

---

## LAB 09 (Bài 20 Các kiểu dữ liệu nâng cao và Sắp xếp)

---

### Mục tiêu:

*Kết thúc bài học này, bạn có thể:*

- Sử dụng cấu trúc và các mảng cấu trúc
- Truyền tham số cấu trúc vào hàm
- Sắp xếp mảng.

Các bước trong bài này được giải thích chi tiết, đầy đủ. Mục đích là nắm được những nội dung trong bài lý thuyết và sử dụng thành thạo được công cụ. Hãy làm theo các bước dưới đây một cách cẩn thận.

### Phần I – Trong thời gian 1 giờ 30 phút đầu:

#### 20.1 Cấu trúc

---

Một cấu trúc là một nhóm các mẫu dữ liệu có thể có kiểu khác nhau. Mỗi cấu trúc phải được định nghĩa trước khi nó được sử dụng trong khai báo biến. Một định nghĩa cấu trúc có thể bao gồm một thành phần là một cấu trúc khác. Việc khởi tạo cấu trúc tương tự như việc khởi tạo mảng.

##### 20.1.1 Các mảng cấu trúc và sắp xếp

Trong C, có thể tạo mảng cấu trúc. Cũng như với mảng, dữ liệu trong các mảng cấu trúc có thể được sắp xếp theo hai phương pháp Selection sort và Bubble sort. Chúng ta hãy viết một chương trình C để cài đặt một hệ thống quản lý thư viện cơ bản. Hệ thống lưu trữ danh mục sách và ghi nhận các giao dịch mượn và trả sách. Sử dụng hệ thống này, ta có thể thêm thông tin chi tiết của một sách, ghi nhận các giao dịch mượn/trả sách và sắp xếp các ghi nhận này. Các bước để tạo hệ thống được liệt kê như sau:

1. Định nghĩa một cấu trúc để lưu trữ chi tiết sách. Câu lệnh sẽ là:

```
struct book_st{
    int book_cd;
    char book_nm[30];
    char author[30];
    int copies;
};
```

2. Định nghĩa một cấu trúc để lưu trữ các giao dịch mượn/trả sách. Lưu ý rằng ngày mượn/trả cũng sẽ là một cấu trúc và cũng phải được định nghĩa. Câu lệnh sẽ là:

```
struct date_st { int month;
                 int day;
                 int year; };

struct tran_st { int book_code;
                 char tran_type;
                 struct date_st tran_dt; };
```

3. Khai báo các biến có hai kiểu cấu trúc trên. Để thực hành, ta giả sử rằng cần lưu trữ chi tiết của 5 quyển sách và 10 giao dịch. Câu lệnh sẽ là:

```
struct book_st books[5];
struct tran_st trans[10];
```

4. Viết một vòng lặp để hiển thị menu các thao tác mà chương trình có thể thực hiện.. Câu lệnh sẽ là:

```
while(choice != 4)
{
    clrscr();
    printf("\nSelect from Menu\n1. Add book names\n2.
        Record Issue/Return\n3. Sort Transactions\n4.
        Exit\n\nEnter choice: ");
    scanf("%d", &choice);
    .
    .
    .
}
```

5. Nếu thao tác được chọn là thêm sách, thì nhập các thông tin chi tiết của sách trong một vòng lặp. Câu lệnh sẽ là:

```
for(i = 0; i < 5 && addflag == 'y'; i++)
{
    books[i].book_cd = i + 1;
    printf("\n\nBook code: %d\n\nBook name: ", i + 1);
    scanf("%s", books[i].book_nm);
    printf("\nAuthor: ");
    scanf("%s", books[i].author);
    printf("\nNumber of copies: ");
    scanf("%d", &books[i].copies);
    printf("\n\nContinue? (y/n): ");
    scanf(" %c", &addflag);
}
```

6. Nếu thao tác được chọn là thêm giao dịch, đặt một vòng lặp để nhập các thông tin chi tiết của giao dịch. Câu lệnh sẽ là:

```
for(i = 0; i < 10 && addflag == 'y'; i++)
{
    printf("\n\nBook code: ");
    scanf("%d", &trans[i].book_code);
    printf("\nIssue or Return?(I/R): ");
    scanf(" %c", &trans[i].tran_type);
    printf("\nDate: ");
    scanf("%d %d %d",
        &trans[i].tran_dt.month, &trans[i].tran_dt.day,
        &trans[i].tran_dt.year);
    printf("\n\nContinue? (y/n): ");
    scanf("%c", &addflag);
}
```

7. Nếu thao tác được chọn là sắp xếp các giao dịch, thì truyền tham số mảng cấu trúc vào hàm. Hàm sẽ sắp xếp mảng theo mã sách sử dụng phương pháp bubble sort. Câu lệnh sẽ là:

```
for(i = 0; i < 10; i++)
```

```

for(j = i + 1; j < 10; j++)
{
    if(tran[i].book_code > tran[j].book_code)
    {
        temptran=tran[i];
        tran[i]=tran[j];
        tran[j]=temptran;
    }
}

```

8. Hiển thị số giao dịch cho mỗi quyển sách trong hàm sắp xếp. Câu lệnh sẽ là:

```

for(i = 0, j = 0; i < 10; j = 0)
{
    tempcode = tran[i].book_code;
    while(tran[i].book_code == tempcode && i < 10)
    {
        j++;
        i++;
    }
    printf("\nBook code %d had %d transactions",
        tempcode, j);
}

```

Tiến hành thực hiện các bước sau đây để có chương trình hoàn chỉnh

**1. Mở chương trình soạn thảo mà bạn dùng để gõ chương trình C.**

**2. Tạo một tập tin mới**

**3. Gõ vào các dòng lệnh sau đây:**

```

#include<stdio.h>

struct book st {    int book cd;
                   char book_nm[30];
                   char author[30];
                   int copies; };

struct date st {    int month;
                   int day;
                   int year; };

struct tran_st {    int book_code;
                   char tran_type;
                   struct date_st tran_dt; };

void main()
{
    int choice = 1, i;
    char addflag;
    struct book_st books[5];
    struct tran_st trans[10];

    while(choice != 4)
    {        clrscr();

```

```

printf("\nSelect from Menu\n1. Add book names\n2.
Record Issue/Return\n3. Sort Transactions\n4.
Exit\n\nEnter choice: ");
scanf("%d", &choice);

if(choice == 1)
{
    addflag = 'y';
    clrscr();
    for(i = 0; i < 5 && addflag == 'y'; i++)
    {
        books[i].book_cd = i + 1;
        printf("\n\nBook code: %d\n\nBook name:",
               i+1);

        scanf("%s", books[i].book_nm);
        printf("\nAuthor: ");
        scanf("%s", books[i].author);
        printf("\nNumber of copies: ");
        scanf("%d", &books[i].copies);
        printf("\n\nContinue? (y/n): ");
        scanf(" %c", &addflag);
    }
}
else if(choice == 2)
{
    addflag = 'y';

    clrscr();
    for(i = 0; i < 10 && addflag == 'y'; i++)
    {
        printf("\n\nBook code: ");
        scanf("%d", &trans[i].book_code);
        printf("\nIssue or Return?(I/R): ");
        scanf(" %c", &trans[i].tran_type);
        printf("\nDate: ");
        scanf("%d %d %d",
               &trans[i].tran_dt.month,
               &trans[i].tran_dt.day,
               &trans[i].tran_dt.year);
        printf("\n\nContinue? (y/n): ");
        scanf(" %c", &addflag);
    }
}
else if(choice == 3)
{
    sorttran(trans);
}

}

sorttran(struct tran_st tran[10])
{
    int i, j, tempcode;
    struct tran_st temptran;

    clrscr();
    for(i = 0; i < 10; i++)
        for(j = i + 1; j < 10; j++)
        {

```

```

        if(tran[i].book_code > tran[j].book_code)
        {
            temptran = tran[i];
            tran[i] = tran[j];
            tran[j] = temptran;
        }
    }

    for(i = 0, j = 0; i < 10; j = 0)
    {
        tempcode = tran[i].book_code;
        while(tran[i].book_code == tempcode && i < 10)
        {
            j++;
            i++;
        }
        printf("\nBook code %d had %d transactions",
                tempcode, j);
    }
    getch();
}

```

Để xem kết quả, thực hiện các bước sau đây:

4. Lưu tập tin với tên structI.C.
5. Biên dịch tập tin, structI.C.
6. Thực thi chương trình, structI.C.
7. Trở về chương trình soạn thảo.

Mẫu kết quả của chương trình như sau:

```

Select from Menu
1. Add book names
2. Record Issue/Return
3. Sort Transactions
4. Exit

```

Enter choice:

Nếu nhập vào 1, mẫu kết xuất của chương trình sẽ là:

```

Book code: 1
Book name: Detective
Author: Hailey
Number of copies: 3

```

Continue? (y/n): y

Nếu nhập vào 2, mẫu kết xuất của chương trình sẽ là:

```

Book code: 1
Issue or Return? (I/R): I
Date: 2 22 03

```

Continue? (y/n): y

---

Nếu nhập vào 3, mẫu kết xuất của chương trình sẽ là:

Book code 1 had 3 transactions  
Book code 2 had 1 transactions  
Book code 3 had 2 transactions  
Book code 4 had 0 transactions  
Book code 5 had 4 transactions

## Phần II – Trong thời gian 30 phút kế tiếp:

1. Viết một chương trình C để lưu trữ các thông tin về sinh viên trong một cấu trúc. Dữ liệu phải bao gồm mã sinh viên, tên sinh viên, khóa học đã đăng ký và năm đăng ký. Viết một hàm để hiển thị các thông tin chi tiết của các sinh viên đã nhập học trong một năm học nào đó. Viết một hàm khác để xác định và hiển thị thông tin chi tiết của một sinh viên khi biết mã của sinh viên đó.

Yêu cầu:

- a. Định nghĩa một cấu trúc để lưu trữ thông tin chi tiết của sinh viên.
- b. Khai báo và khởi tạo biến cấu trúc với thông tin chi tiết của 10 sinh viên.
- c. Viết vòng lặp để hiển thị một menu các thao tác mà chương trình có thể thực hiện.
- d. Nhận vào lựa chọn danh mục và gọi hàm thích hợp với tham số là mảng cấu trúc.
- e. Trong hàm dùng để hiển thị thông tin chi tiết của các sinh viên nhập học trong một năm, viết chương trình để thực hiện nhập vào năm học cần được hiển thị thông tin, sau đó sử dụng vòng lặp để kiểm tra năm nhập học của từng sinh viên, nếu trùng với năm cần hiển thị thông tin yêu cầu thì hiển thị thông tin của sinh viên đó. Ngoài ra, hàm này còn cho phép người dùng có thể tiếp tục thực hiện việc hiển thị thông tin của những năm khác cho đến khi họ không muốn sử dụng chức năng này nữa.
- f. Hàm dùng để hiển thị thông tin chi tiết của sinh viên cho phép nhập vào mã của sinh viên, dùng một vòng lặp để kiểm tra mã của mỗi sinh viên, nếu mã của sinh viên nào trùng với mã đã được nhập thì hiển thị thông tin chi tiết của sinh viên đó. Ngoài ra, hàm này còn cho phép người dùng có thể tiếp tục thực hiện việc hiển thị thông tin của những sinh viên khác cho đến khi họ không muốn sử dụng chức năng này nữa

## Bài tập tự làm

---

1. Viết một chương trình C để lưu trữ 5 độ dài trong một mảng cấu trúc. Mỗi độ dài phải bao gồm 3 thông tin về yards, feet và inches. Sắp xếp và hiển thị các độ dài.
2. Viết một chương trình C để lưu trữ thông tin chi tiết của nhân viên trong một mảng cấu trúc. Thông tin của một nhân viên phải bao gồm mã nhân viên, tên, lương và ngày vào làm. Ngày vào làm phải được lưu trong một cấu trúc khác. Chương trình phải thực hiện các thao tác sau đây dựa trên sự lựa chọn trong menu các chức năng của chương trình:

- a. Tăng lương theo các luật sau:

Salary Range	Percentage increase
<= 2000	15%
> 2000 and <= 5000	10%
>5000	No increase

- b. Hiển thị thông tin chi tiết của các nhân viên đã làm việc trong công ty từ 10 năm trở lên.