



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# TIN HỌC ĐẠI CƯƠNG

Phần II: LẬP TRÌNH C

# Nội dung chính

- Chương 1: Tổng quan về ngôn ngữ C
- Chương 2: Kiểu dữ liệu và biểu thức trong C
- Chương 3: Vào ra dữ liệu
- Chương 4: Cấu trúc điều khiển
- Chương 5: Mảng, con trỏ và chuỗi ký tự
- Chương 6: Cấu trúc
- Chương 7: Hàm
- Chương 8: Tập dữ liệu

# Chương 6: Cấu trúc

## 6.1. Khái niệm cấu trúc

## 6.2. Khai báo cấu trúc

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

## 6.3. Xử lý dữ liệu cấu trúc

- Truy nhập các trường dữ liệu
- Phép gán giữa các biến cấu trúc

## 6.4. Một số ví dụ

# Chương 6: Cấu trúc

## 6.1. Khái niệm cấu trúc

## 6.2. Khai báo cấu trúc

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

## 6.3. Xử lý dữ liệu cấu trúc

- Truy nhập các trường dữ liệu
- Phép gán giữa các biến cấu trúc

## 6.4. Một số ví dụ

# 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 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

unsigned SBD;

char Ten[30];

char KhoiThi;

float KetQua;

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

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

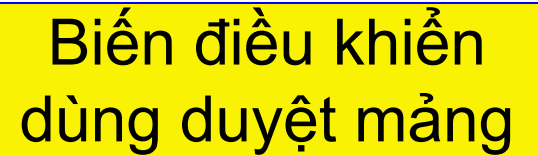
```
#define MAX 1000  
  
unsigned DS_SBD[MAX];  
  
char      DS_Ten[MAX][30];  
  
char      DS_KhoiThi[MAX];  
  
float     DS_KetQua[MAX];
```

DS\_SBD

# DS\_Ten

# DS\_KhoiThi

# DS KetQua



# 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

Mảng các cấu trúc thông tin dành cho thí sinh

Cấu trúc thông tin dành cho một thí sinh

THÔNG TIN 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



# Khái niệm

- 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

# Chương 6: Cấu trúc

## 6.1. Khái niệm cấu trúc

## 6.2. Khai báo cấu trúc

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

## 6.3. Xử lý dữ liệu cấu trúc

- Truy nhập các trường dữ liệu
- Phép gán giữa các biến cấu trúc

## 6.4. 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 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

```
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;
```

# Khai báo biến cấu trúc → 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 mặt phẳng  
}A, B; //Khai báo 2 điểm A, B
```

Kết hợp với khai báo kiểu

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



# 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];  
    struct 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];  
    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};
```

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

# 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 { //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

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

# Chương 6: Cấu trúc

## 6.1. Khái niệm cấu trúc

## 6.2. Khai báo cấu trúc

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

## 6.3. Xử lý dữ liệu cấu trúc

- Truy nhập các trường dữ liệu
- Phép gán giữa các biến cấu trúc

## 6.4. Một số ví dụ



# Truy cập các trường dữ liệu

- Cú pháp

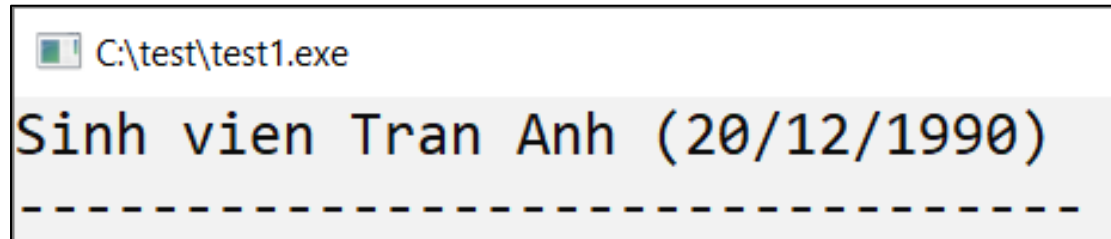
*tên\_biến\_cấu\_trúc.tên\_trường*

- 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

```
#include <stdio.h>
int 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);
    return 0;
}
```



# Ví dụ

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<stdio.h>
#include<string.h>
typedef struct {
    char ten[5];
    int x, y;
} toado;
int main() {
    toado t;
    printf("Nhap thong tin toa do\n");
    printf("Ten diem: ");
    fgets(t.ten, sizeof(t.ten), stdin);
    t.ten[strlen(t.ten) - 1] = '\0'; // Bỏ ký tự xuống dòng
    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);
    return 0;
}
```

C:\test\test1.exe

```
Nhap thong tin toa do
Ten diem: A
Toa do x: 4
Toa do y: 5
Gia tri cac truong
A      4  5
```

# Phép gán giữa các biến cấu trúc

- C 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 TĐC 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` ?

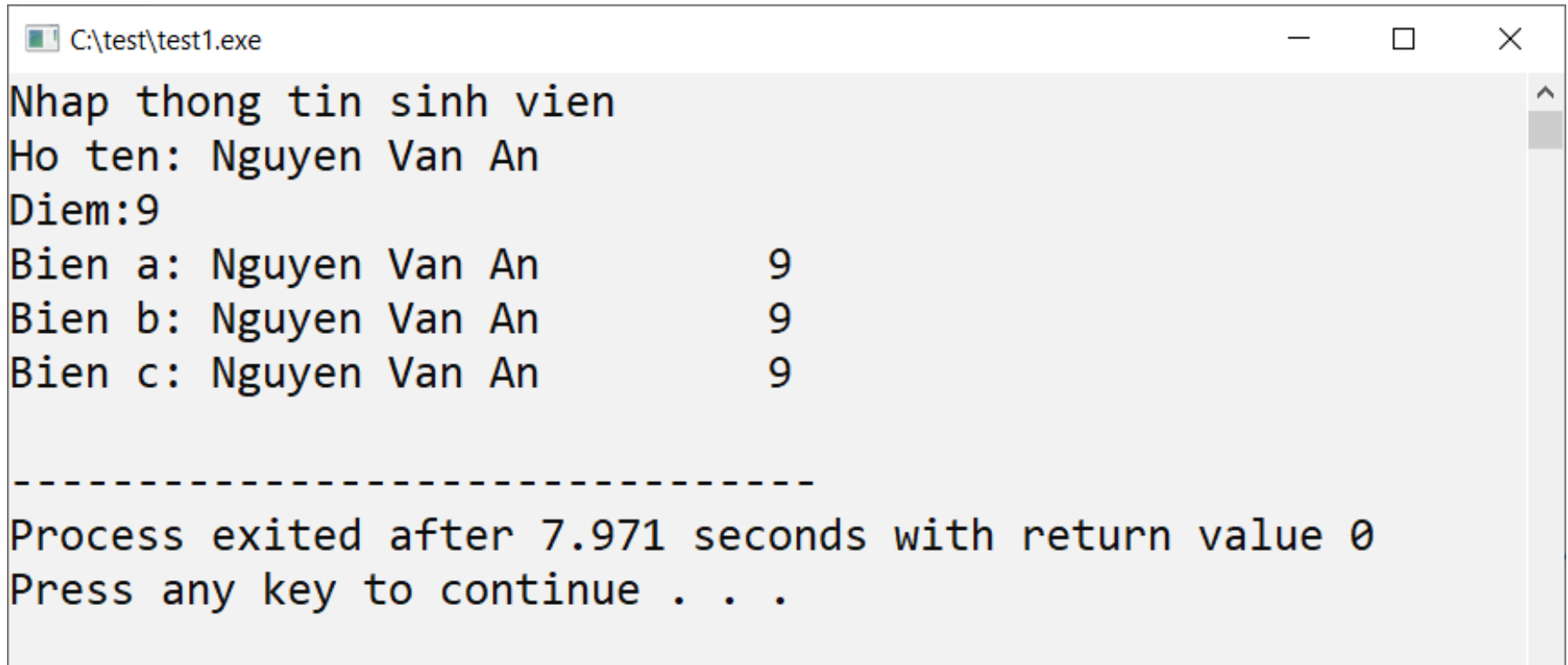
# Ví dụ

```
#include<stdio.h>
#include<string.h>
typedef struct {
    char hoten[20];
    int diem;
} sinhvien;
int main() {
    sinhvien a, b, c;
    printf("Nhap thong tin sinh vien\n");
    printf("Ho ten: ");
    fgets(a.hoten, sizeof(a.hoten), stdin);
    a.hoten[strlen(a.hoten) - 1] = '\0'; // Bỏ ký tự xuống dòng
    printf("Diem:"); scanf("%d", &a.diem);
```

# Ví dụ

```
b = a; //Gan bien cau truc
strcpy(c.hoten, a.hoten); //Gan tung truong
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);
return 0;
}
```

# Ví dụ → Kết quả



```
C:\test\test1.exe
Nhap thong tin sinh vien
Ho ten: Nguyen Van An
Diem:9
Bien a: Nguyen Van An          9
Bien b: Nguyen Van An          9
Bien c: Nguyen Van An          9

-----
Process exited after 7.971 seconds with return value 0
Press any key to continue . . .
```



# Chương 6: Cấu trúc

## 6.1. Khái niệm cấu trúc

## 6.2. Khai báo cấu trúc

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

## 6.3. Xử lý dữ liệu cấu trúc

- Truy nhập các trường dữ liệu
- Phép gán giữa các biến cấu trúc

## 6.4. Một số ví dụ

# 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 1

```
#include <stdio.h>
typedef struct {float re, im;} Complex;
int main(){
    Complex R, R1, R2;
    printf("Phan thuc & phan ao cho so thu nhat :");
    scanf("%f%f",&R1.re,&R1.im);
    printf("Phan thuc & phan ao cho so thu hai :");
    scanf("%f%f",&R2.re,&R2.im);
    R.re = R1.re+R2.re; R.im = R1.im+R2.im; // Phep cong so ao
    printf("(%.1f+%.1fi)+(%.1f+%.1fi)=(%.1f+%.1fi)\n",
           R1.re,R1.im,R2.re,R2.im,R.re,R.im);
    R.re = R1.re*R2.re - R1.im*R2.im;      // Nhan so ao
    R.im = R1.re*R2.im + R1.im*R2.re;
    printf("(%.1f+%.1fi)*(%.1f+%.1fi)=(%.1f+%.1fi)\n",
           R1.re,R1.im,R2.re,R2.im,R.re,R.im);
    return 0;
}
```

# Ví dụ 1 → Kết quả thực hiện

```
C:\test\test1.exe
Phan thuc & phan ao cho so thu nhat :2 5
Phan thuc & phan ao cho so thu hai  :3 4
(2.0+5.0i)+(3.0+4.0i)=(5.0+9.0i)
(2.0+5.0i)*(3.0+4.0i)=(-14.0+23.0i)

-----
Process exited after 3.73 seconds with return value 36
Press any key to continue . . .
```

# Ví dụ 2

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  typedef struct SinhVien {
4.      char Ten[50];
5.      int NS;
6.  } SinhVien;
7.  int main() {
8.      SinhVien SV;
9.      int n = 0;
10.     do {
11.         printf("Nhap du lieu cho sinh vien %d: \n", n+1);
12.         printf("Ho ten: "); fflush(stdin);
13.         fgets(SV.Ten, sizeof(SV.Ten), stdin);
14.         SV.Ten[strlen(SV.Ten) - 1] = '\0'; // Bỏ ký tự xuống dòng
15.         if (strlen(SV.Ten) > 0) {
16.             printf("Nam sinh: "); scanf("%d", &SV.NS);
17.             DS[n] = SV; n = n + 1;
18.         }
19.     } while (strlen(SV.Ten) > 0);
```

# Ví dụ 2

```
20.    // In danh sach sinh vien
21.    printf("\n\n");
22.    printf("                HO & TEN                NAM SINH\n");
23.    for (i = 0; i < n; i++)
24.        printf("%-3d%-30s %6d\n", i+1, DS[i].Ten, DS[i].NS);
25.    printf("\n\n");
26.    // Tim sinh vien lon tuoi nhat
27.    SV = DS[0];
28.    for (i = 0; i < n; i++)
29.        if (DS[i].NS < SV.NS)
30.            SV = DS[i];
31.    printf("Sinh vien: %s sinh nam: %d lon tuoi nhat", SV.Ten, SV.NS);
32.    return 0;
33. }//main
```

# Ví dụ 2→Kết quả thực hiện

```
C:\test\test1.exe

Nhap du lieu cho sinh vien 5:
Ho ten: Tran Hoai Thanh
Nam sinh: 1990
Nhap du lieu cho sinh vien 6:
Ho ten: Nguyen Duc Cuong
Nam sinh: 1990
Nhap du lieu cho sinh vien 7:
Ho ten:

          HO & TEN                NAM SINH
1  Nguyen Thanh Nga                1990
2  Le Anh Nam                      1991
3  Hoang Mai Huynh                 1990
4  Le Danh Tra                     1989
5  Tran Hoai Thanh                  1990
6  Nguyen Duc Cuong                 1990

Sinh vien: Le Danh Tra sinh nam: 1989 lon tuoi nhat
```

# Ví dụ 3 (1/5): Khai báo cần thiết

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  typedef struct {
4.      int SBD;
5.      char Ten[30];
6.      char Khoa[10];
7.      float Diem;
8.  } ThiSinh;
9.  int main() {
10.     ThiSinh DS[100], SV;
11.     int N, i, j, d=0, SBD;
12.     char Ch; //Sử dụng trong thực hiện tìm kiếm nhiều lần
```



# Ví dụ 3 (2/5): Nhập dữ liệu từ bàn phím

```
13.    printf("Nhap so thi sinh: "); scanf("%d", &N);
14.    for (i=0; i < N; i++) {
15.        printf("Nhap du lieu cho thi sinh  %d: \n", i+1);
16.        printf("So bao danh: "); scanf("%d", &DS[i].SBD);
17.        printf("Ho ten: ");
18.        fflush(stdin); fgets(DS[i].Ten, sizeof(DS[i].Ten), stdin);
19.        DS[i].Ten[strlen(DS[i].Ten) - 1] = '\0'; // Bỏ ký tự xuống dòng
20.        printf("Khoa dang ky: ");
21.        fflush(stdin); fgets(DS[i].Khoa, sizeof(DS[i].Khoa), stdin);
22.        DS[i].Khoa[strlen(DS[i].Khoa) - 1] = '\0'; // Bỏ ký tự xuống dòng
23.        printf("Ket qua thi: "); scanf("%f", &DS[i].Diem);
24.        printf("\n");
25.    }//for
26.    printf("\n\nNhan phim bat ky de xem ket qua thi...");
```

[C scanf: floating point format not linked](#)

# Ví dụ 3 (3/5): Danh sách sắp xếp theo điểm thi

```
27.    for (i = 0; i < N-1; i++) //Sắp xếp DS theo điểm thi
28.        for (j = i+1; j < N; j++)
29.            if (DS[i].Diem > DS[j].Diem) {
30.                SV = DS[i];
31.                DS[i] = DS[j];
32.                DS[j] = SV;
33.            }
34.    printf("\n\n                        KET QUA THI \n\n");
35.    for(i = 0; i < N; i++)
36.        printf("%-3d BKA- %-6d %-24s %-6s %-6.1f\n", ++d,
37.            DS[i].SBD, DS[i].Ten, DS[i].Khoa, DS[i].Diem);
37.    printf("\n\nNhan phim bat ky de xem DSSV khoa CNTT");
```

# Ví dụ 3 (4/5): Danh sách trúng tuyển khoa CNTT

```
38. //Dua ra danh sach du thi khoa CNTT
39. d = 0;
40. printf("\n\nDanh sach thi sinh trung tuyen khoa CNTT\n\n");
41. for (i = 0; i < N; i++)
42.     if (strcmp(DS[i].Khoa,"CNTT")==0 && DS[i].Diem >=22.5)
43.         printf("%-3d BKA%-6d %-24s %-f\n", ++d, DS[i].SBD,
                    DS[i].Ten, DS[i].Diem);
44. printf("\n\nNhan phim bat ky de bat dau timkiem theo so bao
    danh");
```

# Ví dụ 3 (5/5): Tìm kiếm sinh viên

```
45.     do {
46.         printf("\nNhap so bao danh can tim: ");
47.         scanf("%d", &SBD);
48.         for (i = 0; i < N; i++)
49.             if (DS[i].SBD == SBD) {
50.                 printf("So bao danh : %d \n", SBD);
51.                 printf("Ho ten      : %s \n", DS[i].Ten);
52.                 printf("Khoa du thi : %s \n", DS[i].Khoa);
53.                 printf("So bao danh : %.1f \n", DS[i].Diem);
54.                 break;
55.             }
56.         if (i == N)
57.             printf("So bao danh %d khong ton tai\n", SBD);
58.         printf("\nCo tiep tục tìm kiếm nữa không (C/K): ");
59.     } while(toupper(getche()) != 'K');
60.     return 0;
61. }//main
```

# 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
```

# Ví dụ 3→Kết quả thực hiện

```
C:\test\test1.exe

Nhan phim bat ky de xem ket qua thi...

                        KET QUA THI

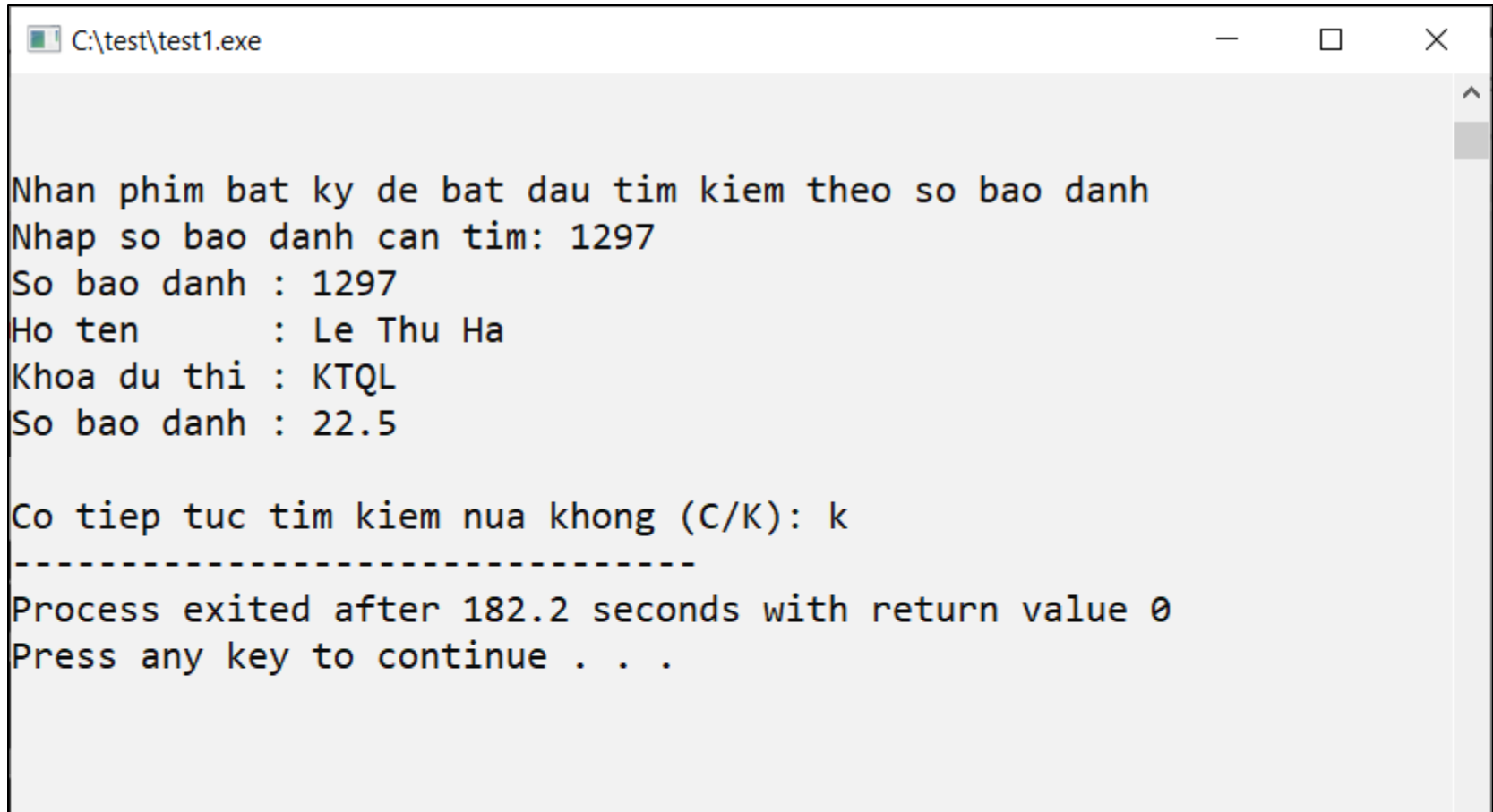
1   BKA- 1350   Le Thanh Son           KTQL   19.5
2   BKA- 1410   Nguyen Anh Tuan        DTVT   21.0
3   BKA- 1485   Tran Trong Viet        CNTT   21.0
4   BKA- 1200   Le Tu Anh              CNTT   21.5
5   BKA- 1322   Tran Hoai Nam           CNTT   22.5
6   BKA- 1297   Le Thu Ha              KTQL   22.5
7   BKA- 1355   Tran Thanh Son          CNTT   23.0
8   BKA- 1288   Nguyen Anh Dung        CNTT   24.5
9   BKA- 1345   Bui Trong Son            CNTT   24.5
10  BKA- 1256   Tran Huu Cuong            DTVT   27.0

Nhan phim bat ky de xem DSSV khoa CNTT

Danh sach thi sinh trung tuyen khoa CNTT

1   BKA1322   Tran Hoai Nam           22.500000
2   BKA1355   Tran Thanh Son          23.000000
3   BKA1288   Nguyen Anh Dung        24.500000
4   BKA1345   Bui Trong Son          24.500000
```

# Ví dụ 3→Kết quả thực hiện



```
C:\test\test1.exe

Nhan phim bat ky de bat dau tim kiem theo so bao danh
Nhap so bao danh can tim: 1297
So bao danh : 1297
Ho ten      : Le Thu Ha
Khoa du thi : KTQL
So bao danh : 22.5

Co tiep tục tìm kiếm nữa không (C/K): k
-----
Process exited after 182.2 seconds with return value 0
Press any key to continue . . .
```

# 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

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  typedef struct {
4.      char Ten[30];
5.      int NS;
6.  } SinhVien;
7.  int main() {
8.      SinhVien DS[100], SV;
9.      int N, i, j, d=0;
10.     char ten[30];
11.     printf("Nhap so sinh vien : "); scanf("%d", &N);
12.     for ( i=0; i < N; i++ ){
13.         printf("Nhap du lieu cho sinh vien %d: \n", i+1);
14.         printf("Ho ten : ");
15.         fflush(stdin); fgets(DS[i].Ten, sizeof(DS[i].Ten), stdin);
16.         DS[i].Ten[strlen(DS[i].Ten) - 1] = '\0'; // Bỏ ký tự xuống dòng
17.         printf("Nam sinh : "); scanf("%d", &DS[i].NS);
18.     }
```

# Bài tập 1 (tiếp)

```
19. printf("\n\nSINH VIEN SINH NAM 1990\n\n");
20. for (i = 0; i < N; i++)
21.     if (DS[i].NS == 1990)
22.         printf("%s\n", DS[i].Ten);
23. printf("\n\nTim SV : ");
24. fflush(stdin); fgets(ten, sizeof(ten), stdin);
25. ten[strlen(ten) - 1] = '\0'; // Bỏ ký tự xuống dòng
26. for (i = 0; i < N; i++)
27.     if (strcmp(Ten, DS[i].Ten) == 0) {
28.         printf("Sinh vien: %s\nSinh nam %d\n", DS[i].Ten,
                DS[i].NS);
29.         d = d + 1;
30.     }
31. if (d == 0)
32.     printf("Khong co sinh vien: %s trong danh sach\n",ten);
```

**Tìm kiếm chỉ theo tên ?**

`strcmp(ten, strchr(DS[i].Ten, 32)+1))`

# Bài tập 1 (tiếp)

```
33.     for (i = 0; i < N-1; i++) //sắp xếp theo pp lựa chọn
34.         for (j = i+1; j < N; j++)
35.             if (strcmp(DS[i].Ten, DS[j].Ten) > 0) {
36.                 SV = DS[i];
37.                 DS[i] = DS[j];
38.                 DS[j] = SV;
39.             }
40.     printf("\n\n DANH SACH SAP XEP\n\n");
41.     for(i = 0; i < N; i++)
42.         printf("%d %-20s %d \n",i+1, DS[i].Ten, DS[i].NS);
43.     return 0;
44. } //main
```

# Bài tập 1 → Kết quả thực hiện

```
C:\test\test1.exe
Nhap so sinh vien : 8
Nhap du lieu cho sinh vien 1:
Ho ten : Nguyen Tuan Anh
Nam sinh : 1992
Nhap du lieu cho sinh vien 2:
Ho ten : Do Trong Khang
Nam sinh : 1990
Nhap du lieu cho sinh vien 3:
Ho ten : Bui Hai Thanh
Nam sinh : 1991
Nhap du lieu cho sinh vien 4:
Ho ten : Hoang Tuan Anh
Nam sinh : 1992
Nhap du lieu cho sinh vien 5:
Ho ten : Le Thu Ha
Nam sinh : 1990
Nhap du lieu cho sinh vien 6:
Ho ten : Bui Thanh Huong
Nam sinh : 1991
Nhap du lieu cho sinh vien 7:
Ho ten : Tran Thanh Son
Nam sinh : 1990
Nhap du lieu cho sinh vien 8:
Ho ten : Nguyen Anh Dung
Nam sinh : 1992
```

```
C:\test\test1.exe
SINH VIEN SINH NAM 1990
Do Trong Khang
Le Thu Ha
Tran Thanh Son

Tim SV : Hoang Tuan Anh
Sinh vien: Hoang Tuan Anh
Sinh nam 1992

DANH SACH SAP XEP
1 Bui Hai Thanh          1991
2 Bui Thanh Huong       1991
3 Do Trong Khang        1990
4 Hoang Tuan Anh        1992
5 Le Thu Ha             1990
6 Nguyen Anh Dung       1992
7 Nguyen Tuan Anh       1992
8 Tran Thanh Son        1990
```

# Bài tập 2 (1/5)

```
1. #include <stdio.h>
2. #include <string.h>
3.
4. typedef struct {
5.     char Ten[30];
6.     struct {
7.         int T, L, H, S; // S = T + L + H
8.     } DT;
9. } SinhVien;
10.
11. int main() {
12.     SinhVien DS[100], TK, SV;
13.     int N, i, j, K;
14.     float C;
```

# Bài tập 2 (2/5)

```
15.    N = 0; // N chứa số sinh viên đã nhập
16.    do {
17.        printf("\nNhap DL cho sv thu %d\n",N+1);
18.        printf("Ten SV : ");
19.        fflush(stdin);
20.        fgets(DS[N].Ten, sizeof(DS[N].Ten), stdin);
21.        DS[N].Ten[strlen(DS[N].Ten) - 1] = '\0'; // Bỏ ký tự xuống dòng
22.        if(strlen(DS[N].Ten) == 0) //Độ dài bằng 0 ⇒ xâu rỗng
23.            break;
24.        else {
25.            printf("Nhap điểm thi T L H của SV %s : ", DS[N].Ten);
26.            scanf("%d%d%d", &DS[N].DT.T, &DS[N].DT.L, &DS[N].DT.H);
27.            DS[N].DT.S = DS[N].DT.T + DS[N].DT.L + DS[N].DT.H;
28.            N++;
29.        }
30.    } while(1);
```

# Bài tập 2 (3/5)

```
31. // In ra danh sách vừa nhập
32. printf("\n\n DANH SACH SINH VIEN\n\n");
33. printf("          Ten SV          Toan  Ly   Hoa Tong \n");
34. for(i = 0; i < N; i++)
35.     printf("%-20s%5d%5d%5d%6d\n", DS[i].Ten, DS[i].DT.T,
            DS[i].DT.L, DS[i].DT.H, DS[i].DT.S);
36. // In danh sách trúng tuyển theo điểm chuẩn
37. printf("\n\nDiem Chuan : "); scanf("%f", &C);
38. printf("\n\n DANH SACH SINH VIEN TRUNG TUYEN \n\n");
39. for(i = 0; i < N; i++)
40.     if((DS[i].DT.S >= C) && (DS[i].DT.T * DS[i].DT.L *
            DS[i].DT.H > 0))
41.         printf("%s\n", DS[i].Ten);
```

# Bài tập 2 (4/5)

```
42. // Tìm điểm của thủ khoa, bỏ qua trường hợp điểm liệt
43. TK = DS[0];
44. for (i = 1; i < N; i++)
45.     if (DS[i].DT.S > TK.DT.S)
46.         TK = DS[i];
47. // Đưa ra danh sách thí sinh có điểm bằng điểm cao nhất
48. for(i = 0; i < N; i++)
49.     if(DS[i].DT.S == TK.DT.S)
50.         printf("\n\n THU KHOA: %s \n\n", TK.Ten);
```



# Bài tập 2 (5/5)

```
51.    printf("\nSo nguoi trung tuyen:"); scanf("%d", &K);
52.    for (i = 0; i < N - 1; i++) //Sắp xếp theo tổng điểm thi
53.        for (j = i+1; j < N; j++)
54.            if (DS[i].DT.S < DS[j].DT.S ) {
55.                SV = DS[i];
56.                DS[i] = DS[j];
57.                DS[j] = SV;
58.            }
59.    while ((K>0) && (DS[K-1].DT.S == DS[K].DT.S)) K--;
60.    if (K>0) {
61.        printf("Diem Chuan La : %4d", DS[K-1].DT.S);
62.        printf("\n\n Danh Sach sinh vien trung tuyen \n");
63.        for (i=0; i < K; i++)
64.            printf("%s\n", DS[i].Ten);
65.    }
66.    return 0;
67. } //main
```

# 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 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 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

# 1. Kết quả đưa ra màn hình

```
#include<stdio.h>
typedef struct {
    int SHSV;
    char Ten[25];
} SV;
int main() {
    SV DS[] = {{12, "Mai"}, {13,
        "Nam"}, {14, "Minh"}};
    printf("%d ", DS[1].SHSV);
    printf("%s\n", (*(DS+2)).Ten);
    return 0;
}
```

a	12 Mai
b	12 Nam
c	13 Nam
d	13 Minh
e	14 Minh

## 2. Chỉ ra câu trả lời đúng

Chỉ ra khai báo hợp lệ cho biến **SV** có kiểu cấu trúc chỉ gồm 2 trường: **Tên** có kiểu xâu ký tự và **NS** có kiểu số nguyên

a	<pre>struct SinhVien{     char Ten[20];     unsigned NS; }; SinhVien SV;</pre>	b	<pre>struct {     char Ten[20];     unsigned NS; } SinhVien; struct SinhVien SV;</pre>
c	<pre>struct {     unsigned NS;     char Ten[20]; } SV;</pre>	d	<pre>typedef struct {     unsigned NS;     char Ten[20]; } SV;</pre>
e	Không có câu trả lời nào đúng		