CÁC BÀI THỰC HÀNH 4
(HÀM VÀ MẢNG 1 CHIỀU)

(Sinh viên bắt buộc làm 7 bài trở lên. Bài do sinh viên tự chọn)

Bài 1: Làm lại các bài tập phần câu lệnh điều khiển rẽ nhánh bằng cách viết:

a). Hàm đổi một ký tự hoa sang ký tự thường;

b). Thủ tục giải phương trình bậc nhất;

c). Thủ tục giải phương trình bậc hai;

d). Hàm xác định giá trị nhỏ nhất của 4 số nguyên;

e). Thủ tục hoán vị hai số nguyên;

f). Thủ tục sắp xếp 4 số nguyên tăng dần.

Với mỗi câu trên, viết hàm main() tương ứng để sử dụng các hàm đó.

Bài 2: Làm lại các bài tập phần câu lệnh lặp bằng cách viết hàm nhận vào một số nguyên dương n rồi thực hiện:

a). Cho ra (trả về) số đảo của số đó;

b). Xác định số đó có phải số đối xứng (cho ra trị true/false);

c). Xác định số đó có phải số chính phương;

d). Xác định số đó có phải số nguyên tố;

e). Tính tổng các chữ số lẻ;

f). Tính tổng các chữ số nguyên tố;

g). Tính tổng các chữ số chính phương;

h). Xác định số đó có phải số hoàn chỉnh;

i). Xác định số đó có phải số Armstrong.

Với mỗi câu trên, viết hàm main() tương ứng để sử dụng các hàm đó.

Bài 3: Làm lại các bài tập phần câu lệnh lặp bằng cách viết hàm nhận vào một số nguyên dương n rồi thực hiện:

a). Tính tổng S1 = 1 + 2 + … + n

b). Tính tổng S2 = $1^2 + 2^2 + \ldots + n^2$

c). Tính tổng S3 = 1 + 1/2 + … + 1/n

d). Tính tích P = n! = 1*2*…*n

e). Tính tổng S4 = 1! + 2! + … + n!

Viết hàm main() để sử dụng chung cho các hàm nói trên.

Bài 4: Viết hàm tính USCLN của hai số nguyên không âm.

Bài 5: Viết hàm in ra n phần tử đầu tiên của dãy Fibonacci.

Bài 6:

a). Viết định nghĩa cho hàm tongLe trong C/C++ với mẫu khai báo là:

long tongLe(int N);

Hàm này trả về tổng của tất cả các số lẻ từ 1 đến N (gồm cả số 1 và N nếu N lẻ, và ta phải giả sử N≥1). Ví dụ khi gọi **tongLe(7)**, kết quả trả về sẽ là 16 (vì 1+3+5+7 = 16); hoặc khi gọi **tongLe(12)** thì kết quả là 36 (vì 1+3+5+7+9+11 = 36).

b). Viết chương trình cho phép đọc giá trị số nguyên N từ bàn phím, nếu đó là số dương thì gọi hàm tongLe và kết quả được in ra màn hình.

Bài 7: Viết định nghĩa cho các hàm:

(1) Hàm tongNDGT để tính tổng $S = \sum_{i=1}^{n} \frac{1}{i!} \sum_{i=1}^{n} \frac{1}{i!}$

(2) Hàm tich để tính tích:

$$P = \begin{cases} 1.3.5 \dots n, & n \text{ lẻ} \\ 2.4.6 \dots n, & n \text{ chẵn} \end{cases} \begin{cases} 1.3.5 \dots n, & n \text{ lẻ} \\ 2.4.6 \dots n, & n \text{ chẵn} \end{cases}$$

Viết hàm main() và chạy thử 2 hàm nói trên. Thực hiện như bài 6, câu b.

Bài 8:

a). Viết định nghĩa hàm tìm ước số chung lớn nhất của hai số nguyên a và b, với khai báo như sau:

int ucln(int a, int b); (giống bài 4)

b). Viết hàm main() cho phép nhập hai số nguyên không âm đại diện cho tử và mẫu của một phân số, gọi hàm ucln ở câu a). để thực hiện rút gọn đến tối giản phân số ấy, rồi in ra phân số ấy sau khi đã rút gọn. Ví dụ nhập vào 2 số là 42 và 56, in ra 3/4.

Bài 9:

a). Viết định nghĩa cho các hàm có khai báo:

(1) void nhap(int A[ ], int &N);, hàm này cho phép nhập số N phần tử của mảng, sau đó tiếp tục nhập vào nội dung (từng giá trị một) của mảng A ứng với số phần tử ấy.

(2) void xuat(int A[ ], int N);, hàm này cho phép in ra N giá trị đang được lưu trữ trong mảng A.

b). Viết hàm main() trong đó khai báo một mảng A các số nguyên có MAX phần tử (tự định trị sao cho thỏa đáng) và một số nguyên N cho biết số số nguyên đang được dùng của mảng A. Sau đó gọi hàm nhap() để nhập vào các giá trị cho A, rồi gọi hàm xuat() để xuất ra nội dung của mảng và xem có đúng là nội dung mà bạn đã nhập vào hay không.

Lưu ý: Hai hàm nhap() và xuat() này sẽ được dùng lại trong các bài tập sau.

Bài 10:

a). Viết định nghĩa cho các hàm có khai báo:

(1) void xuatchan(int A[ ], int N);, có chức năng in ra tất cả các số chẵn của mảng A.

(2) void xuatVTchan(int A[ ], int N);, có chức năng in ra tất cả các số ở vị trí chẵn của mảng A.

(3) void thanhle(int A[ ], int N);, có chức năng cộng những số chẵn của mảng A với 1, nghĩa là sau khi thực hiện, ta đưởng mảng A gồm toàn những số lẻ.

b). Viết hàm main() trong đó khai báo mảng nguyên A có MAX phần tử và một số nguyên N chỉ số phần tử thực tế của mảng này. Gọi hàm nhap() để nhập nội dung cho mảng này, sau đó lần lượt gọi các hàm xuatchan(), xuatVTchan(), sau cùng là hàm thanhle() rồi hàm xuat() để kiểm tra xem liệu mảng A đã được biến đổi thành mảng chứa số lẻ chưa.

Bài 11:

a). Viết định nghĩa cho các hàm có khai báo sau:

(1) int timNhoNhat(int A[ ], int N);, tìm và trả về giá trị nhỏ nhất của dãy số nguyên A có N phần tử.

(2) double trungbinh(int A[ ], int N);, có chức năng tính giá trị trung bình của các phần tử của mảng A có N phần tử.

(3) int timkiem(int A[ ], int N, int k);, có chức năng tìm xem trong mảng A gồm N phần tử có phần tử nào có giá trị bằng với k không; nếu có thì trả về vị trí ấy, nếu không có thì trả về trị -1.

b). Viết hàm main() với các khai báo mảng như bài 10, gọi hàm nhap() và các hàm vừa định nghĩa ở câu a). để xem giá trị nhỏ nhất, trị trung bình của mảng A. Sau đó khai báo và cho nhập giá trị của biến k, gọi hàm timkiem() và in ra thông báo là có tìm được số k trong mảng A hay không.

Bài 12: Dùng lại hàm timkiem() ở bài 11.a.(3). Viết chương trình cho đọc vào các số nguyên có giá trị trong khoảng 10 đến 100. Khi mỗi số được đọc vào, hãy in ra nếu số này chưa từng xuất hiện trước đó. Chương trình cho in ra 20 giá trị số nguyên khác nhau.

Bài 13: Viết hàm cho nhận vào một mảng các số nguyên và số phần tử của mảng. Hàm có chức năng kiểm tra xem đây có phải là mảng đối xúng hay không? Nếu có đối xứng thì trả về 1, nếu không thì trả về 0.

Ví dụ:

Mảng đối xứng: [2 4 5 7 5 4 2]

Mảng không đối xứng: [2 3 8 7 4]

Bài 14: Viết hàm có các chức năng sau:

a). Tính và lưu N số Fibonacci đầu tiên vào một mảng nguyên A.

b). Tính tổng của nghịch đảo N-1 số Fibonacci đầu tiên, không dùng phần tử F_0.

c). Đảo ngược nội dung của mảng A.

Ví dụ: Với N = 7, mảng A là:

A = [0 1 1 2 3 5 8 13]

Sau khi đảo ngược, A trở thành:

A = [13 8 5 3 2 1 1 0]

Bài 15: Sắp xếp chọn (Selection Sort) một dãy số là cách sắp xếp mà ở mỗi bước ta tìm phần tử nhỏ nhất trong dãy rồi phần tử này được hoán chuyển với phần tử đầu dãy (nghĩa là đưa phần tử nhỏ nhất về đầu dãy).

Ví dụ: Dãy ban đầu: 6 5 1 3 □ Dãy hoán đổi: 1 5 6 3

Sau đó lặp lại bước này với dãy con được hình thành bằng dãy trước đó bỏ đi phần tử đầu tiên.

Ví dụ: Ở bước 2, dãy ban đầu: 5 6 3 □ Dãy hoán đổi: 3 6 5

Việc lặp lại này kết thúc khi dãy con chỉ có 1 phần tử. Khi đó, ta được toàn bộ dãy đã được sắp xếp.

Hãy viết hàm nhận vào một mảng các số nguyên và số phần tử của mảng đó, sau đó sắp xếp mảng số nguyên này thành dãy tăng dần dùng phương pháp Selection Sort.