

## 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
- + Cây nhị phân,

### BÀI TẬP

#### DSLK ĐƠN

**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 trình 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 trình 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:
  - Nhập vào 1 DSLK đơn chứa n số nguyên.
  - Tìm phần tử dương nhỏ nhất và phần tử âm lớn nhất trong danh sách.
  - 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?
  - Xóa khỏi danh sách tất cả các nút chứa giá trị x đã được tìm thấy ở bước trên.

## **DANH SÁCH LIÊN KẾT ĐÔI**

### **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

## **NGĂN XẾP, HÀNG ĐỢI**

### **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  $\rightarrow$  BT sai.
- + Nếu duyệt hết biểu thức và ngăn xếp không rỗng  $\rightarrow$  BT sai
- + Nếu duyệt chưa hết biểu thức và ngăn xếp rỗng  $\rightarrow$  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  $\rightarrow$  Lấy phần dư theo chiều ngược lại.

### **CÂY NHI PHÂN**

#### **Bài 17) Viết chương trình thực hiện các yêu cầu sau:**

- Định nghĩa một cấu trúc cây tìm kiếm nhị phân, với các thao tác: 1) *Khởi tạo cây*, 2) *Tạo mới một nút trên cây*; 3) *Chèn nút vào cây*; 4) *Tìm kiếm một nút trên cây*; 5) *Duyệt cây*.
- Chương trình chính: Sử dụng các thao tác đã cài đặt để:
  - + Nhập N số nguyên từ bàn phím và lưu vào cây (2 điểm).
  - + Duyệt cây theo thứ tự trước (Node-Left-Right) và hiển thị giá trị và vị trí của các nút trên cây ra màn hình.
  - + Nhập vào 1 số nguyên x, tìm kiếm xem nút có giá trị x đã được lưu trên cây? Nếu tìm thấy hiển thị ra màn hình vị trí của nút vừa tìm được. Ngược lại, thông báo không tìm thấy nút.

### **Bài 18) Viết chương trình thực hiện các yêu cầu sau:**

- Định nghĩa một cấu trúc cây tìm kiếm nhị phân, với các thao tác: 1) *Khởi tạo cây*, 2) *Tạo mới một nút trên cây*; 3) *Chèn nút vào cây*; 4) *Tìm kiếm một nút trên cây*; 5) *Duyệt cây*.
- Chương trình chính: Sử dụng các thao tác đã cài đặt để:
  - + Nhập **n** ký tự từ bàn phím và lưu vào cây.
  - + Duyệt cây theo thứ tự giữa (Left- Node-Right), hiển thị giá trị và vị trí của các nút trên cây ra màn hình.
  - + Nhập vào 1 ký tự **c** bất kỳ, tìm kiếm xem nút có chứa ký tự **c** đã được lưu trên cây? Nếu tìm thấy hiển thị ra màn hình vị trí của nút vừa tìm được. Ngược lại, thông báo không tìm thấy nút.

### **Bài 19) Viết chương trình thực hiện các yêu cầu sau:**

Viết chương trình trong ngôn ngữ C/C++ thực hiện các công việc sau:

- Định nghĩa một cấu trúc **Nhân viên**, bao gồm các trường thông tin (*Mã nhân viên, Họ tên, Địa chỉ, Số ĐT, Bậc lương, Ngày công*).
- Định nghĩa một cấu trúc cây tìm kiếm nhị phân, với các thao tác cơ bản như: 1 - Khởi tạo cây, 2 - Chèn nút vào cây, 3 - Tìm kiếm một nút trên cây, 4 - Duyệt cây, 5 - Xóa một nút khỏi cây.
- Chương trình chính: Sử dụng các thao tác đã cài đặt để:
  - + Nhập thông tin của N Nhân viên từ bàn phím và lưu vào cấu trúc cây tìm kiếm nhị phân (lấy trường Mã nhân viên làm khóa).
  - + Nhập 1 Mã nhân viên bất kỳ từ bàn phím, tìm xem nút trên cây có giá trị khóa trùng với mã nhân viên vừa nhập, nếu tìm thấy xóa nút đó khỏi cây.
  - + Hiển thị thông tin lưu trữ trên cây sau khi đã xóa nút.

**Bài 20) Viết chương trình trong ngôn ngữ C/C++ thực hiện các công việc sau:**

- Định nghĩa một cấu trúc cây tìm kiếm nhị phân, với các thao tác cơ bản như: 1 - Khởi tạo cây, 2 - Chèn nút vào cây, 3 - Tìm kiếm một nút trên cây, 4 - Duyệt cây, 5 - Xóa một nút khỏi cây.
- Chương trình chính: Sử dụng các thao tác đã cài đặt để:
  - + Nhập N số nguyên từ bàn phím và lưu vào cây.
  - + Duyệt cây theo thứ tự trước (Node-Left-Right) và hiển thị ra màn hình những giá trị được lưu trên cây.
  - + Nhập một số nguyên bất kỳ từ bàn phím. Tìm xem giá trị đó đã được lưu trữ trên cây? Nếu tìm được, xóa nút tương ứng khỏi cây. Hiển thị lại thông tin sau khi đã XÓA.