

# Kỹ Thuật Lập Trình

## Bài kiểm tra thực hành cuối kỳ

Thời gian: 90 phút

Câu 1: **(3.5đ)**

- (0.5đ)** Đề xuất một cấu trúc dữ liệu để lưu một đa thức có bậc không âm (bậc 0, 1, 2, ...).
- (1đ)** Xây dựng hàm nhận vào hai đa thức (P1 và P2) có bậc M và N và trả về một đa thức (P3). Hàm nhân hai đa thức P1 và P2 để tạo ra P3, và trả về P3.
- (1đ)** Xây dựng hàm nhận vào một đa thức. Hàm này in ra đa thức theo định dạng như ví dụ sau:

$$P(x) = 3x^4 + 2x + 1$$

(Nghĩa là không in ra các thành phần có hệ số bằng 0, bậc in ra giảm dần)

- (1đ)** Xây dựng chương trình, hàm main:
  - Cho phép người dùng nhập vào giá trị M và N (giả sử M và N luôn hợp lệ: số nguyên  $> 0$  và  $< 5$ , sinh viên không cần kiểm tra M và N). Khai báo và cấp phát động hai đa thức.
  - Thực hiện việc gán giá trị cho tất cả các hệ số trong 2 đa thức bằng cách sinh ngẫu nhiên.
  - Gọi các hàm ở câu (b) và (c) và in ra điểm kết quả để kiểm tra tính đúng đắn của chúng
  - Giải phóng tất cả vùng nhớ đã cấp phát động.

Câu 2: **(6.5đ)**

Để hỗ trợ hiện thực một hệ thống quản lý khách hàng (*CRM – customer relation management*), chúng ta cần xây dựng một struct **Customer** và một struct **CRM** với mục đích như sau:

- **Customer**: lưu trữ thông tin, những thuộc tính để thao tác với các khách hàng như sau:
  - Khách hàng có thể là cá nhân hoặc đại diện cho một cơ quan, doanh nghiệp nào đó.
  - Trường hợp khách hàng mang tính chất cá nhân, cần lưu những thông tin cá nhân sau: mã khách hàng, họ và tên, và các thông tin bổ sung như tên viết tắt (biệt danh), địa chỉ, ngày sinh (bao gồm ngày tháng

- và năm), mã dịch vụ (kiểu số nguyên, ở đây chúng ta không cần quan tâm đến tên cụ thể của dịch vụ), số lần sử dụng dịch vụ, chi phí và thời gian (ngày, giờ) mỗi lần sử dụng dịch vụ
- Trường hợp khách hàng đại diện cho doanh nghiệp, ngoài thông tin cá nhân cần lưu thêm thông tin về doanh nghiệp như: mã số thuế, tên doanh nghiệp, địa chỉ, điện thoại, fax, email, địa chỉ website doanh nghiệp
  - Loại thẻ của khách hàng (thẻ vàng, thẻ bạc, thẻ thông thường)
- **CRM**: quản lý danh sách tất cả khách hàng (nghĩa là chứa một mảng các đối tượng thuộc lớp **Customer**)
- a. **(1đ)** Hãy khai báo các thuộc tính cần thiết như đặc tả bên trên của hai struct **CRM** và **Customer**.
  - a) **(1,5đ)** Viết một hàm hỗ trợ thêm một khách hàng mới, hãy dùng vòng lặp **do..while** để kiểm tra xem đã có chứa thông tin của khách hàng này chưa; nếu có sẵn thông tin thì không cần thêm vào, trường hợp ngược lại thì thêm vào cuối danh sách.
  - b) **(1,5đ)** Xây dựng một hàm cho phép tìm kiếm một khách hàng theo tên của khách hàng và trả về thông tin tương ứng (trả về **NULL** nếu không tìm thấy)
    - Hãy sử dụng vòng lặp **while(){ ...}** để hiện thực
  - c) **(1,5đ)** Xây dựng hàm cho phép truy xuất và sửa chữa thông tin của khách hàng: thay đổi (nâng cấp) thẻ của một khách hàng (thẻ vàng, thẻ bạc hay thẻ thông thường) và làm thế nào để xác định một khách hàng là thuộc dạng cá nhân hay là doanh nghiệp.
  - d) **(1đ)** Viết hàm **main**, tạo một struct **CRM** và nhập vào khoảng 5 khách hàng cá nhân và 5 doanh nghiệp; sau đó, xuất ra các thông tin thống kê cho sẵn.

**-HẾT-**