

Bài 1

Viết chương trình xây dựng và quản lý danh sách liên kết kép. Thành phần quản lý gồm con trỏ pHead, pTail. Thành phần dữ liệu trong mỗi node là thông tin một sinh viên, bao gồm các trường:

- Mã sinh viên (int)
- Họ tên sinh viên (string)
- Lớp (string)
- Điểm toán (float)
- Điểm lý (float)
- Điểm hóa (float)

Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiển thị toàn bộ danh sách
2. Tìm một sinh viên theo mã sinh viên (nhập vào)
3. Thêm một sinh viên vào cuối danh sách. Đảm bảo không có 2 sinh viên nào trùng mã.
4. Xóa một sinh viên khỏi danh sách từ mã sinh viên (nhập vào)
5. Nhập một lớp. Hiển thị danh sách sinh viên thuộc về lớp đó
6. Tính tổng số sinh viên có điểm toán ≥ 5
7. Hiển thị toàn bộ danh sách sinh viên chứa tên nhập vào. (Ví dụ: nhập tên cần tìm là Khuong, hiển thị toàn bộ sinh viên chứa tên Khuong)
8. Sắp xếp danh sách tăng dần theo mã sinh viên
9. Sắp xếp danh sách tăng dần theo điểm toán, rồi đến điểm lý, rồi đến điểm hóa.
10. Nhập một lớp. Hủy toàn bộ các sinh viên thuộc về lớp đó.

Bài 2

Viết chương trình xây dựng và quản lý danh sách liên kết vòng. Thành phần quản lý gồm con trỏ pList trỏ vào cuối danh sách. Thành phần dữ liệu trong mỗi node là thông tin một mặt hàng, bao gồm các trường:

- Mã mặt hàng (int)
- Tên mặt hàng (string)
- Đơn vị tính (string)
- Trọng lượng (float)
- Đơn giá (float)

Hiện thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiện thị toàn bộ danh sách
2. Tìm một mặt hàng theo mã mặt hàng (nhập vào)
3. Thêm một mặt hàng vào đầu danh sách. Đảm bảo không có 2 mặt hàng nào trùng mã.
4. Xóa một mặt hàng khỏi danh sách từ mã mặt hàng (nhập vào)
5. Nhập một đơn vị tính. Hiện thị danh sách mặt hàng có đơn vị tính đó.
6. Đếm số mặt hàng có đơn vị tính là “hộp”
7. Sắp xếp danh sách tăng dần theo đơn giá.
8. Sau đó, khi thêm một mặt hàng mới vào, đúng vị trí sao cho danh sách mặt hàng vẫn thỏa điều kiện tăng dần theo đơn giá.
9. Hủy toàn bộ danh sách
10. Lưu trữ danh sách sinh viên này vào file text

Bài 3

Viết chương trình xây dựng và quản lý danh sách liên kết đơn. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin một sinh viên, bao gồm các trường:

- Mã sách (int)
- Tên sách (string)
- Tác giả (string)
- Lần xuất bản (int)
- Số trang (int)
- Giá thành (float)

Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiển thị toàn bộ danh sách
2. Tìm một mặt hàng theo mã sách (nhập vào)
3. Thêm một cuốn sách vào cuối danh sách. Đảm bảo không có 2 cuốn nào trùng mã.
4. Xóa một cuốn sách khỏi danh sách biết mã sách (nhập vào)
5. Nhập một tác giả. Hiển thị danh sách các cuốn sách của tác giả đó.
6. Đếm số cuốn sách có lần xuất bản là 3.
7. Sắp xếp danh sách tăng dần theo Giá thành.
8. Sau đó, khi thêm một sách mới vào, đúng vị trí sao cho danh sách sách vẫn thỏa điều kiện tăng dần theo Giá thành.
9. Xóa một cuốn sách ở cuối danh sách.
10. Hủy toàn bộ danh sách

Bài 4

Viết chương trình xây dựng và quản lý danh sách liên kết kép. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin của một hộ dân trong một phường, bao gồm các trường:

- Mã hộ (int)
- Tên chủ hộ (string)
- Số thành viên (int)
- Mức thu nhập (float)

Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiển thị toàn bộ danh sách
2. Tìm một hộ theo Tên chủ hộ (nhập vào)
3. Thêm một hộ vào cuối danh sách. Đảm bảo không có 2 hộ nào trùng mã.
4. Xóa một hộ khỏi danh sách từ mã hộ (nhập vào)
5. Nhập số thành viên. Hiển thị danh sách hộ có số thành viên đó.
6. Đếm số hộ có mức thu nhập trên 2 triệu.
7. Sắp xếp danh sách giảm dần theo số thành viên.
8. Sau đó, khi thêm một hộ mới vào, đúng vị trí sao cho danh sách các hộ vẫn thỏa điều kiện giảm dần theo số thành viên.
9. Tìm kiếm danh sách các hộ có mức thu nhập dưới 10 triệu.
10. Hủy toàn bộ danh sách

Bài 5

Viết chương trình xây dựng và quản lý danh sách liên kết đơn. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin một sinh viên, bao gồm các trường:

- Mã sinh viên (int)
- Họ tên sinh viên (string)
- Lớp (string)
- Điểm toán (float)
- Điểm lý (float)
- Điểm hóa (float)

Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiển thị toàn bộ danh sách
2. Tìm một sinh viên theo mã sinh viên (nhập vào)
3. Thêm một sinh viên vào đầu danh sách. Đảm bảo không có 2 sinh viên nào trùng mã.
4. Xóa một sinh viên khỏi danh sách từ mã sinh viên (nhập vào)
5. Tính tổng số sinh viên có điểm toán ≥ 5
6. Hiển thị toàn bộ danh sách sinh viên chứa tên nhập vào. (Ví dụ: nhập tên cần tìm là Khuong, hiển thị toàn bộ sinh viên chứa tên Khuong)
7. Hiển thị danh sách sinh viên yếu (có điểm trung bình ≤ 4)
8. Sắp xếp danh sách tăng dần theo điểm trung bình
9. Sắp xếp danh sách tăng dần theo điểm toán, rồi đến điểm lý, rồi đến điểm hóa.
10. Nhập một lớp. Hủy toàn bộ các sinh viên thuộc về lớp đó.

Bài 6

Viết chương trình xây dựng và quản lý danh sách liên kết đôi vòng. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin của một cầu thủ của một đội bóng, bao gồm các trường:

- Mã cầu thủ (int)
- Tên cầu thủ (string)
- Số bàn thắng (int)
- Thưởng (float)

Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiển thị toàn bộ danh sách
2. Tìm một cầu thủ theo Tên cầu thủ (nhập vào)
3. Thêm một cầu thủ vào cuối danh sách. Đảm bảo không có 2 cầu thủ nào trùng mã.
4. Xóa một cầu thủ khỏi danh sách từ mã cầu thủ (nhập vào)
5. Nhập số bàn thắng. Hiển thị danh sách cầu thủ có số bàn thắng đó.
6. Đếm số cầu thủ có thưởng trên 1 triệu.
7. Sắp xếp danh sách tăng dần theo số bàn thắng.
8. Sau đó, khi thêm một cầu thủ mới vào, đúng vị trí sao cho danh sách các cầu thủ vẫn thỏa điều kiện tăng dần theo số bàn thắng.
9. Tìm kiếm danh sách các hộ có mức thưởng dưới 3 triệu.
10. Hủy toàn bộ danh sách

Bài 7

Viết chương trình xây dựng và quản lý danh sách liên kết đơn. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin của một vật tư bao gồm các trường:

- Mã vật tư (string)
- Tên vật tư (string)
- Số lượng (int)
- Giá (float)

Hiện thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiện thị toàn bộ danh sách
2. Tìm một vật tư theo Tên vật tư (nhập vào)
3. Thêm một vật tư vào đầu danh sách. Đảm bảo không có 2 vật tư nào trùng mã.
4. Xóa một vật tư khỏi danh sách từ mã vật tư (nhập vào)
5. Nhập số lượng. Hiện thị danh sách vật tư có số lượng đó.
6. Đếm số vật tư có giá 50000.
7. Sắp xếp danh sách giảm dần theo số lượng.
8. Sau đó, khi thêm một vật tư mới vào, đúng vị trí sao cho danh sách các vật tư vẫn thỏa điều kiện giảm dần theo số lượng.
9. Tìm kiếm danh sách các vật tư có giá dưới 200000.
10. Hủy toàn bộ danh sách

Bài 8

Viết chương trình xây dựng và quản lý danh sách liên kết vòng. Thành phần quản lý gồm con trỏ pList trỏ vào cuối danh sách. Thành phần dữ liệu trong mỗi node là thông tin một khoa, bao gồm các trường:

- Mã khoa (string)
- Tên khoa (string)
- Trưởng khoa (string)
- Số GV (int)
- Địa chỉ (string)

Hiện thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiện thị toàn bộ danh sách
2. Tìm một khoa theo tên khoa(nhập vào)
3. Thêm một khoa vào đầu danh sách. Đảm bảo không có 2 khoa nào trùng mã.
4. Xóa một khoa khỏi danh sách từ mã khoa (nhập vào)
5. Nhập một tên trưởng khoa. Hiện thị danh sách khoa có trưởng khoa đó.
6. Đếm số mặt hàng có số GV là > 10 .
7. Sắp xếp danh sách giảm dần theo số GV.
8. Sau đó, khi thêm một khoa mới vào, đúng vị trí sao cho danh sách khoa vẫn thỏa điều kiện giảm dần theo số GV.
9. Hủy toàn bộ danh sách
10. Lưu trữ danh sách khoa này vào file text

Bài 9

Viết chương trình xây dựng và quản lý danh sách liên kết đơn. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin một môn học, bao gồm các trường:

- Mã môn học (string)
- Tên môn học (string)
- Khoa (string)
- Số tín chỉ (int)
- Giảng viên (string)

Hiện thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiện thị toàn bộ danh sách
2. Tìm một môn học theo tên môn học (nhập vào)
3. Thêm một môn học vào cuối danh sách. Đảm bảo không có 2 môn học nào trùng mã.
4. Xóa một môn học khỏi danh sách từ mã môn học (nhập vào)
5. Tính tổng số môn học có số tín chỉ ≥ 3
6. Hiện thị toàn bộ danh sách môn học chứa tên nhập vào. (Ví dụ: nhập tên cần tìm là Kế toán, hiện thị toàn bộ môn học của khoa Kế toán)
7. Hiện thị danh sách các môn học của khoa Cơ khí
8. Sắp xếp danh sách tăng dần theo số tín chỉ.
9. Sắp xếp danh sách tăng dần theo điểm toán, rồi đến điểm lý, rồi đến điểm hóa.
10. Nhập một khoa. Hủy toàn bộ các môn học thuộc về khoa đó.

Bài 10

Viết chương trình xây dựng và quản lý danh sách liên kết kép. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin một học sinh, bao gồm các trường:

- Mã học sinh (int)
- Tên học sinh (string)
- Giới tính (int)
- Tuổi (int)
- Điểm trung bình (float)

Hiện thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiện thị toàn bộ danh sách
2. Tìm một học sinh theo mã học sinh (nhập vào)
3. Thêm một học sinh vào cuối danh sách. Đảm bảo không có 2 học sinh nào trùng mã.
4. Xóa một học sinh khỏi danh sách từ mã học sinh (nhập vào)
5. Hiện thị danh sách học sinh nữ.
6. Đếm số học sinh có điểm trung bình > 8.0
7. Sắp xếp danh sách tăng dần theo điểm trung bình.
8. Sau đó, khi thêm một mặt hàng mới vào, đúng vị trí sao cho danh sách mặt hàng vẫn thỏa điều kiện tăng dần theo đơn giá.
9. Hủy toàn bộ danh sách
10. Lưu trữ danh sách sinh viên này vào file text

Bài 11

Viết chương trình quản lý danh sách lớp. Mỗi sinh viên gồm các thành phần:

- Mã SV: char[10];
- Mã lớp: int
- Tên SV: char[255];
- Điểm toán
- Điểm lý
- Điểm hóa

Mỗi lớp gồm các thông tin:

- Mã lớp: int
- Tên lớp: char[10];
- Khóa

Xây dựng và quản lý danh sách lớp sử dụng cây nhị phân tìm kiếm (Binary Search Tree).
Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm). Thành phần dữ liệu trong mỗi Node là giá trị kiểu integer.

1. Thêm 1 lớp mới.
2. Thêm một sinh viên
 - a. Nếu mã SV đã có thì hiển thị sinh viên đó ra màn hình, cùng với thông báo không thể nhập sinh viên đã có
 - b. Mã lớp phải tồn tại trong danh sách lớp. Nếu chưa có, phải hiện thông báo lỗi.
3. Tìm một sinh viên theo mã SV: Khi tìm thấy, hiển thị mã, tên, điểm, mã lớp và tên lớp.
4. Lưu danh sách sinh viên-lớp vào file
5. Tìm tất cả sinh viên theo tên nhập vào
6. Hiển thị tất cả sinh viên theo mã lớp nhập vào
7. Hiển thị tất cả sinh viên theo tên lớp nhập vào
8. Xóa một sinh viên ra khỏi danh sách
9. Xóa một lớp ra khỏi danh sách
10. Tìm tất cả sinh viên có điểm trung bình lớn nhất

Bài 12

Viết chương trình xây dựng và quản lý danh sách liên kết đôi vòng. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin một khách hàng, bao gồm các trường:

- Mã KH (string)
- Tên KH (string)
- Tuổi (int)
- Tỉnh (string)
- Điện thoại (string)

Hiện thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiện thị toàn bộ danh sách
2. Tìm một khách hàng theo mã khách hàng (nhập vào)
3. Thêm một khách hàng vào đầu danh sách. Đảm bảo không có 2 khách hàng nào trùng mã.
4. Xóa một khách hàng khỏi danh sách từ mã khách hàng (nhập vào)
5. Nhập một tỉnh. Hiện thị danh sách khách hàng ở tỉnh đó.
6. Đếm số khách hàng trên 50 tuổi.
7. Sắp xếp danh sách tăng dần theo tuổi.
8. Sau đó, khi thêm một khách hàng mới vào, đúng vị trí sao cho danh sách khách hàng vẫn thỏa điều kiện tăng dần theo tuổi.
9. Hủy toàn bộ danh sách
10. Lưu trữ danh sách khách hàng này vào file text.

Bài 13

Viết chương trình xây dựng và quản lý danh sách liên kết đơn. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin của một phân xưởng bao gồm các trường:

- Mã phân xưởng (string)
- Tên phân xưởng (string)
- Số nhân viên (int)
- Trưởng phân xưởng (string)

Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiển thị toàn bộ danh sách
2. Tìm một phân xưởng theo mã phân xưởng (nhập vào)
3. Thêm một phân xưởng vào cuối danh sách. Đảm bảo không có 2 phân xưởng nào trùng mã.
4. Xóa một phân xưởng khỏi danh sách từ mã phân xưởng (nhập vào)
5. Nhập số nhân viên. Hiển thị danh sách phân xưởng có số nhân viên > số đó.
6. Đếm số phân xưởng có số nhân viên <10.
7. Sắp xếp danh sách tăng dần theo số nhân viên.
8. Sau đó, khi thêm một phân xưởng mới vào, đúng vị trí sao cho danh sách các phân xưởng vẫn thỏa điều kiện tăng dần theo số nhân viên.
9. Tìm kiếm danh sách các phân xưởng có số nhân viên ≥ 15 .
10. Hủy toàn bộ danh sách

Bài 14

Viết chương trình xây dựng và quản lý danh sách liên kết vòng. Thành phần quản lý gồm con trỏ pList trỏ vào cuối danh sách. Thành phần dữ liệu trong mỗi node là thông tin một sinh viên, bao gồm các trường:

- Mã sách (int)
- Tên sách (string)
- Tác giả (string)
- Lần xuất bản (int)
- Số trang (int)
- Giá thành (float)

Hiện thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiện thị toàn bộ danh sách
2. Tìm một cuốn sách theo mã sách (nhập vào)
3. Thêm một cuốn sách vào đầu danh sách. Đảm bảo không có 2 sách nào trùng mã.
4. Xóa một cuốn sách khỏi danh sách từ mã sách (nhập vào)
5. Hiện thị danh sách xuất bản lần thứ 3.
6. Đếm số sách có số trang ≥ 100
7. Sắp xếp danh sách tăng dần theo số trang.
8. Sau đó, khi thêm một sách mới vào, đúng vị trí sao cho danh sách các sách vẫn thỏa điều kiện tăng dần theo số trang.
9. Hủy toàn bộ danh sách
10. Lưu trữ danh sách cuốn sách này vào file text.

Bài 15

Viết chương trình xây dựng và quản lý danh sách liên kết kép. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin một môn học, bao gồm các trường:

- Mã môn học (string)
- Tên môn học (string)
- Khoa (string)
- Số tín chỉ (int)
- Giảng viên (string)

Hiện thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiện thị toàn bộ danh sách
2. Tìm một môn học theo tên môn học (nhập vào)
3. Thêm một môn học vào cuối danh sách. Đảm bảo không có 2 môn học nào trùng mã.
4. Xóa một môn học khỏi danh sách từ mã môn học (nhập vào)
5. Tính tổng số môn học có số tín chỉ ≥ 3
6. Hiện thị toàn bộ danh sách môn học chứa tên nhập vào. (Ví dụ: nhập tên cần tìm là Kế toán, hiện thị toàn bộ môn học của khoa Kế toán)
7. Hiện thị danh sách các môn học của khoa Cơ khí
8. Sắp xếp danh sách tăng dần theo số tín chỉ.
9. Sắp xếp danh sách tăng dần theo điểm toán, rồi đến điểm lý, rồi đến điểm hóa.
10. Nhập một khoa. Hủy toàn bộ các môn học thuộc về khoa đó.

Bài 16

Viết chương trình xây dựng và quản lý danh sách liên kết vòng. Thành phần quản lý gồm con trỏ pList trỏ vào cuối danh sách. Thành phần dữ liệu trong mỗi node là thông tin một của một hộ dân trong một phường, bao gồm các trường:

- Mã hộ (int)
- Tên chủ hộ (string)
- Số thành viên (int)
- Mức thu nhập (float)

Hiện thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiện thị toàn bộ danh sách
2. Tìm một hộ theo Tên chủ hộ (nhập vào)
3. Thêm một hộ vào cuối danh sách. Đảm bảo không có 2 hộ nào trùng mã.
4. Xóa một hộ khỏi danh sách từ mã hộ (nhập vào)
5. Nhập số thành viên. Hiện thị danh sách hộ có số thành viên đó.
6. Đếm số hộ có mức thu nhập trên 2 triệu.
7. Sắp xếp danh sách giảm dần theo số thành viên.
8. Sau đó, khi thêm một hộ mới vào, đúng vị trí sao cho danh sách các hộ vẫn thỏa điều kiện giảm dần theo số thành viên.
9. Tìm kiếm danh sách các hộ có mức thu nhập dưới 10 triệu.
10. Hủy toàn bộ danh sách

Bài 17

Viết chương trình xây dựng và quản lý danh sách liên kết đơn. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin một mặt hàng, bao gồm các trường:

- Mã mặt hàng (int)
- Tên mặt hàng (string)
- Đơn vị tính (string)
- Trọng lượng (float)
- Đơn giá (float)

Hiện thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiện thị toàn bộ danh sách
2. Tìm một mặt hàng theo mã mặt hàng (nhập vào)
3. Thêm một mặt hàng vào đầu danh sách. Đảm bảo không có 2 mặt hàng nào trùng mã.
4. Xóa một mặt hàng khỏi danh sách từ mã mặt hàng (nhập vào)
5. Nhập một đơn vị tính. Hiện thị danh sách mặt hàng có đơn vị tính đó.
6. Đếm số mặt hàng có đơn vị tính là “hộp”
7. Sắp xếp danh sách tăng dần theo đơn giá.
8. Sau đó, khi thêm một mặt hàng mới vào, đúng vị trí sao cho danh sách mặt hàng vẫn thỏa điều kiện tăng dần theo đơn giá.
9. Hủy toàn bộ danh sách
10. Lưu trữ danh sách sinh viên này vào file text

Bài 18

Viết chương trình xây dựng và quản lý danh sách liên kết vòng. Thành phần quản lý gồm con trỏ pList trỏ vào cuối danh sách. Thành phần dữ liệu trong mỗi node là thông tin một của một nhân viên trong một phòng, bao gồm các trường:

- Mã nhân viên (int)
- Tên nhân viên (string)
- Phòng (string)
- Giới tính (int)
- Địa chỉ (string)
- Lương (float)

Hiện thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiện thị toàn bộ danh sách
2. Tìm một nhân viên theo mã nhân viên (nhập vào)
3. Thêm một nhân viên vào đầu danh sách. Đảm bảo không có 2 nhân viên nào trùng mã.
4. Xóa một nhân viên khỏi danh sách từ mã nhân viên (nhập vào)
5. Nhập một phòng. Hiện thị danh sách nhân viên thuộc về phòng đó
6. Tính tổng số nhân viên nữ.
7. Hiện thị toàn bộ danh sách nhân viên ở địa chỉ nhập vào.
8. Sắp xếp danh sách tăng dần theo tên nhân viên
9. Sắp xếp danh sách tăng dần theo điểm toán, rồi đến điểm lý, rồi đến điểm hóa.
10. Nhập một lớp. Hủy toàn bộ các sinh viên thuộc về lớp đó.

Bài 19

Viết chương trình xây dựng và quản lý danh sách liên kết vòng. Thành phần quản lý gồm con trỏ pHead, pTail. Thành phần dữ liệu trong mỗi node là thông tin một sinh viên, bao gồm các trường:

- Mã sinh viên (int)
- Họ tên sinh viên (string)
- Lớp (string)
- Điểm toán (float)
- Điểm lý (float)
- Điểm hóa (float)

Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

1. Hiển thị toàn bộ danh sách
2. Thêm một sinh viên vào cuối danh sách. Đảm bảo không có 2 sinh viên nào trùng mã.
3. Xóa một sinh viên khỏi danh sách từ mã sinh viên (nhập vào)
4. Nhập một lớp. Hiển thị danh sách sinh viên thuộc về lớp đó
5. Tính tổng số sinh viên có điểm toán ≥ 5
6. Hiển thị danh sách sinh viên giỏi (có điểm trung bình ≥ 8 và không có môn học nào ≤ 6)
7. Sắp xếp danh sách tăng dần theo mã sinh viên
8. Sắp xếp danh sách tăng dần theo từng lớp, trong mỗi lớp tăng dần theo điểm trung bình
9. Nhập một lớp. Hủy toàn bộ các sinh viên thuộc về lớp đó.
10. Hủy tất cả sinh viên có học lực kém (điểm trung bình ≤ 3).

Bài 20

Viết chương trình xây dựng và quản lý danh sách liên kết kép. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin của một phân xưởng bao gồm các trường:

- Mã phân xưởng (string)
- Tên phân xưởng (string)
- Số nhân viên (int)
- Trưởng phân xưởng (string)

Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

11. Hiển thị toàn bộ danh sách
12. Tìm một phân xưởng theo mã phân xưởng (nhập vào)
13. Thêm một phân xưởng vào cuối danh sách. Đảm bảo không có 2 phân xưởng nào trùng mã.
14. Xóa một phân xưởng khỏi danh sách từ mã phân xưởng (nhập vào)
15. Nhập số nhân viên. Hiển thị danh sách phân xưởng có số nhân viên > số đó.
16. Đếm số phân xưởng có số nhân viên < 15.
17. Sắp xếp danh sách giảm dần theo số nhân viên.
18. Sau đó, khi thêm một phân xưởng mới vào, đúng vị trí sao cho danh sách các phân xưởng vẫn thỏa điều kiện giảm dần theo số nhân viên.
19. Tìm kiếm danh sách các phân xưởng có số nhân viên ≥ 10 .
20. Hủy toàn bộ danh sách

Bài 21

Viết chương trình xây dựng và quản lý danh sách liên kết vòng. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin một mặt hàng, bao gồm các trường:

- Mã mặt hàng (int)
- Tên mặt hàng (string)
- Đơn vị tính (string)
- Trọng lượng (float)
- Đơn giá (float)

Xây dựng và quản lý danh sách mặt hàng sử dụng cây nhị phân tìm kiếm (Binary Search Tree). Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm). Thành phần dữ liệu trong mỗi Node là giá trị kiểu integer.

11. Hiển thị toàn bộ danh sách
12. Tìm một mặt hàng theo mã mặt hàng (nhập vào)
13. Thêm một mặt hàng vào đầu danh sách. Đảm bảo không có 2 mặt hàng nào trùng mã.
14. Xóa một mặt hàng khỏi danh sách từ mã mặt hàng (nhập vào)
15. Nhập một đơn vị tính. Hiển thị danh sách mặt hàng có đơn vị tính đó.
16. Đếm số mặt hàng có đơn vị tính là “hộp”
17. Sắp xếp danh sách tăng dần theo đơn giá.
18. Sau đó, khi thêm một mặt hàng mới vào, đúng vị trí sao cho danh sách mặt hàng vẫn thỏa điều kiện tăng dần theo đơn giá.
19. Hủy toàn bộ danh sách
20. Lưu trữ danh sách sinh viên này vào file text

Bài 22

Viết chương trình xây dựng và quản lý danh sách liên kết kép. Thành phần quản lý gồm con trỏ pHead trỏ vào đầu danh sách. Thành phần dữ liệu trong mỗi node là thông tin một sinh viên, bao gồm các trường:

- Mã sách (int)
- Tên sách (string)
- Tác giả (string)
- Lần xuất bản (int)
- Số trang (int)
- Giá thành (float)

Hiện thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

11. Hiện thị toàn bộ danh sách
12. Tìm một mặt hàng theo mã sách (nhập vào)
13. Thêm một cuốn sách vào cuối danh sách. Đảm bảo không có 2 cuốn nào trùng mã.
14. Xóa một cuốn sách khỏi danh sách biết mã sách (nhập vào)
15. Nhập một tác giả. Hiện thị danh sách các cuốn sách của tác giả đó.
16. Đếm số cuốn sách có lần xuất bản là 2.
17. Sắp xếp danh sách giảm dần theo Giá thành.
18. Sau đó, khi thêm một sách mới vào, đúng vị trí sao cho danh sách vẫn thỏa điều kiện giảm dần theo Giá thành.
19. Xóa một cuốn sách ở cuối danh sách.
20. Hủy toàn bộ danh sách

Bài 23

Viết chương trình quản lý danh sách lớp. Mỗi sinh viên gồm các thành phần:

- Mã SV: char[10];
- Mã lớp: int
- Tên SV: char[255];
- Điểm toán
- Điểm lý
- Điểm hóa

Mỗi lớp gồm các thông tin:

- Mã lớp: int
- Tên lớp: char[10];
- Khóa

Xây dựng và quản lý danh sách lớp sử dụng cây nhị phân tìm kiếm (Binary Search Tree).
Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm). Thành phần dữ liệu trong mỗi Node là giá trị kiểu integer.

1. Thêm 1 lớp mới.
2. Thêm một sinh viên
 - a. Nếu mã SV đã có thì hiển thị sinh viên đó ra màn hình, cùng với thông báo không thể nhập sinh viên đã có
 - b. Mã lớp phải tồn tại trong danh sách lớp. Nếu chưa có, phải hiện thông báo lỗi.
3. Tìm một sinh viên theo mã SV: Khi tìm thấy, hiển thị mã, tên, điểm, mã lớp và tên lớp.
4. Lưu danh sách sinh viên-lớp vào file
5. Đọc danh sách sinh viên từ file.
6. Hiển thị danh sách sinh viên tăng dần theo mã SV
7. Tìm tất cả sinh viên theo tên nhập vào

8. Hiển thị tất cả sinh viên theo tên lớp nhập vào
9. Xóa một sinh viên ra khỏi danh sách
10. Tìm tất cả sinh viên có điểm trung bình lớn nhất