



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

# C PROGRAMMING INTRODUCTION

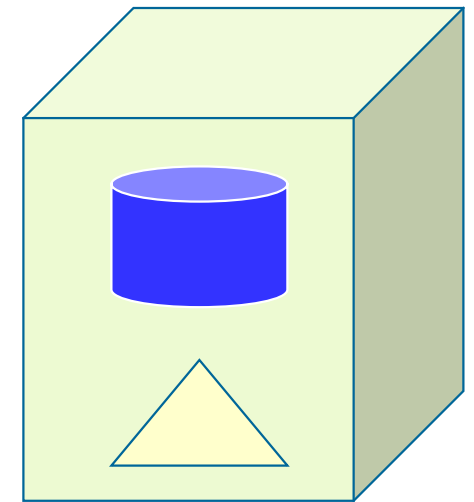
## TUẦN 14: CẤU TRÚC

# Nội dung

- Cấu trúc
  - Ôn tập
    - Khai báo cấu trúc
    - Sử dụng typedef
    - Truy cập vào các trường
  - Thực hành

# Cấu trúc

- Cấu trúc trong C là một tập hợp các phần tử có kiểu khác nhau.



# Định nghĩa **struct** trong C

```
struct struct-name  
{  
    field-type1 field-name1;  
    field-type2 field-name2;  
    field-type3 field-name3;  
    . . .  
};
```

# VD

- Định nghĩa struct sinh viên có các thông tin tên, tuổi, và điểm

```
struct student {  
    char name[20];  
    int age;  
    float grade;  
};
```

# VD

- Định nghĩa cấu trúc car chứa các thông tin tên, model, và năm sản xuất

```
struct car {  
    char* make;  
    char* model;  
    int   year;  
};
```

# Khai báo và khởi tạo biến

- Sử dụng từ khóa struct khi khai báo biến

```
struct student s1;
```

```
struct car mycar;
```

```
struct student s1 = (“Nguyen Le”, 19, 8.0);
```

```
struct car mycar = (“Fiat”, "Punto", 2004);
```

# Định nghĩa cấu trúc với typedef

```
typedef struct student {  
    char name[20];  
    int age;  
    float grade;  
} student_t;
```

```
typedef struct car {  
    char* make;  
    char* model;  
    int year;  
} car_t;
```

Chương trình có các kiểu mới **student\_t** và **car\_t**



# Khai báo biến

- Với typedef, không cần sử dụng từ khóa struct khi khai báo biến

- VD:

```
car_t mycar;
```

```
student_t excellentP;
```

# Truy cập vào thành viên của struct

- Sử dụng toán tử ‘.’

```
car_t mycar;  
mycar.year = 2004;
```

```
student_t excellentp;  
excellentp.age = 18;  
excellentp.grade = 7.8;
```

# Bài tập 14.1

- a) Tạo ra cấu trúc Date chứa các trường ngày, tháng, năm
- b) Viết hàm nhập dữ liệu cho cấu trúc có kiểm tra tính hợp lệ của dữ liệu
- c) Viết hàm datecmp để so sánh hai ngày date\_1 và date\_2
  - 1 nếu date\_1 < date\_2
  - 0 nếu date\_1 = date\_2
  - 1 nếu date\_1 > date\_2
- d) Viết chương trình yêu cầu người dùng nhập vào hai ngày và in ra kết quả so sánh

VD: *2/10/1997 > 23/8/1997*

# Lời giải

```
#include <stdio.h>
typedef struct date
{
    unsigned char day;
    unsigned char month;
    int nam;
}date_t;
```

# Lời giải

```
date input_date() {
    date tmp;
    do {
        printf("The day (between 1 and 31):");
        scanf("%u", &tmp.day);
    } while ((tmp.day < 1) || (tmp.day > 31))
    do {
        printf("The month (between 1 and 12):");
        scanf("%u", &tmp.month);
    } while ((tmp.day < 1) || (tmp.day > 12))
    do {
        printf("The year (between 1 and 10000):");
        scanf("%d", &tmp.year);
    } while ((tmp.day < 1) || (tmp.day > 1000))

    return tmp;
}
```

# Lời giải

```
int datecmp(date d1, date d2) {  
    if (d1.year < d2.year) return -1;  
    else if (d1.year > d2.year) return 1;  
    else {  
        if (d1.month < d2.month) return -1;  
        else if (d1.month > d2.month) return 1;  
        else {  
            if (d1.day < d2.day) return -1;  
            else if (d1.day > d2.day) return 1;  
            else return 0;  
        }  
    }  
}
```

# Lời giải

```
int main() {
    date date1, date2;
    int m;
    printf("Enter the first date.\n");
    date1= input_date();
    printf("Enter the second date.\n");
    date2= input_date();
    m = datecmp(date1, date2);
    if (m==0) printf("Two date are identical.\n");
    else if (m<0) printf("%d/%d/%d is before %d/%d/%d\n",
        date1.day, date1.month, date1.year, date2.day,
        date2.month, date2.year);
    else printf("%d/%d/%d is after %d/%d/%d\n", date1.day,
        date1.month, date1.year, date2.day, date2.month,
        date2.year);
    return 0;
}
```

# Bài tập 14.2

- Viết chương trình sử dụng cấu trúc để lưu thông tin thời tiết trong tháng
  - *Tổng lượng mưa*
  - *Nhiệt độ cao nhất*
  - *Nhiệt độ thấp nhất*
  - *Nhiệt độ trung bình*
- Chương trình sử dụng mảng 12 phần tử để lưu thông tin thời tiết các tháng trong năm. Chương trình yêu cầu người dùng nhập thông tin từng tháng, tính toán và in ra lượng mưa trung bình, tổng lượng mưa, nhiệt độ cao nhất và nhiệt độ thấp nhất trong năm.
- Kiểm tra dữ liệu: Nhiệt độ nằm trong khoảng -40 đến 50 độ Celcius.



# Lời giải

```
#include <stdio.h>
typedef struct wt{
    int total_rain; // in mm
    int high_temp; // in celcius
    int low_temp;
}weather;
const int MONTHS = 12;
```

# Lời giải

```
int main() {
    weather data[MONTHS];
    int i;
    char months[MONTHS][10]={"January", "February",
    "March", "April"};
    int total_rain = 0;
    int max_temp = -100;
    int max_month = 0;
    int min_temp = 100;
    int min_month = 0;
    // read input data for each month
    for (i = 0; i < MONTHS; i++) {
        printf("Weather Data for %d: \n", months[i]);
        printf("    Total Rainfall (mm): ");
        scanf("%d",&data[i].total_rain);
        printf("    High Temperature (C): ");
        scanf("%d",&data[i].high_temp);
        printf("    Low Temperature (C): ");
        scanf("%d",&data[i].low_temp);
    }
```

# Lời giải

```
// find min, max temperature
for (i = 0; i < MONTHS; i++) {
    total_rain += data[i].total_rain;
    if (max_temp < data[i].high_temp) {
        max_temp = data[i].high_temp;
        max_month = i;
    }
    if (min_temp > data[i].low_temp) {
        min_temp = data[i].low_temp;
        min_month = i;
    }
}

printf ("\nWeather Statistic for the Year:\n");
printf ("    Total Rainfall: %d mm\n", total_rain);
printf (" Monthly Rainfall: %2.3f mm\n",
double(total_rain)/MONTHS );
printf (" Highest Temperature: " : %d °C in %s \n",
max_temp, months[max_month]);
printf (" Lowest Temperature: " : %d °C in %s \n",
min_temp, months[min_month]);
return 0;
```

# Bài tập 14.3

- Viết chương trình quản lý sinh viên sử dụng cấu trúc sau

```
typedef struct
{
    char id[6];
    char name[31];
    float grade;
    char classement
} student;
```

Sinh viên được xếp loại dựa trên điểm:

- 9 - 10: A (Excellent)
- 8 - 9: B (Good)
- 6.5 - 8: C (Medium)
- < 6.5 : D (Bad)

# Bài tập 14.3 (tiếp)

- Chương trình yêu cầu người dùng nhập thông tin cho n sinh viên từ bàn phím và in ra danh sách sinh viên theo điểm giảm dần

Name	Grade	Classment
Dao Tiem	9.3	A
Dinh Lan	8.2	B
Bui Luu Van	5.7	D

# Lời giải

```
#include <stdio.h>
#include <string.h>
#define MAX 100
typedef struct {
    char id[6];
    char name[31];
    float grade;
    char classement
} student;

void printStudent(student s)
{
    printf("ID | Name      | Grade | Classment\n");
    printf("%s | %s    | %1.1f | %c\n",s.id,s.name,
        s.grade,s.classement);
}
```

# Lời giải

```
int main(){
    int i, n;
    student std_list[MAX], tmp;
    printf("Enter the number of student (>0):");
    scanf("%d", &n);
    for(i=0; i<n; i++){
        printf("ID:"); gets(std[i].id);
        printf("name:"); gets(std[i].name);
        printf("Grade:"); scanf("%f",&std[i].grade);
        if (std[i].grade >= 9 && std[i].grade <= 10)
            std[i].classment = 'A';
        else if (std[i].grade >= 8 && std[i].grade < 9)
            std[i].classment = 'B';
        else if (std[i].grade >= 6.5 && std[i].grade < 8)
            std[i].classment = 'C';
        else std[i].classment = 'D';
    }
```

# Lời giải

```
for(i=0; i<n-1; i++)
    for (j=i+1; j<n; j++)
        if (std[i].grade < std[j].grade) {
            tmp=std[i];
            std[i]=std[j];
            std[j]=tmp;
        }
for(i=0; i<n; i++) printStudent(std[i]);
return 0;
}
```



# Bài tập 14.4

- Định nghĩa cấu trúc để biểu diễn phân số. Viết chương trình xử lý phân số.
  - Nhập một mảng phân số
  - In ra nội dung của mảng
  - Đảo ngược phân số
  - So sánh hai phân số

# Lời giải

```
#include <stdio.h>
#define MAX 100
typedef struct {
    int numerator;
    int Denominator;
}fraction;

void fractionInput(fraction *ps);
void fractionOutput (fraction ps);
void fractionArrayInput(fraction dsps[], int n);
void fractionArrayOutput(fraction dsps[], int n);
int fractionCmp(fraction ps1, fraction ps2);
fraction inverse(fraction ps);
void inverseArray(fraction dsps[], int n);
```

# Lời giải

```
void fractionInput(fraction *ps){
    int n,d;
    printf("Numerator:"); scanf("%d",&n);
    ps->numerator = n;
    do {
        printf("Denominator:"); scanf("%d",&d);
    } while (d==0);
    ps->denominator = d;
}

void fractionArrayInput(fraction dsps[], int n){
    int i;
    for (i=0; i<n; i++){
        printf("Data input for the i-th fraction:\n");
        fractionInput(dsps[i]);
    }
}
```

# Lời giải

```
void fractionOutput(fraction ps){
    printf(" %d/%d ", ps.numerator,
ps.denominator);
}

void fractionArrayOutput(fraction dsps[], int
n) {
    int i;
    printf("Data output for the array of
fraction:\n");
    for (i=0; i<n; i++){
        fractionOutput(dsps[i]);
    }
    printf("\n");
}
```

# Lời giải

```
fraction inverse(fraction ps){
    fraction tmp;
    if (ps.numerator ==0) {
        printf("Can not have an inverse fraction of 0!\n");
        exit(1);
    }
    tmp.numerator = ps.denominator;
    tmp.denominator=ps.numerator;
    return tmp;
}

int fractionCmp(fraction ps1, fraction ps2){
    long smd =(ps1.numerator*ps2.denominator -
ps2.numerator*ps1.denominator
)*ps1.denominator*ps2.denominator;
    if(smd> 0)return 1;
    else if (smd<0) return -1;
    else return 0;
}
```

# Lời giải

```
void inverseArray(fraction dsps[], int n){
    int i;
    for(i=0; i<n; i++) {
        dsps[i]= inverse(dsps[i]);
    }
}

int main(){
    int n;
    fraction a[MAX], max, s, p;
    printf("Enter the number of fractions:");
    scanf("%d",&n);
    fractionArrayInput(a, n);
    fractionArrayOutput(a, n);
    printf("Inverse all fractions in the array.\n");
    inverseArray(a,n);
}
```

# Lời giải

```
printf("Enter first fraction to compare:");  
fractionInput(s);  
printf("Enter second fraction to compare:");  
fractionInput(p);  
if (fractionCmp(s,p) ==0)  
    printf("Two fractions are equals.\n");  
else if (fractionCmp(s,p) < 0) {  
    fractionOutput(s);  
    printf("is smaller than");  
    fractionOutput(p);  
    printf("\n");  
}  
return 0;  
}
```

# Lời giải

```
s=TongCacPS(a, n);  
    printf("\nTong gia tri cac phan so co trong mang: ");  
    XuatPS(s);  
  
    p=TichCacPS(a, n);  
    printf("\nTich gia tri cac phan so co trong mang: ");  
    XuatPS(p);  
  
    NghichDaoCacPS(a, n);  
    printf("\nMang phan so sau khi nghich dao cac phan tu: ");  
    XuatMangPS(a, n);  
    return 0;  
}
```



# Bài tập 14.5

- Viết chương trình thực hiện các thao tác sau:
  - Tối giản phân số
  - Nhân, cộng hai phân số
- Tích hợp vào bài 14.4



25 YEARS ANNIVERSARY  
**SOICT**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you  
for your  
attentions!**

