

# BÀI GIẢNG MÔN LẬP TRÌNH NÂNG CAO

Giảng viên: TS.GVC Bùi Thị Thanh Xuân

Bộ môn: Tin học và KTTT

Năm học: 2020-2021

# 1959

#### Chương 4: Kiểu cấu trúc và hợp

- 4.1. Kiểu cấu trúc
- 4.2. Cấu trúc tự trỏ và danh sách liên kết
- 4.2.1. Cấu trúc tự trỏ
- 4.2.2. Khái niệm danh sách liên kết đơn
- 4.2.3. Các phép toán trên danh sách liên kết đơn
- 4.3. Con trở tới thành phần
- 4.4. Kiểu hợp
- 4.5. Kiểu liệt kê

# 1959

# Ví dụ → Bài toán quản lý thí sinh thi đại học

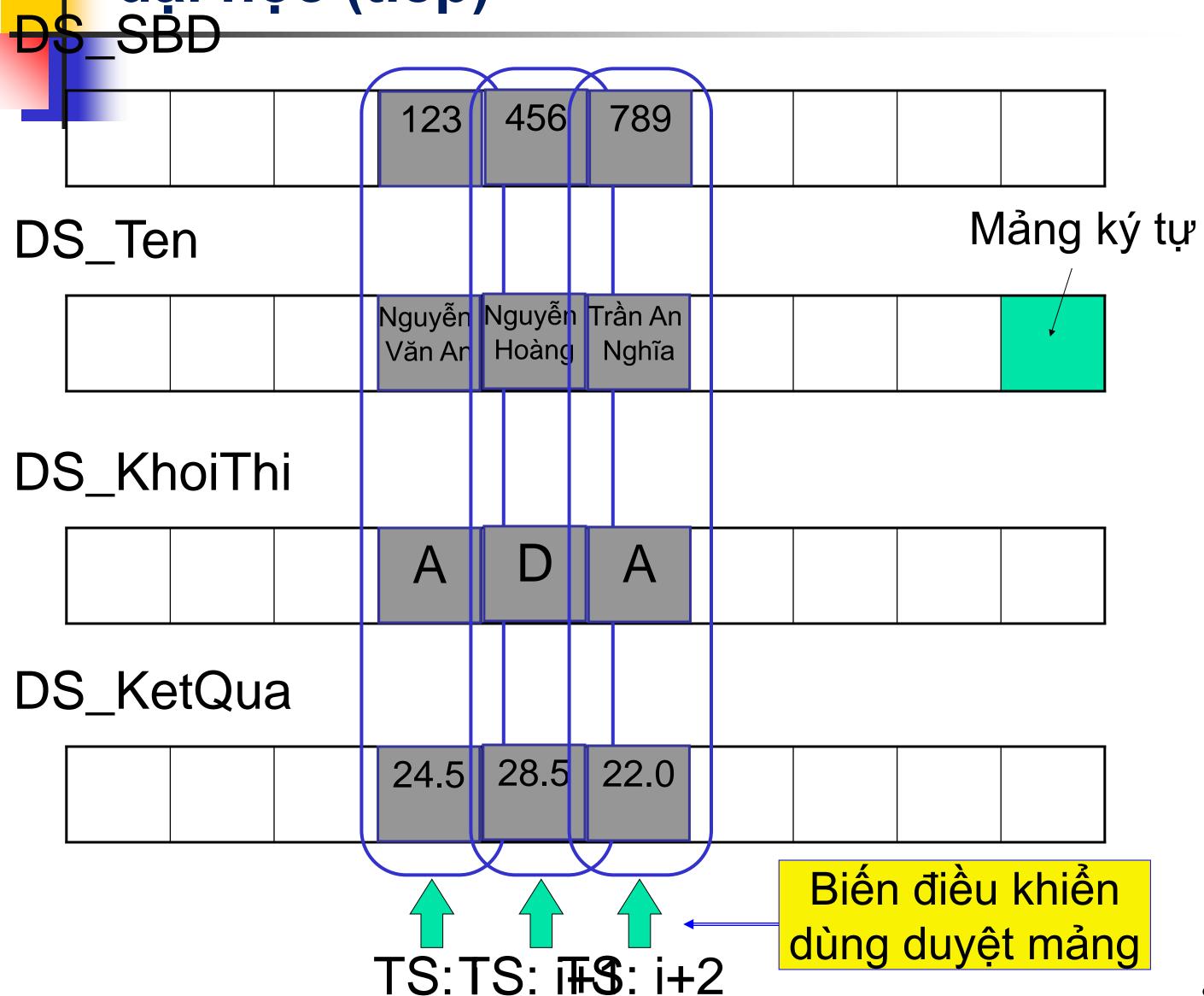
Đế quản lý cần lưu trữ các thông tin

- Số báo danh: Số nguyên không dấu
- Họ tên sinh viên: Chuỗi ký tự (thường không quá 30)
- Khối thi: Ký tự (A,B,C..)

Tổng điểm 3 môn thi: kiểu thực

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

# Ví dụ → Bài toán quản lý thí sinh the đại học (tiếp) SBD





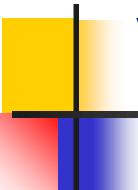


Mảng các cấu trúc thông tin dành cho thí sinh Cấu trúc thông tin dành cho một thí sinh

THÔNG TIN THÍ SINH Số báo THÔNG TIN THÍ SINH Số báo danh:.. Họ và Tên:. Khối Kết

Mỗi phần tử của mảng là một cấu trúc thông tin





# 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 kiểu cấu trúc

- Kiểu cấu trúc là kiểu dữ liệu phức hợp, do người dụng 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.

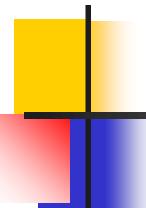
Có thể coi một biến cấu trúc là một tập hợp của một hay nhiều biến rời rạc, thường có kiểu khác nhau thành một biến có một tên duy nhất để dễ dàng quản lý và sử dụng.



#### Khái niệm → Ví dụ

- Kết quả học tập của sinh viên
  - -TenSV: Chuỗi ký tự
  - -MaSV: Chuỗi số/ số nguyên
  - -Điem: Số thực
- Điểm trong mặt phẳng
  - -Tên điểm: Ký tự (A, B, C..)
  - -Hoành độ: Số thực
  - -Tung độ: Số thực





#### 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 dữ liệu mới: kiểu cấu trúc
- Tên\_kiểu\_cấu\_truc: 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



#### Khai báo kiểu cấu trúc → Ví dụ

Thẻ sinh viên Sợ hiệu:...(C*huỗi 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 dùng 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





#### Khai báo biến cấu trúc → Cú pháp

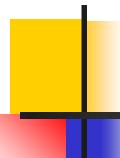
Tồn tại định nghĩa kiểu cấu trúc

```
Kiểu_cấu_ trúc Tên_biến;
```

Khai báo trực tiếp

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

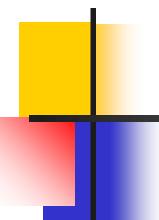




#### Khai báo biến cấu trúc → Cú pháp

#### Kết hợp với khai báo kiểu





## Khai báo biến cấu trúc → Ví dụ

Tồn tại định nghĩa kiểu cấu trúc

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

#### Khai báo trực tiếp

```
struct {
  float x, y; //Topa đô trên mặt
  phẳng
}A, B; //Khai báo 2 điểm A, B
```





# Khai báo biến cấu trúc → Ví dụ

Kết hợp với khai báo kiểu

```
struct Point_3D{
  float x, y, z;// Toa đô không gian
}A, B;
```



#### 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



Nhập danh sách trúng tuyển gồm n sinh viên với các thông tin: Mã SV, Họ và tên, Điểm trúng tuyển
In ra danh sách trúng tuyển đã nhập

```
Whap so sinh vien, n = 3
Whap danh sach sinh vien:
Nhap Ma SV: SV1
Nhap Ten SV: Hoang Mai
Vhap Diem: 23
Nhap Ma SV: SV2
Nhap Ten SV: Nguyen Anhi
Vhap Diem: 26
Vhap Ma SV: SV3
     <u>Ten SV: Tran Hai</u>
Whap Diem: 25
  en thi danh sach:
                          Diem
23
2<u>6</u>
  TS
          Ho Ten
        Hoang Mai
         Nguyen <u>Anh</u>i
         Tran Hai
```



Nhập danh sách trúng tuyển gồm n sinh viên với các thông tin: Mã TS, Họ và tên, Điểm trúng tuyển
In ra danh sách trúng tuyển

```
struct SinhVien{
    string MaTS;
    string HoTen;
    float Diem;
};
```

SinhVien ds[100];

```
Nhap so sinh vien, n = 3
Whap danh sach sinh vien:
Vhap Ma SV: SV1
Mhap Ten SV: Hoang Mai
lhap Diem: 23
Jhap Ma SV: SV2
Nhap Ten SV: Nguyen Anhi
√hap Diem: 26
lhap Ma SV: SV3
     <u>Ten SV: Tran Hai</u>
Whap Diem: 25
 en thi danh sach:
                         Diem
23
2<u>6</u>
  TS
         Ho Ten
        Hoang Mai
        Nguyen Anhi
         Tran Hai
```





```
#include <iostream>
#include <string>
using namespace std;
struct SinhVien{
        string MaTS;
        string HoTen;
        float Diem;
void NhapDS(SinhVien ds[], int n){
    for(int i = 0; i<n; i++){
        cin.ignore();
        cout<<"Nhap Ma SV: "; getline(cin,ds[i].MaTS);</pre>
        cout<<"Nhap Ten SV: "; getline(cin,ds[i].HoTen);</pre>
        cout<<"Nhap Diem: "; cin>>ds[i].Diem;
void HienThiDS(SinhVien ds[], int n){
    cout<<"Ma TS \t Ho Ten \t Diem"<<endl;
    for(int i = 0; i<n; i++)
         cout<<ds[i].MaTS<<"\t"<<ds[i].HoTen <<"\t"<<ds[i].Diem<<endl;</pre>
int main(){
    SinhVien ds[100];
    int n;
    cout<<"Nhap so sinh vien, n ="; cin>>n;
    cout<<"Nhap danh sach sinh vien:"<<endl;
    NhapDS(ds,n);
    cout<<"Hien thi danh sach:"<<endl;
    HienThiDS(ds,n);
    return 0;
```



Nhập danh sách trúng tuyển gồm n sinh viên với các thông tin: Mã SV, Họ và tên, Điểm trúng tuyển
In ra danh sách trúng tuyển tuyển sắp xếp tăng dần theo điểm trúng tuyển

```
Whap so sinh vien, n = 3
Whap danh sach sinh vien:
Nhap Ma SV: SV1
Nhap Ten SV: Hoang Mai
Whap Diem: 23,
Nhap Ma SV: SV2
Nhap Ten SV: Nguyen Anh
Vhap Diem: 26
Vhap Ma SV: SV3
     <u>Ten SV: Tran Hai</u>
Vhap Diem: 25
 en thi danh sach:
                         Diem
23
2<u>6</u>
         Ho Ten
        Hoang Mai
        Nguyen Anhi
```



```
void swap(SinhVien *sv1, SinhVien *sv2){
    SinhVien sv = *sv1;
    *sv1 = *sv2;
    *sv2 = sv;
}
void sap_xep_diem(SinhVien ds[], int n){
    for(int i = 0; i<n-1; i++)
        for(int j = i+1; j<n; j++)
        if(ds[i].Diem>ds[j].Diem)
            swap(ds[i],ds[j]);
}
```

```
#include <iostream>
#include <string>
using namespace std;
struct SinhVien{
       string MaTS;
       string HoTen;
       float Diem;
void NhapDS(SinhVien ds[], int &n){
    cout<<"Nhap so sinh vien n = "; cin>>n;
   for(int i = 0; i<n; i++){</pre>
       cin.ignore();
       cout<<"Nhap Ma SV: "; getline(cin,ds[i].MaTS);</pre>
       cout<<"Nhap Ten SV: "; getline(cin,ds[i].HoTen);</pre>
       cout<<"Nhap Diem: "; cin>>ds[i].Diem;
void HienThiDS(SinhVien ds[], int n){
    cout<<"Ma TS \t Ho Ten \t Diem"<<endl;
    for(int i = 0; i<n; i++)
         cout<<ds[i].MaTS<<"\t"<<ds[i].HoTen <<"\t\t"<<ds[i].Diem<<endl;
void swap(SinhVien *sv1, SinhVien *sv2){
    SinhVien sv = *sv1;
    *sv1 = *sv2;
                                                     int main(){
    *sv2 = sv;
                                                         SinhVien ds[100];
                                                         int n;
void sap xep diem(SinhVien ds[], int n){
                                                         NhapDS(ds,n);
    for(int i = 0; i<n-1; i++)
                                                         sap xep diem(ds,n);
         for(int j = i+1; j < n; j++)
                                                         HienThiDS(ds,n);
         if(ds[i].Diem>ds[j].Diem)
                                                         return 0;
             swap(ds[i],ds[j]);
```



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];
 diem_thi ket_qua;
} thi_sinh_1, thi sinh 2;
```



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;
```



Có thể gán giá trị khởi đầu cho một biến cấu trúc, theo nguyên tắc như kiểu mảng

```
Ví du:
    struct Date {
        int day;
        int month;
        int year;
     };
     struct SinhVien{
        char Ten[30];
        Date NS;
     } SV = {"Tran Anh",20,12,1990}, x;
```



```
struct SinhVien{
     string MaTS;
     string HoTen;
     struct Date {
       int day;
       int month;
       int year;
     } NgaySinh;
     float Diem;
  };
SinhVien SV;
```

```
struct Date {
    int day;
    int month;
    int year;
struct SinhVien{
    string MaTS;
    string HoTen;
    Date NgaySinh;
    float Diem;
  };
SinhVien SV;
```



#### Ví dụ 1

```
#include <iostream>
#include <string>
                            Tai thong tin SV:
                                              12/5/2002
using namespace std;
                                Hoang Bach
struct SinhVien{
                         Process exited after 0.2106 seconds with re
      string MaTS;
                          ress any key to continue . . .
      string HoTen;
      struct Date {
         int day;
         int month;
         int year;
      } NgaySinh;
      float Diem;
   };
int main(){
   SinhVien SV1 = {"M01", "Hoang Bach", 12,5,2002,23};
   cout<<"In lai thong tin SV:"<<endl;</pre>
   cout<<SV1.MaTS<<"\t"<<SV1.HoTen
<<"\t"<<SV1.NgaySinh.day<<"/"<<
SV1.NgaySinh.month<<"/"<<SV1.NgaySinh.year<<"\t"<<SV1
.Diem;
   return 0;
```



#### Ví dụ 2

```
#include <iostream>
#include <string>
                           Vhap Ma SV: A1
                           Vhap Ten SV: Hoang Long
using namespace std;
                          Nhap ngay sinh SV: 23
struct Date {
                           Whap thang sinh SV: 4
    int day;
                           Whap nam sinh SV: 2002
    int month;
                          Nhap Diem: 26
    int year;
                             lai thong tin SV:
  };
                                                   23/4/2002
                                                                   26
                                  Hoang Long
struct SinhVien{
       string MaTS;
       string HoTen;
       Date NgaySinh;
       float Diem;
   };
int main(){
   SinhVien SV;
   cout<<"Nhap Ma SV: "; getline(cin, SV.MaTS);</pre>
   cout<<"Nhap Ten SV: "; getline(cin,SV.HoTen);</pre>
   cout<<"Nhap ngay sinh SV: "; cin>>SV.NgaySinh.day;
   cout<<"Nhap thang sinh SV: "; cin>> SV.NgaySinh.month;
   cout<<"Nhap nam sinh SV: "; cin>>SV.NgaySinh.year;
   cout<<"Nhap Diem: "; cin>>SV.Diem;
   cout<<"In lai thong tin SV:"<<endl;</pre>
   cout<<SV.MaTS<<"\t"<<SV.HoTen <<"\t"<<SV.NgaySinh.day<<"/"<<</pre>
SV.NgaySinh.month<<"/"<<SV.NgaySinh.year<<"\t"<<SV.Diem;</pre>
   return 0;
```





- Nhập danh sách trúng tuyển gồm n sinh viên với các thông tin: Mã SV, Họ và tên, Ngày sinh, Điểm trúng tuyển
- In ra danh sách trúng tuyến
- Sắp xếp danh sách theo họ tên theo abc
- Tìm sinh viên có điểm trúng tuyển cao nhất



#### Định nghĩa kiểu dữ liệu với typedef

```
typedef <tên_cũ> <tên_mới>;
```

#### Muc 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 long mask;
typedef double db;
```

```
mask a, b;
db x, y;
```





#### Định nghĩa kiếu dữ liệu với typedef

Thường được kết hợp với kiếu cấu trúc đế khai báo một bí danh cho một cấu trúc

-Giúp khai báo trở nên quen thuộc và ít bị sai hơn

```
typedef struct{
  string MaTS;
  string HoTen;
  float Diem;
 } SinhVien; //Đặt tên cho cấu trúc là SinhVien
SinhVien SV; //Tạo một biến cấu trúc
```





Cho phép đặt tên mới trùng với tên cũ

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



#### Chú ý

```
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
```





Bài toán: Xây dựng một cấu trúc biểu diễn điểm trong không gian 2 chiều.

- -Nhập giá trị cho một biến kiểu cấu trúc này
- -Hiển thị giá trị các trường dữ liệu của biến này ra màn hình.

#### Thực hiện:

- -Cấu trúc gồm: 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

```
#include<iostream>
                               Nhap thong tin toa do
#include<cstring>
                               Ten diem: A
using namespace std;
                               Toa do x:
                               Toa do y: 3
typedef struct {
                               Gia tri cac truong
  char ten[2];
  int x, y;
}toado;
int main() {
  toado t;
  cout<<"Nhap thong tin toa do\n";</pre>
  cout<<"Ten diem: "; cin.getline(t.ten,2);</pre>
  cout<<"Toa do x: "; cin>>t.x;
  cout<<"Toa do y: "; cin>>t.y;
  cout<<"Gia tri cac truong\n";</pre>
  cout<< t.ten<<" "<<t.x<<" "<< t.y;</pre>
  return 0;
```



#### Ví dụ

```
Nhap thong tin toa do
Ten diem: A
Toa do x: 1
Toa do y: 3
Gia tri cac truong
A 1 3
```

```
typedef struct {
  char ten[2];
  int x, y;
}toado;
void nhap_diem(toado &t)
 cout<<"Ten diem: "; cin.getline(t.ten,2);</pre>
   cout<<"Toa do x: "; cin>>t.x;
   cout<<"Toa do y: "; cin>>t.y;
void xuat(toado t)
 cout<< t.ten<<" "<<t.x<<" "<< t.y;</pre>
int main() {
  toado t;
  cout<<"Nhap thong tin toa do\n";</pre>
  nhap_diem(t);
  cout<<"Gia tri cac truong\n";</pre>
  xuat(t);
  return 0;
```

#### Phép gán giữa các biến cấu trúc

Cho phép gán hai biến cấu trúc cùng kiểu:

Biến\_cấu\_trúc\_1 = biến\_cấu\_trúc\_2

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



#### Ví dụ

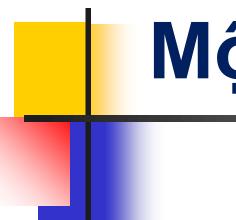
```
#include<iostream>
#include<cstring>
using namespace std;
typedef struct {
  char hoten[30];
  float diem;
} SinhVien;
int main() {
   SinhVien a, b, c;
   cout<<"Nhap thong tin sinh vien\n";</pre>
   cout<<"Ho ten: "; cin.getline(a.hoten,30);</pre>
   cout<<"Diem: "; cin>>a.diem;
```



#### Ví dụ

```
strcpy(c.hoten, a.hoten); //Gan tung truong
c.diem = a.diem;
cout<<"Bien a: "; cout<<a.hoten <<"\t"<<a.diem<<endl;</pre>
cout<<"Bien b: "; cout<<b.hoten <<"\t"<<b.diem<<endl;</pre>
cout<<"Bien c: "; cout<<c.hoten <<"\t"<<c.diem<<endl;</pre>
return 0;
```





## Một số ví dụ

- 1. Nhập vào 2 số phức và đưa ra tổng và tích của chúng
- 2. Nhập vào một danh sách (<100) sinh viên gồm họ tên, năm sinh. Kết thúc nhập khi gặp SV có tên là rỗng
  - Đưa danh sách vừa nhập ra màn hình.
  - Đưa ra màn hình sinh viên lớn tuổi nhất
- 3. Nhập danh sách có N (N < 100, nhập từ bàn phím) thí sinh gồm họ tên, số báo danh, khoa dự thi và điểm thi
  - Đưa ra DSSV đã sắp xếp theo kết quả thi
  - Đưa ra danh sách sinh viên dự thi khoa CNTT có điểm thi từ 22.5 trở lên
  - Nhập vào một số báo danh và in ra họ tên, điểm thi và khoa đăng ký của thí sinh nếu tìm thấy. Nếu không tìm thấy thí sinh thì đưa ra thông báo « không tìm thấy »



#### Ví dụ 3→Kết quả thực hiện

C:\test\test1.exe Nhap so thi sinh: 10 Nhap du lieu cho thi sinh 1: So bao danh: 1200 Ho ten: Le Tu Anh Khoa dang ky: CNTT Ket qua thi: 21.5 Nhap du lieu cho thi sinh 2: So bao danh: 1256 Ho ten: Tran Huu Cuong Khoa dang ky: DTVT Ket qua thi: 27 Nhap du lieu cho thi sinh 3: So bao danh: 1288 Ho ten: Nguyen Anh Dung Khoa dang ky: CNTT Ket qua thi: 24.5 Nhap du lieu cho thi sinh 4: So bao danh: 1297 Ho ten: Le Thu Ha Khoa dang ky: KTQL Ket qua thi: 22.5 Nhap du lieu cho thi sinh 5: So bao danh: 1322 Ho ten: Tran Hoai Nam Khoa dang ky: CNTT Ket qua thi: 22.5

C:\test\test1.exe Nhap du lieu cho thi sinh 6: So bao danh: 1345 Ho ten: Bui Trong Son Khoa dang ky: CNTT Ket qua thi: 24.5 Nhap du lieu cho thi sinh 7: So bao danh: 1350 Ho ten: Le Thanh Son Khoa dang ky: KTQL Ket qua thi: 19.5 Nhap du lieu cho thi sinh 8: So bao danh: 1355 Ho ten: Tran Thanh Son Khoa dang ky: CNTT Ket qua thi: 23 Nhap du lieu cho thi sinh 9: So bao danh: 1410 Ho ten: Nguyen Anh Tuan Khoa dang ky: DTVT Ket qua thi: 21 Nhap du lieu cho thi sinh 10: So bao danh: 1485 Ho ten: Tran Trong Viet Khoa dang ky: CNTT Ket qua thi: 21





- Lập trình đọc vào một danh sách không quá 100 sinh viên gồm: Họ tên, năm sinh
  - Đưa ra DS những sinh viên sinh năm 1990
  - Nhập tên sinh viên, cho biết năm sinh nếu tìm thấy
  - Đưa ra DSSV đã sắp xếp theo thứ tự ABC của họ và tên
- 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
  - Đọc tiếp vào một điểm chuẩn; đưa ra danh sách thí sinh trúng tuyển (không có điểm liệt - 0)
  - Đưa ra thí sinh có kết quả thi cao nhất
  - Tìm điểm chuẩn, nếu chỉ lấy K SV, K nhập vào. Nếu có nhiều người bằng điếm nhau; loại cả





#### 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
- Đưa ra DS thuốc được sắp xếp theo năm hết hạn



#### Bài tập 2

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 thích 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