



KHOA CÔNG NGHỆ THÔNG TIN

Faculty of Computer Science and Engineering-Thuyloi University

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

Chương 4: Kiểu cấu trúc và hợp

4.1. Kiểu cấu trúc

4.2. Cấu trúc tự trở và danh sách liên kết

4.2.1. Cấu trúc tự trở

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 trỏ tới thành phần

4.4. Kiểu hợp

4.5. Kiểu liệt kê

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

Để 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 rạc

```
#define MAX 1000
```

```
int      DS_SBD[MAX];
```

```
string   DS_Ten[MAX];
```

```
char     DS_KhoiThi[MAX];
```

```
float    DS_KetQua[MAX];
```

Ví dụ → Bài toán quản lý thí sinh thi đại học (tiếp)

DS_SBD

			123	456	789				
--	--	--	-----	-----	-----	--	--	--	--

DS_Ten

Mảng ký tự

			Nguyễn Văn An	Nguyễn Hoàng	Trần An Nghĩa				
--	--	--	---------------	--------------	---------------	--	--	--	--

DS_KhoiThi

			A	D	A				
--	--	--	---	---	---	--	--	--	--

DS_KetQua

			24.5	28.5	22.0				
--	--	--	------	------	------	--	--	--	--

↑ ↑ ↑
TS: TS: TS: TS: i+2

Biến điều khiển
dùng duyệt mảng

Vấn đề & giải pháp

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 danh:.....
Họ và Tên:.....
Khối thi:.....
Kết quả:.....

Mỗi phần tử của mảng là một cấu trúc thông tin



Vấn đề & giải pháp

Dùng nhiều mạng

- 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 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.

Khái niệm → Ví dụ

- Kết quả học tập của sinh viên
 - TenSV: Chuỗi ký tự
 - MaSV: Chuỗi số/ số nguyên
 - Điểm: 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_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

Khai báo kiểu cấu trúc → 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 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

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

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; //Tọa độ 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; // Tọa độ 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

Ví dụ vận dụng 1

- 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

```
Nhap so sinh vien, n = 3
Nhap danh sach sinh vien:
Nhap Ma SV: SV1
Nhap Ten SV: Hoang Mai
Nhap Diem: 23
Nhap Ma SV: SV2
Nhap Ten SV: Nguyen Anh
Nhap Diem: 26
Nhap Ma SV: SV3
Nhap Ten SV: Tran Hai
Nhap Diem: 25
Hien thi danh sach:
Ma TS      Ho Ten      Diem
SV1        Hoang Mai    23
SV2        Nguyen Anh  26
SV3        Tran Hai   25
```

Ví dụ vận dụng 1

- 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  
Nhap danh sach sinh vien:  
Nhap Ma SV: SV1  
Nhap Ten SV: Hoang Mai  
Nhap Diem: 23  
Nhap Ma SV: SV2  
Nhap Ten SV: Nguyen Anh  
Nhap Diem: 26  
Nhap Ma SV: SV3  
Nhap Ten SV: Tran Hai  
Nhap Diem: 25  
Hien thi danh sach:  
Ma TS      Ho Ten      Diem  
SV1        Hoang Mai    23  
SV2        Nguyen Anh   26  
SV3        Tran Hai     25
```

Ví dụ vận dụng 1

```
#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);
        cout<<"Nhap Ten SV: "; getline(cin,ds[i].HoTen);
        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;
}
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;
}
```

Ví dụ vận dụng 2

- 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 sắp xếp tăng dần theo điểm trúng tuyển

```
Nhap so sinh vien, n = 3
Nhap danh sach sinh vien:
Nhap Ma SV: SV1
Nhap Ten SV: Hoang Mai
Nhap Diem: 23
Nhap Ma SV: SV2
Nhap Ten SV: Nguyen Anh
Nhap Diem: 26
Nhap Ma SV: SV3
Nhap Ten SV: Tran Hai
Nhap Diem: 25
Hien thi danh sach:
Ma TS      Ho Ten      Diem
SV1        Hoang Mai    23
SV2        Nguyen Anh   26
SV3        Tran Hai    25
```

Ví dụ vận dụng 2

```
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++){
        cin.ignore();
        cout<<"Nhap Ma SV: "; getline(cin,ds[i].MaTS);
        cout<<"Nhap Ten SV: "; getline(cin,ds[i].HoTen);
        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;
    *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]);
}

int main(){
    SinhVien ds[100];
    int n;
    NhapDS(ds,n);
    sap_xep_diem(ds,n);
    HienThiDS(ds,n);
    return 0;
}
```

Khai báo biến cấu trúc → Chú ý

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;
```

Khai báo biến cấu trúc → Chú ý

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;
```


Khai báo biến cấu trúc → Chú ý

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[30];  
    Date NS;  
} SV = {"Tran Anh",20,12,1990}, x;
```

Khai báo biến cấu trúc → Chú ý

```
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>
using namespace std;
struct SinhVien{
    string MaTS;
    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;
    cout<<SV1.MaTS<<"\t"<<SV1.HoTen
    <<"\t"<<SV1.NgaySinh.day<<"/"<<
    SV1.NgaySinh.month<<"/"<<SV1.NgaySinh.year<<"\t"<<SV1
    .Diem;
    return 0;
}
```

```
In lai thong tin SV:
M01      Hoang Bach      12/5/2002      23
-----
Process exited after 0.2106 seconds with re
Press any key to continue . . .
```

Ví dụ 2

```
#include <iostream>
#include <string>
using namespace std;
struct Date {
    int day;
    int month;
    int year;
};
struct SinhVien{
    string MaTS;
    string HoTen;
    Date NgaySinh;
    float Diem;
};
int main(){
    SinhVien SV;
    cout<<"Nhap Ma SV: "; getline(cin, SV.MaTS);
    cout<<"Nhap Ten SV: "; getline(cin,SV.HoTen);
    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;
    cout<<SV.MaTS<<"\t"<<SV.HoTen <<"\t"<<SV.NgaySinh.day<<"/"<<
SV.NgaySinh.month<<"/"<<SV.NgaySinh.year<<"\t"<<SV.Diem;
    return 0;
}
```

```
Nhap Ma SV: A1
Nhap Ten SV: Hoang Long
Nhap ngay sinh SV: 23
Nhap thang sinh SV: 4
Nhap nam sinh SV: 2002
Nhap Diem: 26
In lai thong tin SV:
A1      Hoang Long      23/4/2002      26
```

Ví dụ vận dụng 3

- 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 tuyển
- 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>;
```

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 long mask;
```

```
typedef double db;
```

```
mask a, b;
```

```
db x, y;
```

Định nghĩa kiểu dữ liệu với typedef

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

Chú ý

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;
```


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

Ví dụ 3

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>
#include<cstring>
using namespace std;

typedef struct {
    char ten[2];
    int x, y;
}toado;

int main() {
    toado t;
    cout<<"Nhap thong tin toa do\n";
    cout<<"Ten diem: "; cin.getline(t.ten,2);
    cout<<"Toa do x: "; cin>>t.x;
    cout<<"Toa do y: "; cin>>t.y;
    cout<<"Gia tri cac truong\n";
    cout<< t.ten<<" "<<t.x<<" "<< t.y;
    return 0;
}
```

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

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);
    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;
}
int main() {
    toado t;
    cout<<"Nhap thong tin toa do\n";
    nhap_diem(t);
    cout<<"Gia tri cac truong\n";
    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";
    cout<<"Ho ten: "; cin.getline(a.hoten,30);
    cout<<"Diem: "; cin>>a.diem;
```

Ví dụ

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

Bài tập

1. 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 điểm nhau; loại cả

Bài tập 1

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 giáKết 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 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