

<b>Trường ĐH CNTP TP.HCM</b> <b>Khoa: CNTT</b> <b>Bộ môn: CNPM</b> <b>Môn: TH NMLT</b>	<b>BÀI 9. MẢNG CẤU TRÚC</b>	
---	-----------------------------	--

### A. MỤC TIÊU:

- Định nghĩa được kiểu cấu trúc trong bài toán cụ thể
- Khai báo được biến kiểu mảng cấu trúc.
- Phân tích được tình huống sử dụng mảng cấu trúc để lưu trữ dữ liệu.
- Cài đặt được hàm nhập, xuất, xử lý dữ liệu bằng cách sử dụng mảng cấu trúc
- Vận dụng được kiểu mảng cấu trúc để giải quyết bài toán cụ thể

### B. DỤNG CỤ - THIẾT BỊ THÍ NGHIỆM CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	– Máy tính (có thể là máy bàn hoặc laptop).	1	cái	

### C. VẬT LIỆU

- Visual C++.Net.
- Phần mềm hỗ trợ dạy học: NetOp School hoặc NetSupport School.

### D. NỘI DUNG THỰC HÀNH

#### I. Tóm tắt lý thuyết

##### 1. Khai báo mảng struct:

Khai báo mảng các cấu trúc hoàn toàn tương tự như khai báo mảng khác, chỉ có điều ở phần kiểu sẽ là tên một cấu trúc. W

**<Tên cấu trúc> <Tên mảng>[<kích thước mảng>;**

##### Ví dụ:

```
struct SinhVien
{
    char mssv[50];
    char hoten[50];
    int solanvang;
};
SinhVien a[10];
```

##### 2. Truy cập thành phần của mảng cấu trúc:

Việc truy xuất đến một phần tử của mảng như a[2] ta sẽ thu được một biến có cấu trúc SinhVien và có thể lại tiếp tục truy xuất đến các thành phần của nó.

Ví dụ:

```
printf("\n Nhập ma so sinh vien:");
gets(a[2].mssv);
printf("\n Nhập ho ten:");
gets(a[2].hoten);
printf("\n Nhập so lan vang:");
scanf("%d", &a[2].solanvang);
```

## II. Bài tập mẫu hướng dẫn thực hành.

- Khai báo cấu trúc CongNhan lưu trữ thông tin: Họ tên, năm sinh và số giờ làm.

### *Yêu cầu:*

- Nhập mảng gồm n công nhân, xuất mảng công nhân vừa nhập
- Liệt kê tên công nhân nhận thưởng biết rằng công nhân nhận thưởng là các công nhân có giờ làm lớn hơn 40 giờ.
- Sắp xếp mảng công nhân theo giờ làm giảm dần.

### *Hướng dẫn:*

- Khai báo thêm các thư viện cơ bản cho chương trình.

```
#include<stdio.h>
#include<conio.h>
#include <math.h>
```

- Khai báo struct CongNhan cho chương trình.

```
struct CongNhan
{
    char ten[50];
    int namsinh;
    int giolam;
};
```

- Viết các định nghĩa hàm cần thiết cho chương trình như sau:

```
void NhapMangCN(int &n, CongNhan a[])
{
    do
    {
        printf("Cho biet so cong nhan: ");
        scanf_s("%d", &n);
    }while(n<=0);
    for(int i=0; i<n; i++)
    {
        printf("Thong tin cong nhan thu %d la:",i+1);
        printf("\nTen la:");
        fflush(stdin);
```

```

        gets_s(a[i].ten,100);
        printf("Nam Sinh:");
        scanf_s("%d", &a[i].namsinh);
        printf("Gio lam:");
        scanf_s("%d", &a[i].giolam);

    }
}
//=====
void XuatMangCN(CongNhan a[], int n)
{
    for(int i=0; i<n; i++)
    {
        printf("\n=====");
        printf("\nThong tin cong nhan thu %d la:", i+1);
        printf("\nTen la: %s",a[i].ten);
        printf("\nNam Sinh la %d:",a[i].namsinh);
        printf("\nGio lam %d:",a[i].giolam);
    }
}
//=====
void XuatCN(CongNhan x,int i)
{
    printf("\nThong tin cong nhan nhan thuong la:", i+1);
    printf("\nTen la: %s",x.ten);
    printf("\nNam Sinh la %d:",x.namsinh);
    printf("\nGio lam %d:",x.giolam);
}
//=====
void ThuongCN(CongNhan a[], int n)
{
    printf("\n=====");
    printf("\nDanh sach cong nhan nhan thuong:");
    for(int i=0; i<n; i++)
    {
        if(a[i].giolam>40)
        {
            XuatCN(a[i],i);
        }
    }
}
}

```

```

//=====
void InsertionSort(CongNhan a[], int n)
{
    for(int i=1; i<n; i++)
    {
        CongNhan x=a[i];
        for(int j=i-1; (j>=0 && a[j].giolam < x.giolam); j--)
        {
            a[j+1]=a[j];
            a[j]=x;
        }
    }
}
//=====
void SapXep(CongNhan a[], int n)
{
    InsertionSort(a,n);
}

```

- Viết hàm main để thực thi chương trình.

```

void main()
{
    CongNhan A[1000]; // Mảng struct công nhân
    int N; // N công nhân biến kiểu nguyên
    NhapMangCN(N,A);
    ThuongCN(A,N);
    SapXep(A,N);
    XuatMangCN(A,N);
    getch();
}

```

### III. Bài tập thực hành ở lớp

#### 1. Khai báo cấu trúc Thí Sinh lưu trữ thông tin:

- Mã số thí sinh
- Tên thí sinh
- Điểm Toán, Điểm Lý, Điểm Hóa
- Khu vực dự thi
- Xếp loại tốt nghiệp

Sau đó thực hiện các chức năng:

- Nhập mảng gồm n thí sinh, xuất mảng thí sinh vừa nhập

- Nhập vào vị trí 1 thí sinh, tính điểm cộng của thí sinh vừa nhập. Điểm cộng: Nếu xếp loại đạt giỏi (>8) thì cộng 2 điểm, nếu đạt loại khá (từ 7-8) thì cộng 1 điểm, các trường hợp khác không cộng.
- Nhập vào vị trí 1 thí sinh, tính điểm ưu tiên của thí sinh vừa nhập. Điểm ưu tiên: nếu là khu vực 3 được 2 điểm, nếu là khu vực 2 thì cộng 1 điểm, các khu vực khác là không.
- Nhập vào vị trí của 1 thí sinh, tính tổng điểm của thí sinh vừa nhập. Tổng điểm = điểm toán + điểm lý + điểm hóa + điểm cộng + điểm ưu tiên.
- In danh sách thí sinh (tên, điểm toán, điểm lý, điểm hóa, điểm cộng, điểm ưu tiên, tổng điểm)
- In danh sách thí sinh đậu (tổng điểm lớn hơn hoặc bằng 15).
- Nhập mã số thí sinh, xuất thông tin thí sinh (tên, điểm toán, điểm lý, điểm hóa, điểm cộng, điểm ưu tiên, tổng điểm).
- Tìm điểm toán cao nhất, sau đó tìm thí sinh có điểm toán cao nhất
- In thông tin thí sinh đậu thủ khoa
- Tính tỉ lệ % đậu của kì thi.

Gợi ý:

```
struct ThiSinh
{
    char mats[10];
    char hten[50];
    float toan, ly, hoa;
    char kvthi[30];
    char loaiTN [10];
};
```

2. Tạo file txt lưu thông tin N nhân viên theo định dạng sau:

- Dòng đầu tiên lưu số lượng nhân viên N
- 7xN dòng tiếp theo lưu mã số (không có hai người trùng mã số), họ, tên, ngày sinh, nơi sinh, ngày công tác, lương. Mỗi thông tin được nằm trên 1 dòng.

Ví dụ: dữ liệu file text txt

```
3
001
Nguyen
Van
23/07/1990
Kiên Giang
10/03/2008
3000
```

002

Tran Van

Hau

5/10/1988

Ha Tien

14/03/2005

5000

004

Truong

Hung

10/4/1987

Nam Dinh

03/07/2005

4000

Sau đó viết hàm thực hiện các chức năng sau:

- Đọc mảng nhân viên bằng cách đọc dữ liệu từ file.
- Xuất mảng nhân viên.
- Tìm nhân viên theo mã số.
- Tìm một nhân viên theo tên.
- Thêm 1 nhân viên vào cuối danh sách.
- Xóa 1 nhân viên theo mã số nhân viên.
- Tìm thông tin nhân viên có năm công tác là NAM (NAM là giá trị nhập từ phím).

Ví dụ, nhập NAM = 2005, in thông tin 2 nhân viên sau:

002 Tran Van Hau 5/10/1988 Ha Tien 14/03/2005 5000

004 Truong Hung 10/4/1987 Nam Dinh 03/07/2005 4000

#### **IV. Bài tập về nhà:**

1. Tạo file txt lưu thông tin N nhân viên theo định dạng sau:

- Dòng đầu tiên lưu số lượng nhân viên N
- 4xN dòng tiếp theo lưu Mã số nhân viên, Tên nhân viên, Hệ số lương, Chức vụ. Mỗi thông tin được nằm trên 1 dòng

Ví dụ:

3

001

Nguyen Hoang Oanh

2.34

Nhan vien

003

Nguyen Thanh Binh

3.0

Truong phong

002

Tran Bao Tien

4

Giam doc

Sau đó viết hàm thực hiện các chức năng sau:

- Nhập mảng gồm n nhân viên bằng cách đọc dữ liệu từ file.
- Xuất mảng nhân viên.
- In danh sách lương thực lãnh của nhân viên gồm các thông tin: mã nhân viên, tên nhân viên, thực lãnh. Trong đó,  
 $thực\ lãnh = hệ\ số\ lương * lương\ cơ\ bản + phụ\ cấp\ chức\ vụ$ , với  
+ lương cơ bản là một hằng số có giá trị 1,210 VNĐ.  
+ phụ cấp chức vụ: phụ cấp 500 VNĐ cho giám đốc và 300VNĐ cho trưởng phòng.
- Tính tổng lương của tất cả nhân viên
- Tìm lương thực lãnh cao nhất
- Tìm thông tin nhân viên có lương thực lãnh cao nhất.
- In ra bảng lương danh sách nhân viên theo thứ tự mã số nhân viên giảm dần

2. Tạo file txt lưu thông tin N môn học theo định dạng sau:

- Dòng đầu tiên lưu số lượng môn N
- N dòng tiếp theo lưu Mã số môn, Tên môn, Số tiết, Hình thức. Mỗi môn được nằm trên 1 dòng, mỗi thông tin ngăn cách nhau bởi dấu #.

Ví dụ:

4

L1 # Ngon ngu lap trinh # 30 # Ly thuyet

L2 # Thuc hanh Cau truc du lieu # 60 # Thuc hanh

L3 # Huong doi tuong # 45 # Ly thuyet

L5 # Toan Cao cap # 45 # Ly thuyet

Sau đó viết hàm thực hiện các chức năng sau:

- Nhập mảng gồm N môn học bằng cách đọc dữ liệu từ file.
- Xuất mảng môn học.
- In thông tin môn học có thông tin không đúng (môn học lý thuyết là bội của 15 và môn học thực hành là bội của 30)

- In danh sách số tín chỉ theo từng môn học, biết rằng số tín chỉ = số tiết / 15 (đối với môn học lý thuyết) , và số tín chỉ = số tiết / 30 (đối với môn học thực hành). Ghi chú, không in môn học không đúng thông tin.
- Hiệu chỉnh thông tin môn học cho phù hợp (chỉ cho phép hiệu chỉnh số tiết và hình thức).
- Xóa môn học khi nhập vào tên môn học.
- Thêm 1 môn học mới vào cuối danh sách môn học.