



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 (dùng struct hoặc class) để lưu một đa thức có bậc không âm (bậc 0, 1, 2, ...). Gợi ý: có thể dùng mảng để lưu các hệ số, kể cả khi hệ số đó bằng 0.
- (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: (1.5đ) – PHẢI viết bằng đệ quy

Xây dựng một hàm nhận vào hai tham số: (a) con trỏ đến phần tử đầu tiên của một mảng các số nguyên và (b) số lượng phần tử có trong mảng. Hàm này in ra các giá trị không âm trên mảng đầu vào theo thứ tự từ phần tử đầu đến phần tử cuối.

Ví dụ: mảng: -1, 3, -5, 15, 7 (có 5 phần tử): hàm in ra là: 3, 15, 7

Câu 3: (1.5đ) – PHẢI viết bằng đệ quy

Xây dựng một hàm tính tổng và in ra kết quả của các cặp phần tử nằm ở vị trí đối xứng nhau trên mảng 1 chiều, như ví dụ sau. Sinh viên tự đề xuất tên hàm và danh



sách các thông số. Khi gọi hàm này và truyền mảng và các thông số cần thiết vào thì hàm sẽ tính tổng và in ra.

Ví dụ:

Mảng vào: 1,5,7,8 (4 phần tử). Hàm tính và in ra: 9, 12. Lưu ý: $9 = 1 + 8$; $12 = 5 + 7$.

Mảng vào: 1,5, 4, 7,8 (4 phần tử). Hàm tính và in ra: 9, 12, 4. Lưu ý: $9 = 1 + 8$; $12 = 5 + 7$, 4 không cặp với phần tử nào nên kết quả vẫn là 4.

Câu 4:

- (0.5đ)** Đề xuất cấu trúc dữ liệu để lưu được một mảng 2 chiều (dùng struct hoặc class), gọi tên kiểu này là Matrix.
- (1đ)** Xây dựng một hàm nhận vào một mảng 2 chiều, gọi là **A**, có N hàng và M cột (N và M bất kỳ). Sinh viên có thể truyền số hàng, số cột cùng với con trỏ đến phần tử đầu tiên hoặc chỉ truyền vào một đối tượng Matrix – tùy cách dùng class hay struct. Hàm tính và trả về một ma trận có 1 hàng và M cột, gọi là **B**. Phần tử ở cột thứ i của ma trận B sẽ là tổng các phần tử phần ở cột thứ i của ma trận A.

Ví dụ:

A = 1, 2

3, 4

5, 6

B = 9, 12

(Lưu ý: $9 = 1 + 3 + 5$; $12 = 2 + 4 + 6$)

- (0.5đ)** Viết chương trình, hàm main:
 - Khai báo và khởi động một mảng 2 chiều có 4 hàng và 3 cột, các phần tử trong khởi động có giá trị tùy ý.
 - Gọi hàm ở câu (b) để tính ra ma trận kết quả; in ra ma trận kết quả.
 - Giải phóng các vùng nhớ đã xin được.

Câu 5: (1.5đ)

Xây dựng lớp (class) Date cho phép người lập trình thực hiện công việc như đoạn chương trình sau:

```
int main() {  
    Date d;  
    std::cin >> d; //nhập: 14 7 2017 và gõ ENTER
```



```
std::cout << d; //Xuất: 14/07/2017  
return 0;  
}
```

Đoạn chương trình trên cho phép nhập và xuất như trong chú thích
(Lưu ý: toán tử >> và << và định dạng đầu vào-ra).

---Hết---