

ĐỀ CƯƠNG ÔN TẬP MÔN CTDL & GT

Hình thức thi: Thực hành

Thời gian làm bài thi: 45 phút (1 bài hoàn chỉnh)

Các nội dung trọng tâm:

- + DSLK kê
- + DSLK đơn
- + DSLK đôi
- + Ngăn xếp
- + Hàng đợi

BÀI TẬP

Bài 1) Viết chương trình thực hiện các yêu cầu sau:

- Định nghĩa một cấu trúc **Sinh Viên** bao gồm các trường thông tin như sau: *Mã lớp, Mã sinh viên, Tên sinh viên, Năm sinh, Quê quán.*
- Cài đặt một cấu trúc danh sách liên kết đơn, với các thao tác: 1) *Khởi tạo danh sách;* 2) *Tạo mới một nút chứa dữ liệu;* 3) *Chèn nút vào danh sách;* 4) *Xóa một nút bất kỳ khỏi danh sách;* 5) *Duyệt danh sách.*
- Chương trình chính: Sử dụng các thao tác trên để:
 - + Tạo một danh sách liên kết đơn bao gồm n sinh viên với đầy đủ các thông tin như trên.
 - + Hiển thị danh sách đã nhập ra màn hình.
 - + Nhập vào từ bàn phím một *mã sinh viên* bất kỳ, kiểm tra xem sinh viên có mã tương ứng với mã vừa nhập đã được lưu trong danh sách chưa? Nếu có xóa sinh viên đó khỏi danh sách và hiển thị lại danh sách sau khi đã xóa. Ngược lại, thông báo không tìm thấy.

Bài 2) Viết chương trình thực hiện các yêu cầu sau:

- Khai báo kiểu cấu trúc **Nhân viên** bao gồm các thông tin: *Mã nhân viên, tên nhân viên, năm sinh, hệ số lương.*
- Cài đặt một cấu trúc **danh sách Liên kết đơn**, với các thao tác: 1) *Khởi tạo danh sách;* 2) *Tạo mới một nút chứa dữ liệu;* 3) *Chèn nút vào danh sách;* 4) *Duyệt danh sách;* 5) *Tìm kiếm một phần tử trong danh sách.*
- Chương trình chính: Sử dụng các thao tác đã cài đặt để:

- + Tạo một danh sách Liên kết đơn lưu **n** Nhân viên với đầy đủ các thông tin như trên (n bất kỳ nhập từ bàn phím).
- + Hiển thị danh sách Nhân viên đã nhập ra màn hình.
- + Nhập vào một mã nhân viên bất kỳ, tìm kiếm xem nhân viên có mã tương ứng đã được lưu trong danh sách, nếu tìm thấy hiển thị thông tin tương ứng của viên đó ra màn hình. Ngược lại, thông báo không tìm thấy.'

Bài 3) Viết chương trình thực hiện các yêu cầu sau:

- Định nghĩa cấu trúc **Giangvien** gồm các thông tin: *Mã giảng viên, Tên giảng viên, Học vị, Mức lương.*
- Cài đặt một cấu trúc **danh sách Liên kết đơn**, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào danh sách*; 4) *Duyệt danh sách*; 5) *Xóa phần tử trong danh sách.*
- Chương trình chính: Sử dụng các thao tác đã cài đặt để:
 - + Nhập vào 1 danh sách gồm **n** giảng viên (n bất kỳ).
 - + Nhập vào 1 mã giảng viên bất kỳ, tìm kiếm xem nhân viên có mã tương ứng đã được lưu trong danh sách, nếu có xóa khỏi danh sách
 - + Hiển thị danh sách trước và sau khi xóa ra màn hình.

Bài 4) Viết chương thực hiện các công việc sau:

- Định nghĩa một cấu trúc **Sách**, bao gồm các trường thông tin như sau: *Mã sách, Tên sách, Tên tác giả, Năm xuất bản, Nhà xuất bản, Giá thành.*
- Cài đặt một cấu trúc danh sách liên kết đơn, với các thao tác cơ bản như *khởi tạo danh sách, Tạo mới một nút chứa dữ liệu, Chèn nút vào danh sách, Xóa nút khỏi danh sách, Duyệt danh sách.*
- Chương trình chính:
 - + Nhập từ bàn phím thông tin về N cuốn sách (N là một số nguyên dương bất kỳ) và lưu vào danh sách đã tạo. Hiển thị danh sách đã nhập ra màn hình.
 - + Sắp xếp (giảm dần) danh sách đã nhập theo năm xuất bản.
 - + Hiển thị danh sách đã nhập ra màn hình.

Bài 5) Viết chương thực hiện các công việc sau:

- Định nghĩa một cấu trúc **Khách hàng**, bao gồm các trường thông tin (*mã KH, loại KH, Họ tên, Địa chỉ, Số ĐT*).
- Cài đặt một cấu trúc danh sách liên kết đơn, với các thao tác cơ bản như *khởi tạo danh sách, Tạo mới một nút chứa dữ liệu, Chèn nút vào danh sách, Xóa nút khỏi danh sách, Duyệt danh sách.*

- Chương trình chính:
 - + Nhập 1 danh sách gồm n khách hàng (n bất kỳ).
 - + Sắp xếp danh sách trên theo chuỗi họ tên khách hàng (thứ tự alphabet).
 - + Hiển thị danh sách đã sắp ra màn hình.

Bài 6) Viết chương thực hiện các công việc sau:

- Định nghĩa một cấu trúc **Dự án**, gồm các trường thông tin (*mã dự án, Tên dự án, Kiểu dự án, Tổng kinh phí*) – trong đó kiểu dự án có thể gồm: nhỏ, trung bình hoặc lớn (1 điểm).
- Cài đặt một cấu trúc danh sách liên kết đơn, với các thao tác cơ bản như *khởi tạo danh sách, Tạo mới một nút chứa dữ liệu, Chèn nút vào danh sách, Xóa nút khỏi danh sách, Duyệt danh sách*.
- Chương trình chính:
 - + Nhập 1 danh sách gồm n dự án (n bất kỳ)
 - + Sắp xếp danh sách trên theo tổng kinh phí bằng thuật toán Bubble_Sort.
 - + Hiển thị danh sách đã sắp ra màn hình.

Bài 7) Viết chương trình thực hiện các yêu cầu sau:

- Định nghĩa một cấu trúc **Sách**, bao gồm các trường thông tin như sau: *Mã sách, Tên sách, Tên tác giả, Năm xuất bản, Nhà xuất bản, Giá thành*.
- Cài đặt một cấu trúc danh sách liên kết đơn, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào danh sách*; 4) *Duyệt danh sách*; 5) *Sắp xếp danh sách đã nhập theo năm xuất bản (tăng dần)*.
- Chương trình chính: Sử dụng các thao tác trên để:
 - + Tạo một danh sách liên kết đơn bao gồm n cuốn sách (n là một số nguyên dương bất kỳ) và lưu vào danh sách đã tạo.
 - + Hiển thị danh sách các cuốn sách đã nhập ra màn hình.
 - + Hiển thị danh sách đã sắp xếp theo năm xuất bản ra màn hình.

Bài 8: Viết chương trình cài đặt một cấu trúc DSLK đơn với các thao tác tương ứng.

- Chương trình chính:
 - o Nhập vào 1 DSLK đơn chứa n số nguyên.
 - o Tìm phần tử dương nhỏ nhất và phần tử âm lớn nhất trong danh sách.
 - o Nhập vào 1 số nguyên x bất kỳ, kiểm tra xem x xuất hiện bao nhiêu lần (có bao nhiêu nút chứa giá trị x) trong danh sách?
 - o Xóa khỏi danh sách tất cả các nút chứa giá trị x đã được tìm thấy ở bước trên.

Bài 9) Viết chương trình thực hiện các công việc sau:

- Định nghĩa một cấu trúc Định nghĩa một cấu trúc **Sinh Viên** bao gồm các trường thông tin như sau: *Mã lớp, Mã sinh viên, Tên sinh viên, Ngày sinh* (kiểu Date, gồm 3 trường thông tin ngày, tháng, năm), *Điểm môn 1, Điểm môn 2, Điểm môn 3*.
- Cài đặt một cấu trúc danh sách liên kết đôi, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào danh sách*; 4) *Duyệt danh sách*; 5) *Tìm kiếm nút trong danh sách*, 6) *xóa phần tử trong danh sách*.
- Chương trình chính:
 - + Tạo 1 danh sách liên kết đôi lưu n Sinh viên.
 - + Hiển thị danh sách đã nhập ra màn hình.
 - + Sắp xếp danh sách trên theo thứ tự tăng dần của tổng điểm (điểm môn 1+ điểm môn 2+ điểm môn 3). Hiển thị danh sách đã sắp ra màn hình
 - + Nhập vào 1 tên sinh viên bất kỳ, hiển thị ra màn hình đầy đủ thông tin của những sinh viên có tên trùng với tên vừa nhập.
 - + Nhập vào 1 mã sinh viên bất kỳ, tìm kiếm xem sinh viên có mã tương ứng đã được lưu trong danh sách chưa? Nếu tìm thấy, xóa khỏi danh sách.

Bài 10) Viết chương trình thực hiện các công việc sau:

- Định nghĩa một cấu trúc Định nghĩa một cấu trúc **Phân số** bao gồm các trường thông tin như sau: *tử số, mẫu số*.
- Cài đặt một cấu trúc danh sách liên kết đôi, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào danh sách*; 4) *Duyệt danh sách*; 5) *Tìm kiếm nút trong danh sách*, 6) *xóa phần tử trong danh sách*.
- Chương trình chính:
 - + Tạo 1 danh sách liên kết đôi lưu n phân số.
 - + Hiển thị danh sách đã nhập ra màn hình.
 - + Tìm phân số lớn nhất và nhỏ nhất trong danh sách, hiển thị kết quả ra màn hình.
 - + Sắp xếp danh sách theo thứ tự giảm dần, hiển thị danh sách đã sắp ra màn hình.

Bài 11) Viết chương trình thực hiện các công việc sau:

- Định nghĩa một cấu trúc **Mặt hàng**, bao gồm các trường thông tin (*mã hàng, Tên hàng, Nhóm hàng, Giá bán*) – trong đó Nhóm hàng có thể là: Hàng thời trang, Hàng tiêu dùng, Hàng điện máy, Hàng gia dụng.

- Cài đặt một cấu trúc danh sách liên kết đôi, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào danh sách*; 4) *Duyệt danh sách*; 5) *Tìm kiếm nút trong danh sách*, 6) *xóa phần tử trong danh sách*.
- Chương trình chính:
 - + Tạo một danh sách liên kết đôi chứa n mặt hàng từ bàn phím sử dụng các thao tác đã cài đặt ở trên.
 - + Hiển thị danh sách đã tạo ra màn hình.
 - + Sắp xếp danh sách đã tạo theo thứ tự tăng dần của mã hàng, hiển thị danh sách đã sắp ra màn hình.
 - + Nhập vào 1 mã hàng bất kỳ, kiểm tra xem mã hàng vừa nhập đã được lưu trong danh sách chưa? Nếu chưa, nhập đầy đủ thông tin của mặt hàng có mã vừa nhập và chèn vào danh sách theo đúng thứ tự đã sắp (không làm mất tính đã sắp của

Bài 12) Viết chương trình thực hiện các yêu cầu sau:

- Cài đặt một cấu trúc **ngăn xếp** bằng mảng, với các thao tác: 1) *Khởi tạo ngăn xếp*; 2) *Kiểm tra ngăn xếp rỗng*; 3) *Kiểm tra ngăn xếp đầy*; 4) *Đưa một phần tử vào ngăn xếp*, 5) *lấy một phần tử ra khỏi ngăn xếp*.
- Chương trình chính: Sử dụng các thao tác đã cài đặt để:
 - + Nhập vào một số nguyên dương N bất kỳ, sử dụng cấu trúc ngăn xếp để in ra biểu diễn nhị phân của N.

Ví dụ: $N = 11 \Rightarrow$ Biểu diễn nhị phân: 1011

Bài 13) Viết chương trình thực hiện các yêu cầu sau:

- Cài đặt một cấu trúc **ngăn xếp** bằng mảng, với các thao tác: 1) *Khởi tạo ngăn xếp*; 2) *Kiểm tra ngăn xếp rỗng*; 3) *Kiểm tra ngăn xếp đầy*; 4) *Đưa một phần tử vào ngăn xếp*, 5) *lấy một phần tử ra khỏi ngăn xếp*.
- Chương trình chính: Sử dụng các thao tác đã cài đặt để:
 - + Nhập vào một số nguyên dương N bất kỳ (hệ cơ số 10), sử dụng cấu trúc ngăn xếp để in ra giá trị ở hệ cơ số 8 của N ra màn hình:

Ví dụ: $n = 1042_{10} \rightarrow 2022_8$

Bài 14: Dùng Stack kiểm tra tính hợp lệ của một

Ví dụ 1: $((()()))$ là hợp lệ

Ví dụ 1: $((()))()$ là không hợp lệ

Gợi ý: Đọc dấu từ trái sang phải:

- Nếu gặp dấu (thì đưa vào ngăn xếp.
- Nếu gặp dấu) thì pop 1 dấu ra khỏi ngăn xếp.
- + Nếu gặp dấu) mà ngăn xếp rỗng → BT sai.
- + Nếu duyệt hết biểu thức và ngăn xếp không rỗng → BT sai
- + Nếu duyệt chưa hết biểu thức và ngăn xếp rỗng → BT đúng

Bài 15) Viết chương trình thực hiện các yêu cầu sau:

- Cài đặt một cấu trúc hàng đợi, với các thao tác: 1) *Khởi tạo hàng đợi*; 2) *Kiểm tra hàng đợi rỗng*; 3) *Kiểm tra hàng đợi đầy*; 4) *Đưa một phần tử vào hàng đợi*; 5) *Lấy một phần tử ra khỏi hàng đợi* (4 điểm).
- Chương trình chính: Sử dụng các thao tác đã cài đặt để:
 - + Nhập vào một số nguyên dương N bất kỳ
 - + Sử dụng cấu trúc hàng đợi để tính tổng các chữ số của N (5 điểm).

Ví dụ: $N = 1996 \Rightarrow \text{Tổng} = 1 + 9 + 9 + 6 = 25$.

Bài 16: Viết chương trình cài đặt cấu trúc Stack bằng danh sách liên kết với các thao tác tương ứng. Sử dụng cấu trúc Stack trên để chuyển một số từ hệ thập phân (cơ số 10) sang hệ thập lục phân (cơ số 16).

Gợi ý:

- Nhập vào 1 số nguyên hệ thập phân.
- Chia liên tiếp số đó cho 16 → Lấy phần dư theo chiều ngược lại.

Bài 17: Viết chương trình thực hiện yêu cầu sau:

Định nghĩa một cấu trúc **Sinh viên** bao gồm các trường thông tin như sau: *Mã Sinh viên* (kiểu số nguyên), *tên Sinh viên* (kiểu chuỗi ký tự), *lớp* (kiểu chuỗi ký tự), *điểm tổng kết* (kiểu số thực), *hạng kiểm* (Tốt, khá, trung bình, yếu).

Cài đặt một cấu trúc **danh sách liên kết đơn** cho kiểu dữ liệu Sinh viên, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào cuối danh sách*; 4) *Tìm kiếm phần tử trong danh sách*; 5) *Xóa nút cuối cùng khỏi danh sách*; 6) *Duyệt danh sách*.

Chương trình chính: Sử dụng cấu trúc danh sách liên kết đơn và các thao tác ở trên để:

- Nhập vào một danh sách liên kết đơn gồm n Sinh viên (n bất kỳ).
- Hiển thị danh sách đã nhập ra màn hình.

- Liệt kê ra màn hình danh sách tất cả những Sinh viên thuộc lớp “D13CNPM”.
- Xóa nút cuối cùng khỏi danh sách. Hiện thị lại danh sách sau khi xóa ra màn hình.

Bài 18: Viết chương trình thực hiện yêu cầu sau:

Định nghĩa một cấu trúc **Học viên** bao gồm các trường thông tin như sau: *Mã học viên* (kiểu số nguyên), *tên học viên* (kiểu chuỗi ký tự), *lớp* (kiểu chuỗi ký tự), *chuyên ngành* (Khoa học máy tính, Thương mại điện tử, Quản trị an ninh mạng), *điểm tổng kết* (kiểu số thực).

Cài đặt một cấu trúc **danh sách liên kết đôi** cho kiểu dữ liệu Học viên, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào cuối danh sách*; 4) *Tìm kiếm phần tử trong danh sách*; 5) *Duyệt danh sách*.

Chương trình chính: Sử dụng cấu trúc danh sách liên kết đôi và các thao tác đã tạo để:

- Nhập và 1 danh sách gồm n Học viên (n bất kỳ).
- Hiện thị danh sách đã nhập ra màn hình.
- Liệt kê ra màn hình danh sách tất cả những Học viên thuộc chuyên ngành “Thương mại điện tử” và có điểm tổng kết ≥ 8.0 .
- Sắp xếp danh sách đã nhập theo tên học viên (thứ tự alphabet), hiện thị danh sách đã sắp ra màn hình.

Bài 19: Viết chương trình thực hiện yêu cầu sau: Khai báo cấu trúc **date** gồm các thông tin: *ngày, tháng, năm*.

```
struct date{  
    int ngay, thang, nam;  
};
```

Khai báo cấu trúc loại hàng hóa bao gồm các thông tin: *Mã hàng hóa, tên hàng hóa, ngày xuất hàng (kiểu ngày/tháng/năm), ví dụ 10/10/2000), giá xuất hàng đi (đơn vị triệu đồng)*:

```
struct hang  
{  
    char ID[6];  
    char ten[256];  
    date ngay_xuat;  
    float gia_xuat;  
};
```

Viết các hàm nhập vào từ bàn phím giá trị của một biến cấu trúc hàng hóa, hiển thị lên màn hình.

Nhập vào một số nguyên dương *n*, sử dụng cấu trúc dữ liệu ngăn xếp (stack) nhập vào một chồng hàng gồm *n* hàng hóa.

Hiển thị lên màn hình thông tin các hàng hóa trong chồng hàng có giá xuất hàng (trường **gia_xuat**) > 1.0.

Xuất kho chồng hàng hóa cần ghi lại thông tin hàng hóa mỗi khi xuất hàng cho đến hết hàng vào một file text là **hanghoa.txt** theo cách ghi như sau (1đ):

Giá trị các phần của chồng hàng theo từng dòng (mỗi hàng hóa 1 dòng, mỗi giá trị trường cấu trúc hàng hóa cách nhau 1 dấu tab, tức ký tự ‘\t’)

Ví dụ :

```
001    TV1           10/09/2019   20  
002    TV2           10/09/2019    6
```

Bài 20: Viết chương trình thực hiện yêu cầu sau: Khai báo cấu trúc **date** gồm các thông tin: *ngày, tháng, năm*.

```
struct date  
{
```



```
int ngay, thang, nam;  
  
};
```

Khai báo cấu trúc hàng hóa cần bốc xếp trên băng chuyền bao gồm các thông tin: *Mã hàng hóa, tên hàng hóa, ngày xuất hàng, giá xuất hàng đi* (đơn vị triệu đồng):

```
struct hang  
  
{  
  
    char ID[6];  
  
    char ten[256];  
  
    date ngay_xuat;  
  
    float gia_xuat;  
  
};
```

Viết các hàm nhập vào từ bàn phím giá trị của một biến cấu trúc hàng hóa, hiển thị lên màn hình.

Nhập vào 1 số nguyên dương n , sử dụng cấu trúc dữ liệu hàng đợi (queue) nhập vào băng chuyền gồm n hàng hóa cần bốc xếp.

Hiển thị lên màn hình thông tin tất cả các hàng hóa trong hàng đợi có mã hàng hóa (trường ID) là “001”.

Hiển thị lên màn hình thông tin hàng hóa đã bốc xếp (hiện tại băng chuyền đã rỗng).

Bài 21: Viết chương trình thực hiện yêu cầu sau: Khai báo cấu trúc sinh viên bao gồm các thông tin mã sinh viên (ID), họ tên, năm sinh, lớp, điểm tổng kết:

```
struct HocSinh  
  
{  
  
    char ID[6];  
  
    char ho_ten[80];  
  
    int nam_sinh;  
  
    char lop[100];
```

```
float diem_tk;  
};
```

Viết các hàm nhập vào từ bàn phím giá trị của một biến cấu trúc **Học sinh**, hiển thị lên màn hình.

Nhập vào 1 số nguyên dương **n**, sử dụng cấu trúc danh sách (dùng cài đặt mảng – danh sách kê hoặc cài đặt con trỏ-danh sách liên kết), nhập vào danh sách gồm **n** học sinh. Hiển thị danh sách lên màn hình.

Hiển thị lên màn hình thông tin đầy đủ của học sinh có điểm tổng kết lớn nhất.

Chèn vào đầu danh sách một học sinh mới được chuyển trường tới (thông tin sinh viên này được nhập từ bàn phím). Hiển thị danh sách sau khi chèn ra màn hình.

Bài 22: Viết chương trình thực hiện yêu cầu sau:

Định nghĩa một cấu trúc **Khách hàng**, bao gồm các trường thông tin như sau: *Mã khách hàng* (kiểu số nguyên), *tên khách hàng* (kiểu chuỗi ký tự), *địa chỉ* (kiểu chuỗi ký tự), *số điện thoại* (kiểu chuỗi ký tự), *tổng tiền thanh toán* (kiểu số).

Cài đặt một **cấu trúc danh sách liên kết đôi** cho kiểu dữ liệu Khách hàng, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào cuối danh sách*; 4) *Tìm kiếm phần tử trong danh sách*; 5) *Sắp xếp danh sách theo số lượng dịch vụ sử dụng giảm dần*; 6) *Duyệt danh sách*;

Xây dựng chương trình chính, sử dụng các thao tác đã cài đặt để:

- Tạo một danh sách liên kết đôi gồm n khách hàng (n bất kỳ).
- Hiển thị danh sách đã nhập ra màn hình.
- Liệt kê ra màn hình thông tin đầy đủ của tất cả những khách hàng có địa chỉ tại Hà Nội.
- Sắp xếp danh sách khách hàng theo tổng tiền thanh toán giảm dần, hiển thị danh sách đã sắp ra màn hình.

Bài 23: Vết chương trình thực hiện yêu cầu sau: Khai báo cấu trúc **date** gồm các thông tin: *ngày, tháng, năm*.

```
struct date
{
    int ngay, thang, nam;
};
```

Khai báo cấu trúc nhân viên bao gồm các thông tin: *Mã nhân viên, họ tên, ngày sinh (kiểu ngày/tháng/năm, ví dụ 10/10/2000), quê quán, lương* (đơn vị triệu đồng):

```
struct nv
{
    char ID[6];
    char ho_ten[80];
    date ngay_sinh;
    char que_quan[100];
    float luong;
};
```

Viết các hàm nhập vào từ bàn phím giá trị của một biến cấu trúc **nv**, hiển thị lên màn hình.

Nhập vào số nguyên dương **n**, sử dụng cấu trúc dữ liệu danh sách liên kết đôi, nhập vào danh sách **n** nhân viên.

Hiển thị danh sách liên kết đôi lên màn hình theo kiểu hiển thị từ **cuối** danh sách.

Bổ sung một nhân viên mới tuyển dụng (nhập thông tin nhân viên từ bàn phím) vào **đầu** danh sách. Hiển thị danh sách sau khi chèn lên màn hình.

Bài 24: Viết chương trình thực hiện yêu cầu sau:

Định nghĩa kiểu dữ liệu cấu trúc **Nhân viên** bao gồm các thông tin: *Mã nhân viên* (kiểu số nguyên), *tên nhân viên* (kiểu chuỗi ký tự), *tuổi* (kiểu số nguyên), *hệ số lương* (kiểu số thực).

Cài đặt một cấu trúc **danh sách liên kết đơn**, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào đầu danh sách*; 4) *Sắp xếp danh sách theo hệ số lương tăng dần*; 5) *Duyệt danh sách*;

Xây dựng chương trình chính sử dụng các thao tác đã cài đặt để:

- Tạo vào một danh sách liên kết đơn gồm **n** nhân viên (n bất kỳ).

- Hiển thị danh sách Nhân viên đã tạo ra màn hình.
- Liệt kê ra màn hình thông tin đầy đủ của những nhân viên đến tuổi về hưu (60 tuổi).
- Sắp xếp danh sách đã tạo theo hệ số lương tăng dần, hiển thị danh sách sau khi sắp ra màn hình.

Bài 25: Trên ngôn ngữ C/C++, viết chương trình thực hiện yêu cầu sau:

Định nghĩa một cấu trúc **Tài liệu**, bao gồm các trường thông tin như sau: *Mã tài liệu* (kiểu số nguyên), *Tên tài liệu* (kiểu chuỗi ký tự), *Loại tài liệu* (giáo trình, tài liệu báo cáo kỹ thuật, tài liệu hướng dẫn sử dụng), *Năm xuất bản* (kiểu số nguyên), *Nơi xuất bản* (kiểu chuỗi ký tự).

Cài đặt một cấu trúc **danh sách liên kết đôi**, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào cuối danh sách*; 4) *Sắp xếp danh sách*; 5) *Tìm kiếm nút trong danh sách*; 6) *Duyệt danh sách*.

Xây dựng chương trình chính sử dụng các thao tác trên để:

- Tạo một danh sách liên kết đôi bao gồm n tài liệu (n bất kỳ).
- Hiển thị danh sách đã tạo ra màn hình.
- Sắp xếp danh sách theo mã tài liệu tăng dần.
- Chèn vào cuối danh sách một tài liệu mới (do người dùng nhập từ bàn phím), hiển thị danh sách sau khi chèn ra màn hình.

Bài 26: Viết chương trình thực hiện yêu cầu sau:

Định nghĩa một cấu trúc **Mặt hàng**, bao gồm các trường thông tin như sau: *Mã hàng* (kiểu số nguyên), *tên hàng* (kiểu chuỗi ký tự), *Loại hàng* (gia dụng, điện máy, tiêu dùng, mỹ phẩm), *năm sản xuất* (kiểu số nguyên), *xuất xứ* (kiểu chuỗi ký tự), *giá bán* (kiểu số thực).

Cài đặt một cấu trúc danh **sách liên kết đôi** cho kiểu dữ liệu Mặt hàng, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào cuối danh sách*; 4) *Xóa khỏi danh sách tất cả các mặt hàng cơ xuất xứ tại nước x (bất kỳ)*; 5) *Duyệt danh sách*.

Xây dựng chương trình chính, sử dụng các thao tác đã cài đặt để:

- Tạo một danh sách liên kết đôi gồm n mặt hàng (n bất kỳ).
- Hiển thị danh sách đã tạo ra màn hình.
- Liệt kê ra màn hình tất cả các mặt hàng “gia dụng” có giá bán từ 500.000 đến 1000.000 đồng.
- Xóa khỏi danh sách tất cả các mặt hàng có xuất xứ ở “Trung Quốc”.

Bài 28: Viết chương trình thực hiện yêu cầu sau:

Định nghĩa một cấu trúc **Vé máy bay**, bao gồm các trường thông tin như sau: *Mã vé* (kiểu số nguyên), *loại vé* (hạng nhất, hạng thương gia, hạng phổ thông), *điểm khởi hành* (chuỗi ký tự), *điểm đến* (kiểu chuỗi ký tự), *giá vé* (kiểu số thực).

Cài đặt một cấu trúc **danh sách liên kết đơn** cho kiểu dữ liệu Vé máy bay, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút cuối danh sách*; 4) *Duyệt danh sách*;

Xây dựng chương trình chính, sử dụng các thao tác đã cài đặt để:

- Tạo một danh sách liên kết đơn gồm n Vé máy bay (n bất kỳ).
- Hiển thị danh sách đã tạo ra màn hình.
- Sắp xếp danh sách theo giá vé tăng dần, hiển thị danh sách sau khi sắp ra màn hình.
- Liệt kê ra màn danh sách tất cả những Vé máy bay có điểm khởi hành là sân bay “Nội Bài”.

Bài 29: Viết chương trình thực hiện yêu cầu sau:

Định nghĩa một cấu trúc **Phân số** bao gồm các trường thông tin như sau: *tử số* (kiểu số nguyên), *mẫu số* (kiểu số nguyên).

Cài đặt một cấu trúc hàng đợi bằng danh sách liên kết, với các thao tác: 1) *Khởi tạo hàng đợi*; 2) *Kiểm tra hàng đợi rỗng*; 4) *Đưa một phần tử vào hàng đợi*; 5) *Lấy một phần tử ra khỏi hàng đợi*.

Xây dựng chương trình chính để, sử dụng cấu trúc hàng đợi và các thao tác đã cài đặt trên hàng đợi để:

- Đưa vào hàng đợi n phân số bất kỳ (được nhập từ bàn phím).
- Hiển thị các phân số đã lưu trong hàng đợi và tổng của chúng ra màn hình.

Bài 30: Viết chương trình thực hiện yêu cầu sau:

Định nghĩa một cấu trúc **Dự án**, gồm các trường thông tin như sau: *Mã dự án* (kiểu số nguyên), *Tên dự án* (kiểu chuỗi ký tự), *Quy mô dự án* (nhỏ, trung bình, lớn), *Tổng kinh phí* (kiểu số thực- đơn vị tính bằng tỷ đồng).

Cài đặt một cấu trúc **danh sách liên kết đơn**, với các thao tác cơ bản như *khởi tạo danh sách*, *Tạo mới một nút chứa dữ liệu*, *Chèn nút vào đầu danh sách*, *Xóa nút khỏi danh sách*, *Duyệt danh sách*.

Xây dựng chương trình chính, sử dụng cấu trúc danh sách liên kết đơn và các thao tác đã cài đặt để:

- Nhập 1 danh sách gồm n dự án (n bất kỳ)
- Hiển thị danh sách đã nhập ra màn hình.
- Liệt kê ra màn hình những dự án những dự án có tổng kinh phí từ 5000-10.000 tỷ.
- Nhập vào một mã dự án bất kỳ, tìm kiếm xem mã dự án tương ứng đã được lưu trong danh sách hay chưa? Nếu tìm thấy xóa dự án đó khỏi danh sách. Hiển thị lại danh sách sau khi xóa ra màn hình.

Bài 31: Viết chương trình thực hiện yêu cầu sau:

Định nghĩa một cấu trúc **Công trình**, gồm các trường thông tin như sau: Mã công trình (kiểu số nguyên), Tên công trình (kiểu chuỗi ký tự), Qui mô công trình (nhỏ, trung bình, lớn), Tổng kinh phí (kiểu số thực- đơn vị tính bằng tỷ đồng).

Cài đặt một cấu trúc **danh sách liên kết đôi**, với các thao tác cơ bản như *khởi tạo danh sách*, *Tạo mới một nút chứa dữ liệu*, *Chèn nút vào đầu danh sách*, *Xóa nút khỏi danh sách*, *Duyệt danh sách*.

Xây dựng chương trình chính, sử dụng cấu trúc danh sách liên kết đơn và các thao tác đã cài đặt để:

- Nhập 1 danh sách gồm n Công trình (n bất kỳ)
- Hiển thị danh sách đã nhập ra màn hình.
- Liệt kê ra màn hình những Công trình những Công trình có tổng kinh phí từ 5000-10.000 tỷ.
- Nhập vào một mã Công trình bất kỳ, tìm kiếm xem mã dự án tương ứng đã được lưu trong danh sách hay chưa? Nếu tìm thấy xóa dự án đó khỏi danh sách. Hiển thị lại danh sách sau khi xóa ra màn hình.

Bài 32: Viết chương trình thực hiện yêu cầu sau: Khai báo cấu trúc **date** gồm các thông tin: *ngày, tháng, năm*.

```
struct date
```

```
{
```

```
    int ngay, thang, nam;
```

```
};
```

Khai báo cấu trúc **Món ăn** gồm các thông tin: *Mã món ăn, tên món ăn, ngày sản xuất, hạn sử dụng, giá thành* (đơn vị ngàn đồng).

```
struct MonAn
```

```
{
```

```
    char ID[6];
```

```
    char ten[100];
```

```

        date ngay_sx;
        int han_sd;
        float gia;

    };

```

Viết các hàm nhập vào từ bàn phím giá trị của một biến cấu trúc **Món ăn**, hiển thị lên màn hình.

Nhập vào 1 số nguyên dương **n**, sử dụng cấu trúc danh sách (dùng cài đặt mảng – danh sách kề hoặc cài đặt con trỏ-danh sách liên kết), nhập vào danh sách gồm **n** món ăn. Hiển thị danh sách lên màn hình.

Hiển thị lên màn hình thông tin đầy đủ của tất cả những món ăn được sản xuất vào ngày “05/10/2019”.

Chèn vào đầu danh sách một món ăn mới (do người dùng nhập từ bàn phím), hiển thị danh sách sau khi chèn ra màn hình.

Bài 33:Viết chương trình thực hiện yêu cầu sau:

1 (1 điểm): Định nghĩa một cấu trúc **Cầu thủ** bao gồm các trường thông tin như sau: *Mã cầu thủ* (kiểu số nguyên), *Tên cầu thủ* (kiểu chuỗi ký tự), *vị trí thi đấu* (thủ môn, tiền vệ, trung vệ, hậu vệ), *tuổi* (kiểu số nguyên), *tiền thưởng* (kiểu số thực – đơn vị tính bằng triệu đồng).

2 (4 điểm): Cài đặt một cấu trúc **danh sách liên kết đơn**, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào cuối danh sách*; 4) *Tìm kiếm phần tử trong danh sách*; 5) *Sắp xếp danh sách*; 6) *Duyệt danh sách*.

3 (5 điểm): Chương trình chính: Sử dụng danh sách liên kết đơn và các thao tác đã tạo để:

- Nhập vào một danh sách liên kết đơn gồm n cầu thủ (n bất kỳ).
- Hiển thị danh sách đã nhập ra màn hình.
- Sắp xếp danh sách cầu thủ theo tuổi giảm dần, hiển thị danh sách đã sắp ra màn hình.
- Liệt kê ra màn hình danh sách những cầu thủ thi đấu ở vị trí “trung vệ”.

Bài 34:Viết chương trình thực hiện yêu cầu sau:

Định nghĩa một cấu trúc **Vận động viên** bao gồm các trường thông tin như sau: *Mã Vận động viên* (kiểu số nguyên), *Tên Vận động viên* (kiểu chuỗi ký tự), *tuổi* (kiểu số nguyên), *Quê quán* (kiểu chuỗi ký tự), *Lương* (kiểu float – đơn vị triệu đồng).

Cài đặt một cấu trúc **danh sách liên kết đôi**, với các thao tác: 1) *Khởi tạo danh sách*; 2) *Tạo mới một nút chứa dữ liệu*; 3) *Chèn nút vào cuối danh sách*; 4) *Tìm kiếm phần tử trong danh sách*; 5) *Xóa nút trong danh sách*; 6) *Duyệt danh sách*.

Chương trình chính: Sử dụng danh sách liên kết đôi và các thao tác trên để:

- Nhập vào một danh sách gồm n Vận động viên (n bất kỳ).
- Hiển thị danh sách đã nhập ra màn hình.
- Liệt kê ra màn hình danh sách những Vận động viên quê ở “Nghệ An”.
- Xóa khỏi danh sách tất cả những vận động viên đã hết tuổi thi đấu (từ 50 tuổi trở lên).

Bài 35:Viết chương trình thực hiện yêu cầu sau: Khai báo cấu trúc **Môn học** gồm các thông tin: *Mã môn học, tên môn học, số tín chỉ, khoa quản lý, kỳ học*.

```
struct MonHoc
{
    char ID[6];
    char ten[50];
    int so_tc;
    char khoa_ql;
    int hoc_ky;
};
```

Viết các hàm nhập vào từ bàn phím giá trị của một biến cấu trúc **MonHoc**, hiển thị lên màn hình.

Nhập vào 1 số nguyên dương **n**, sử dụng cấu trúc danh sách liên kết đôi, nhập vào danh sách gồm **n** môn học. Hiển thị danh sách lên màn hình.

Hiển thị lên màn hình thông tin đầy đủ của tất cả những môn học do khoa “Công nghệ Thông tin” quản lý.

Sắp xếp danh sách môn học theo học kỳ (tăng dần), hiển thị danh sách sau khi sắp ra màn hình.

Bài 36:

Viết chương trình thực hiện yêu cầu sau: Khai báo cấu trúc **date** gồm các thông tin: *ngày, tháng, năm*.

```
struct date
{
    int ngay, thang, nam;
};
```

Khai báo cấu trúc **Công nhân** gồm các thông tin: *Mã công nhân, tên công nhân, ngày sinh, quê quán, lương*.

```
struct CongNhan
{
    char ID[6];
    char ten[50];
    date ngay_sinh;
    char que_quan[100];
    float lương;
};
```

Viết các hàm nhập vào từ bàn phím giá trị của một biến cấu trúc **CongNhan**, hiển thị lên màn hình.

Nhập vào 1 số nguyên dương **n**, sử dụng cấu trúc danh sách (dùng cài đặt mảng – danh sách kê hoặc cài đặt con trỏ-danh sách liên kết), nhập vào danh sách gồm **n** công nhân. Hiển thị danh sách lên màn hình.

Sắp xếp danh sách trên theo tên công nhân (thứ tự alphabet), hiển thị danh sách sau khi sắp ra màn hình.

Xóa khỏi danh sách tất cả những công nhân sinh trước năm 1960, hiển thị danh sách sau khi xóa lên màn hình.