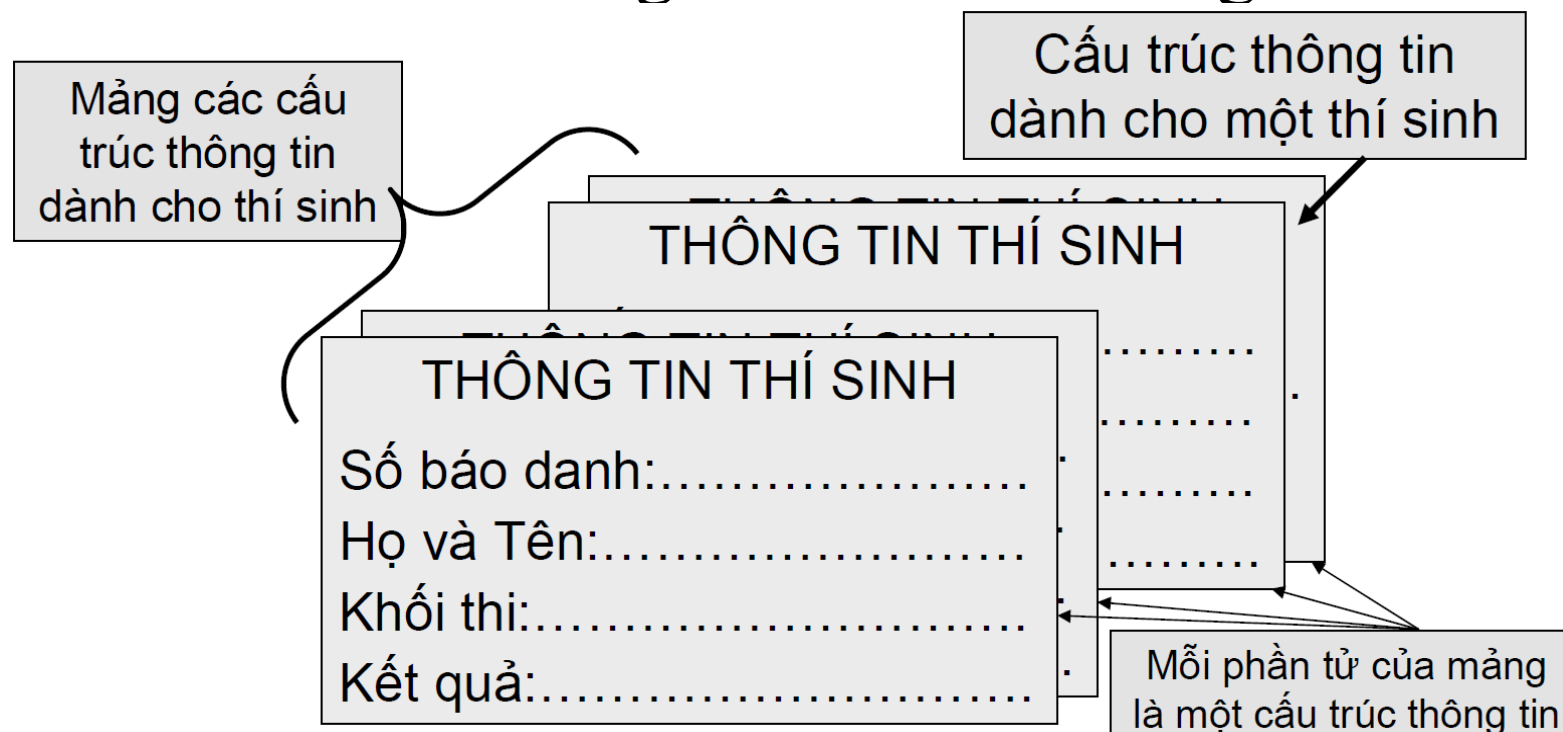
```
#define MAX      1000  
  
unsigned        DS_SBD[MAX];  
  
char            DS_Ten[MAX][30];  
  
char            DS_KhoiThi[MAX];  
  
float           DS_KetQua[MAX];
```

Ví dụ: Quản lý thí sinh thi Đại học (3/3)



Vấn đề & Giải pháp

- Dùng nhiều mảng
- - Khó quản lý, dễ nhầm lẫn
- - Không thể hiện cấu trúc thông tin dành cho từng thí sinh



Khái niệm Cấu trúc

- **Cấu trúc** là *kiểu dữ liệu phức hợp*, do người dùng *tự định nghĩa*
 - Kiểu cấu trúc bao gồm nhiều thành phần có thể thuộc các kiểu dữ liệu khác nhau
 - Các thành phần: gọi là trường dữ liệu (**field**)
 - Các thành phần, không được truy nhập theo chỉ số (như mảng) mà theo tên của trường.
- Có thể coi một biến cấu trúc là một tập hợp của một hay nhiều biến rời rạc, thường có kiểu khác nhau thành một biến có một tên duy nhất để dễ dàng quản lý và sử dụng

Ví dụ

1) Kết quả học tập của sinh viên

- TenSV: Chuỗi ký tự
- MaSV: Chuỗi số/ số nguyên
- Điểm: Số thực

2) Điểm trong mặt phẳng

- Tên điểm: Ký tự (A, B, C..)
- Hoành độ: Số thực
- Tung độ: Số thực

Nội dung chính

1. Khái niệm cấu trúc

- Khái niệm

2. Khai báo cấu trúc

- Khai báo kiểu cấu trúc
- Khai báo biến cấu trúc
- Định nghĩa kiểu dữ liệu với **typedef**

3. Xử lý dữ liệu cấu trúc

- Truy cập các trường dữ liệu
- Phép gán giữa các biến cấu trúc
- Một số ví dụ

Khai báo kiểu cấu trúc

```
struct Tên_kiểu_cấu_trúc {  
    <Khai báo các trường dữ liệu>  
};
```

- **struct:** từ khóa, cho phép người dùng khai báo kiểu cấu trúc
- **Tên_kiểu_cấu_trúc:** Tên của kiểu cấu trúc do người dùng tự định nghĩa
 - Tuân theo nguyên tắc đặt tên đối tượng trong C
- **Khai báo các trường dữ liệu:** Danh sách các khai báo thành phần (*trường:field*) của cấu trúc
 - Giống khai báo biến
 - Các trường có thể có kiểu bất kỳ

Ví dụ

Thẻ sinh viên

Số hiệu:...(Chuỗi ký tự)..
Tên sinh viên: (Chuỗi ký tự)
Năm sinh:...(Số nguyên)...
Khóa:.....(Số nguyên).....
Lớp:..... :(Chuỗi ký tự). ...

```
struct SinhVien{  
    char SHSV[10];  
    char Ten[30];  
    int NS;  
    int Khoa;  
    char Lop [10];  
};
```

Point2D

Hoành độ (x)...(Số thực)..
Tung độ (y).....(Số thực)..

```
struct Point{  
    float x, y;  
};
```

Khai báo biến cấu trúc

- Khai báo **kiểu** cấu trúc nhằm tạo định nghĩa toàn thể cho các cấu trúc sẽ được dung sau này
 - Không cung cấp không gian nhớ cho kiểu
- Khai báo **biến** cấu trúc nhằm yêu cầu chương trình tạo vùng nhớ để lưu trữ các dữ liệu cho biến cấu trúc
 - Chứa dữ liệu của các trường của cấu trúc

Cú pháp khai báo

- Tồn tại định nghĩa kiểu cấu trúc

```
struct Kiểu_cấu_trúc Tên_biến;
```

- Khai báo trực tiếp

```
struct {  
    <Khai báo các trường dữ liệu>  
} Tên_biến;
```

- Kết hợp với khai báo kiểu

```
struct Kiểu_cấu_trúc {  
    <Khai báo các trường dữ liệu>  
} Tên_biến;
```


Ví dụ

- Tồn tại định nghĩa kiểu cấu trúc

```
struct SinhVien SV1, SV2, Thu_khoa;
```

- Khai báo trực tiếp

```
struct {  
    float x, y; //Tọa độ trên mặtP  
}A, B;          //Khai báo 2 điểm A, B
```

- Kết hợp với khai báo kiểu

```
struct Point_3D {  
    float x, y, z; //Tọa độ ko gian  
}A, B;
```

Chú ý (1/3)

- Các cấu trúc có thể được khai báo lồng nhau

```
struct diem_thi {  
    float dToan, dLy, dHoa;  
}  
  
struct thi_sinh{  
    char SBD[10];  
    char ho_va_ten[30];  
    struct diem_thi ket_qua;  
} thi_sinh_1, thi_sinh_2;
```

Chú ý (2/3)

- Có thể khai báo trực tiếp các trường dữ liệu của một cấu trúc bên trong cấu trúc khác

```
struct thi_sinh{  
    char SBD[10];  
    char ho_va_ten[30];  
    struct{  
        float dToan, dLy, dHoa;  
    } ket_qua;  
} thi_sinh_1, thi_sinh_2;
```

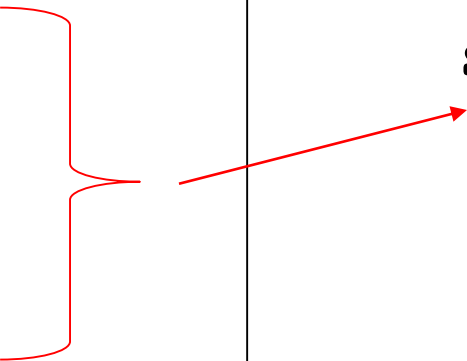
Chú ý (3/3)

- Có thể gán giá trị khởi đầu cho một biến cấu trúc, theo nguyên tắc như kiểu mảng. Ví dụ:

```
struct Date{
    int day;
    int month;
    int year;
};

struct SinhVien{
    char Ten[20];
    struct Date NS;
} SV = {"Tran Anh", 20, 12, 1990 };
```

```
struct SinhVien{
    char Ten[20];
    struct Date{
        int day;
        int month;
        int year;
    } NS;
} SV = {"Tran Anh", 20, 12, 1990 };
```



**Printf => SV.Ten;
SV.NS.day**

Định nghĩa kiểu dữ liệu với typedef (1/2)

```
typedef <tên_cũ> <tên_mới>;
```

Mục đích

- Đặt tên mới đồng nghĩa với tên của một kiểu dữ liệu đã được định nghĩa
 - Thường được sử dụng cho kiểu cấu trúc
 - Giúp cho khai báo trở nên quen thuộc và ít bị sai hơn

Ví dụ

```
typedef char Str80[80] ;  
typedef long mask;  
Str80 str="Bonjour tout le monde !";  
mask a, b;
```

Định nghĩa kiểu dữ liệu với typeof (2/2)

- Thường được kết hợp với kiểu cấu trúc để khai báo một ***bí danh*** cho một cấu trúc => *Giúp khai báo trở nên quen thuộc và ít bị sai hơn*

```
typedef struct { //Định nghĩa một cấu trúc
```

```
    char SHSV[10];
```

```
    char Ten[30];
```

```
    int NS;
```

```
    int Khoa;
```

```
    char Lop [10];
```

```
} SinhVien; //Đặt tên cho cấu trúc là SinhVien
```

```
SinhVien SV; //Tạo một biến cấu trúc
```

Chú ý (1/2)

- Cho phép đặt tên mới trùng với tên cũ

Ví dụ:

```
struct point_3D{  
    float x, y, z;  
}  
struct point_3D M;  
typedef struct point_3D  
point_3D;  
point_3D N;
```

```
typedef struct {  
    float x, y, z;  
}point_3D;  
point_3D M;  
point_3D N;
```

Chú ý (2/2)

```
typedef struct point_2D {  
    float x, y;  
}point_2D, diem_2_chieu,  
ten_bat_ki;  
point_2D X;  
diem_2_chieu Y;  
ten_bat_ki Z;
```

- **Chú ý:**

point_2D, diem_2_chieu, ten_bat_ki là các tên cấu trúc,
không phải tên biến

Nội dung chính

1. Khái niệm cấu trúc

- Khái niệm

2. Khai báo cấu trúc

- Khai báo kiểu cấu trúc
- Khai báo biến cấu trúc
- Định nghĩa kiểu dữ liệu với **typedef**

3. Xử lý dữ liệu cấu trúc

- Truy cập các trường dữ liệu
- Phép gán giữa các biến cấu trúc
- Một số ví dụ

Truy cập các trường dữ liệu

- **Cú pháp**

tên_biến_cấu_trúc.tên_trường

- **Lưu ý**

- Dấu “.” là toán tử truy cập vào trường dữ liệu trong cấu trúc
- Nếu trường dữ liệu là một cấu trúc => sử dụng tiếp dấu “.” để truy cập vào thành phần mức sâu hơn

Ví dụ 1

```
#include <stdio.h>
void main(){
    struct{
        char Ten[20];
        struct Date{
            int day;
            int month;
            int year;
        } NS;
    } SV = {"Tran Anh", 20, 12, 1990 };
    printf(" Sinh vien %s (%d/%d/%d)",
        SV.Ten, SV.NS.day, SV.NS.month, SV.NS.year);
}
```

Sinh vien Tran Anh (20/12/1990)

Ví dụ 2 : Biểu diễn điểm 2D (1/2)

- Xây dựng một cấu trúc biểu diễn điểm trong không gian 2 chiều. Nhập giá trị cho một biến kiểu cấu trúc này, sau đó hiển thị giá trị các trường dữ liệu của biến này ra màn hình.
 - Cấu trúc: tên điểm, tọa độ x, tọa độ y
 - Nhập, hiển thị từng trường của biến cấu trúc như các biến dữ liệu khác.

Ví dụ 2 : Biểu diễn điểm 2D (2/2)

```
#include<stdio.h>
#include<conio.h>
typedef struct{
    char ten[5];
    int x,y;
}toado;
void main(){
    toado t;
    printf("Nhap thong tin toa do\n");
    printf("Ten diem: ");gets(t.ten);
    printf("Toa do x: ");scanf("%d",&t.x);
    printf("Toa do y: ");scanf("%d",&t.y);
    printf("Gia tri cac truong\n");
    printf("%-5s%3d%3d\n",t.ten,t.x,t.y);
    getch();
}
```

```
Nhap thong tin toa do
Ten diem: A
Toa do x: 12
Toa do y: 6
Gia tri cac truong
A      12  6
```

Phép gán giữa các biến cấu trúc

- Muốn sao chép dữ liệu từ biến cấu trúc này sang biến cấu trúc khác cùng kiểu
 - Gán lần lượt từng trường trong hai biến cấu trúc
 - C cung cấp phép gán hai biến cấu trúc cùng kiểu:
`biến_cấu_trúc_1 = biến_cấu_trúc_2;`

Ví dụ (1/2)

- **Ví dụ:** Xây dựng cấu trúc gồm họ tên và điểm TĐC của sinh viên
 - a, b, c là 3 biến cấu trúc.
 - Nhập giá trị cho biến a.
 - Gán $b=a$,
 - gán từng trường của a cho c.
 - So sánh a, b và c ?

Ví dụ (2/2)

```
#include<stdio.h>
#include<conio.h>
typedef struct{
    char hoten[20];
    int diem;
}sinhvien;
void main(){
    sinhvien a,b,c;
    printf("Nhap thong tin sinh vien\n");
    printf("Ho ten: "); gets(a.hoten);
    printf("Diem:"); scanf("%d",&a.diem);
    b = a;
    strcpy(c.hoten,a.hoten);
    c.diem = a.diem;
```

```
printf("Bien a: ");
printf("%-20s%3d\n",a.hoten,a.diem);
printf("Bien b: ");
printf("%-20s%3d\n",b.hoten,b.diem);
printf("Bien c: ");
printf("%-20s%3d\n",c.hoten,c.diem);
getch();
}
```

```
Nhap thong tin sinh vien
Ho ten: LE VAN ANH
Diem:10
Bien a: LE VAN ANH          10
Bien b: LE VAN ANH          10
Bien c: LE VAN ANH          10
```


Bài tập

- 1) **Lập trình đọc vào một danh sách không quá 100 sinh viên** gồm: Họ tên, năm sinh
 1. Đưa ra DS những sinh viên sinh năm 2000
 2. Đưa ra DSSV đã sắp xếp theo thứ tự ABC
- 2) **Lập trình đọc vào DS thí sinh** gồm Họ tên, điểm thi 3 môn Toán, Lý, Hóa, kết thúc nhập khi gặp sinh viên có tên rỗng
 1. Đọc tiếp vào một điểm chuẩn; đưa ra danh sách thí sinh trúng tuyển (không có điểm liệt - 0)
 2. Đưa ra thí sinh cao điểm nhất
 3. Tìm điểm chuẩn, nếu chỉ lấy K SV, K nhập vào. Nếu có nhiều người bằng điểm nhau; loại cả

Giải bài 1

```
#include <stdio.h>
#include <string.h>
typedef struct{
    char Ten[30];
    int NS;
}SinhVien;
void main(){
    SinhVien DS[100], SV;
    int N,i,j;
    printf("Nhap So sinh vien : ");
    scanf("%d",&N);
    fflush(stdin);
    for ( i=0; i < N; i++ ){
        fflush(stdin);
        printf("Nhap du lieu cho sinh vien %d: \n", i+1);
        printf("Ho ten : "); gets(DS[i].Ten);
        printf("Nam sinh :");scanf("%d", &DS[i].NS);
    }
    printf("\n\n DANH SACH SINH VIEN\n\n");
```

```
    for(i = 0; i < N; i ++){
        if(DS[i].NS == 2000) //Chỉ in SV sinh năm 2000
            printf("%s\n",DS[i].Ten);
        for(i = 0; i < N - 1; i ++){
            for(j = i+1; j < N; j ++){
                if(strcmp(DS[i].Ten,DS[j].Ten) > 0){
                    SV= DS[i];
                    DS[i]=DS[j];
                    DS[j] = SV;
                }
            }
        }
        printf("\n\n DANH SACH SINH VIEN DA SAP XEP\n\n");
        for(i = 0; i < N; i ++){
            printf("%s\n",DS[i].Ten);
        }
    }
} //main
```

Kết quả bài 1

```
Nhap So sinh vien : 5
Nhap du lieu cho sinh vien 1:
Ho ten : NGUYEN THI BUOI
Nam sinh :1999
Nhap du lieu cho sinh vien 2:
Ho ten : LE THI XINH
Nam sinh :2000
Nhap du lieu cho sinh vien 3:
Ho ten : TRAN THI MO
Nam sinh :2001
Nhap du lieu cho sinh vien 4:
Ho ten : LE THI DEP
Nam sinh :2000
Nhap du lieu cho sinh vien 5:
Ho ten : NGUYEN VAN ANH
Nam sinh :2000
```

DANH SACH SINH VIEN

```
LE THI XINH
LE THI DEP
NGUYEN VAN ANH
```

DANH SACH SINH VIEN DA SAP XEP

```
LE THI DEP
LE THI XINH
NGUYEN THI BUOI
NGUYEN VAN ANH
TRAN THI MO
```

Bài tập

3) Lập trình thực hiện các công việc sau

- Đọc vào từ bàn phím một danh sách thuốc gồm
 - Tên thuốc (chuỗi không quá 20 ký tự)
 - Năm hết hạn
 - Số lượng còn
 - Đơn giá

Kết thúc nhập khi gặp thuốc có tên »*** »

- Đưa danh sách thuốc ra màn hình
- Đưa ra danh sách các thuốc đã hết hạn
- Xóa khỏi danh sách những thuốc đã hết hạn. Đưa danh sách mới ra màn hình
- Tính tổng giá trị các thuốc đã hết hạn

Bài tập

4) Cho một danh sách thành tích thi đấu bóng đá của 32 đội tuyển bao gồm: Tên đội bóng, số bàn thắng, số bàn thua, số thẻ đỏ, số thẻ vàng

Viết chương trình thực hiện:

- Nhập dữ liệu vào từ bàn phím
- Nhập vào tên đội bóng,
 - đưa ra thành tích của đội này
 - Nếu không tồn tại, thông báo: không tìm thấy
- Tính và đưa ra màn hình số điểm của các đội nếu
 - Mỗi bàn thắng được tính 10 điểm
 - Mỗi bàn thua bị phạt 5 điểm, mỗi thẻ vàng trừ 2 điểm, thẻ đỏ trừ 5 điểm



Cấu trúc và Con trỏ

Nội dung

- 1) Các cấu trúc bao nhau
- 2) Con trỏ trỏ đến cấu trúc
- 3) Mảng con trỏ trỏ đến cấu trúc
- 4) Thành phần của cấu trúc là con trỏ
- 5) Tham số của hàm là cấu trúc
- 6) Một số ví dụ

**BÀI TẬP NHÓM 1
BÁO CÁO VÀO
BUỔI SAU**



Cấu trúc - Danh sách liên kết đơn - Lớp list trong C++

Nội dung

- 1) Cấu trúc danh sách liên kết đơn
- 2) Khái niệm về danh sách liên kết đơn
- 3) Các phép toán trên danh sách liên kết đơn
- 4) Giới thiệu lớp list của C++
- 5) Một số ví dụ minh họa

**BÀI TẬP NHÓM 2
BẢO CÁO VÀO
BUỔI SAU**

Hoạt động nhóm

- Mỗi nhóm chọn *một trong những chủ đề* sau đây về tập tin (File).
- Xây dựng slide báo cáo nhóm theo cấu trúc nội dung được giao.
- ***Yêu cầu:***
 - **Về nội dung (60%):** Tối thiểu là 30 slide. Nêu rõ ràng khái niệm, định nghĩa, kèm ví dụ (code), giải thích code, và code minh họa
 - **Về hình thức (20%):** Trình bày bài báo cáo bắt mắt, dễ hiểu, súc tích
 - **Về trình bày (20%):** Người trình bày nói to, rõ ràng, lưu loát
 - Làm xong nộp vào email: hnlong@hueuni.edu.vn

Q&A

