**ĐỀ THI MẪU CUỐI KỲ CƠ SỞ LẬP TRÌNH**

**Bài 1 (5 điểm)**

*Yêu cầu 1 (2 điểm):* Viết 4 hàm sau (mỗi hàm 0.5 điểm)

* Viết hàm Nhan2So với khai báo và danh sách tham số như sau:

**double Nhan2So (double a, double b)**: Hàm này trả về tích (kết quả phép nhân) 2 số a và b.

* Viết hàm Tong2So với khai báo và danh sách tham số như sau:

**double Tong2So (double a, double b)**: Hàm này trả về tổng (kết quả phép cộng) 2 số a và b.

* Viết hàm TongBinhPhuong với khai báo và danh sách tham số như sau:

**double TongBinhPhuong (double a, double b)**: Hàm này trả về tổng bình phương (a2+ b2) của 2 số a và b.

* Viết hàm CanBac2 với khai báo và danh sách tham số như sau:

**double CanBac2 (double a)**: Hàm này trả về căn bậc 2 của số a.

*Yêu cầu 2 (3 điểm)*

Sau đó viết chương trình chính (hàm main) thực hiện nhiều lần công việc sau:

Hiện menu đơn giản sau đây trên màn hình:

1- Thông tin hình vuông

2- Thông tin hình chữ nhật

3- Thông tin tam giác vuông.

4- Chấm dứt.

Chương trình thực hiện nhiều lần đến khi nào người dùng chọn số 4. Nếu chọn 1 thì nhập cạnh hình vuông sau đó in ra chu vi, diện tích và chiều dài đường chéo hình vuông, nếu chọn 2 thì nhập chiều dài 2 cạnh sau đó cũng in ra chu vi, diện tích và chiều dài đường chéo hình chữ nhật. Nếu chọn 3 thì nhập chiều dài 2 cạnh vuông của tam giác vuông và sau đó cũng in ra chu vi, diện tích và chiều dài cạnh huyền của tam giác vuông.

Lưu ý:

* Phải tính toán bằng cách gọi các hàm đã thiết kế bên trên.
* Khi người dùng nhập vào chiều dài các cạnh không hợp lệ thì phải cho nhập lại.

**Bài 2 (5 điểm)**

*Yêu cầu 1 (2 điểm):*

Viết 4 hàm sau (mỗi hàm 0,5 điểm)

* Thiết kế hàm **void NhapMang(double a[ ], long n)** hàm này thực hiện việc nhập n số thực vào mảng a (mảng a có n phần tử chứa số thực)
* Thiết kế hàm **long PhamVi(const double a[ ], long n, double min, double max)**, hàm này sẽ in ra các phần tử trong mảng a (mảng a có n phần tử chứa số thực) thỏa điều kiện lớn hơn hay bằng min và nhỏ hơn hay bằng max, sau đó hàm này trả về số lượng phần tử trong mảng a thỏa điều kiện này.
* Thiết kế hàm **double GiaTriTrungBinh(const double a[ ], long n)**, hàm này sẽ trả về giá trị trung bình của các phần tử trong mảng a (mảng a có n phần tử chứa số thực)
* Thiết kế hàm **double GiaTriMax(const double a[ ], long n)**, hàm này sẽ trả về giá trị lớn nhất trong mảng a (mảng a có n phần tử chứa số thực)

*Yêu cầu 2 (3 điểm):*

Viết chương trình chính thực hiện công việc sau:

Khai báo 1 mảng có 7 phần tử chứa số thực để lưu nhiệt độ 7 ngày trong tuần (theo thứ tự từ Thứ Hai, Thứ Ba,.., Thứ Bảy, Chủ Nhật)

Sau đó nhập dữ liệu vào mảng này.

Sau khi nhập xong dữ liệu, xuất hiện 1 menu đơn giản cho phép thực hiện nhiều lần công việc sau:

1-Nhiệt độ cao nhất.

2-Nhiệt độ trung bình cả tuần.

3-Nhiệt độ trong phạm vi [x,y]

4-Nhiệt độ trung bình n ngày đầu tiên

5-Kết thúc.

Nếu chọn 1: chương trình sẽ cho biết nhiệt độ cao nhất trong tuần là bao nhiêu?

Nếu chọn 2: Nhiệt độ trung bình trong tuần.

Nếu chọn 3, người dùng sẽ nhập vào 2 nhiệt độ x và y (x<=y), sau đó chương trình sẽ hiện ra các nhiệt độ trong tuần nằm trong phạm vi [x,y] và số ngày trong tuần có nhiệt độ trong phạm vi này.

Nếu chọn 4: Người dùng sẽ nhập 1 số nguyên n trong phạm vi từ 1-7, sau đó chương trình sẽ in ra nhiệt độ trung bình của n ngày đầu tiên trong tuần (Gợi ý: Gọi hàm **GiaTriTrungBinh** đã thiết kế bên trên với tham số mô tả số phần tử trong mảng phù hợp).

Chương trình thực hiện nhiều lần cho đến khi người dùng chọn số 5.

***--Hết—***