

Hướng dẫn giải bài tập cơ bản - Bài thực hành 6

HÀM

1. Hướng dẫn giải bài 1:

❖ Hàm tính diện tích hình chữ nhật:

- Xác định mục đích/mục tiêu của hàm: Tính diện tích hình chữ nhật
- Định nghĩa dữ liệu hàm sẽ nhận từ nơi gọi (*parameters*).
 - + Chiều dài hình chữ nhật (số thực)
 - + Chiều rộng hình chữ nhật (số thực)
- Định nghĩa dữ liệu hàm sẽ tính toán và trả về (*return value*): diện tích hình chữ nhật (số thực)
- Định nghĩa các bước được thực hiện để đạt được mục tiêu (*algorithm*)
 - + Tính chiều dài * chiều rộng và gán kết quả cho `dienTich`;
 - + Trả về giá trị của `dienTich`

Định nghĩa hàm:

```
double tinhDT_HCN (double chieuDai, double chieuRong)
{
    double dienTich;
    dienTich = chieuDai * chieuRong;
    return dienTich;
}
```

❖ Hàm tính chu vi hình chữ nhật:

Tương tự như hàm tính diện tích, nhưng tính chu vi = (chiều dài + chiều rộng) * 2;

❖ Hàm main:

- Nhập chiều dài, chiều rộng hình chữ nhật.
- Gọi hàm tính diện tích, xuất kết quả.
- Gọi hàm tính chu vi, xuất kết quả.

2. Hướng dẫn giải bài 2:

- Hàm tìm số lớn nhất của 2 số nguyên

- Mục đích: tìm số lớn nhất của 2 số nguyên
- Dữ liệu nhận vào: 2 số nguyên a và b
- Dữ liệu hàm trả về: số lớn nhất (số nguyên)
- Thuật giải:
 - Nếu $a \geq b$ thì $\text{max} = a$;
 - Ngược lại thì $\text{max} = b$;
 - Trả về kết quả max

Định nghĩa hàm:

```
int timSoLonNhat (int a, int b)
{
    int max;
    if (a >= b)
        max = a;
    else
        max = b;
    return max;
}
```

- **Hàm tìm số lớn nhất của 3 số nguyên:** tìm số lớn nhất của 2 số đầu tiên, sau đó so giá trị lớn nhất đó với số thứ 3 để có kết quả cuối cùng.
- *Sinh viên tự xây dựng hàm main để gọi thực thi 2 hàm vừa xây dựng.*

3. Hướng dẫn giải bài 3:

- Mục đích: tính lũy thừa x^y
- Dữ liệu nhận vào: số thực x, số nguyên y
- Dữ liệu hàm trả về: kết quả lũy thừa (số thực)
- Thuật giải:
 - Ngược lại thì nếu $y \geq 0$ thì $\text{luyThua} = x * x * \dots * x$ (y lần)
 - Ngược lại (tức là $y < 0$) thì $\text{luyThua} = x / x / \dots / x$ (-y lần)
 - Trả về kết quả luyThua

Định nghĩa hàm:

```
double tinhLuyThua(double x, int y)
{
    double luyThua = 1.0;
    if (y >= 0)
        for (int i = 1; i <= y; i++)
            luyThua *= x;
    else
        for (int i = 1; i <= -y; i++)
            luyThua /= x;
    return luyThua;
}
```

- Sinh viên tự xây dựng hàm main để gọi thực thi hàm vừa xây dựng.

4. Hướng dẫn giải bài 4

- **Hàm tính tổng các số từ 1 đến n:**

- Mục đích: tính tổng các số từ 1 đến n
- Dữ liệu nhận vào: số nguyên dương n
- Dữ liệu hàm trả về: kết quả tổng
- Thuật giải:
 - Khởi tạo $i = 1$; $kqTong = 0$;
 - Lặp lại các bước sau trong khi $i \leq n$:
 - $kqTong += i$;
 - tăng i lên 1 đơn vị
 - Trả về kết quả biến $kqTong$;

- **Hàm tính $n!$**

Tương tự hàm tính tổng nhưng thay đổi phép tính.

- Sinh viên tự xây dựng hàm main để gọi thực thi 2 hàm vừa xây dựng

5. Hướng dẫn giải bài 5

- Mục đích: xuất hình vuông
- Dữ liệu nhận vào: số đo cạnh hình vuông cạnh (số nguyên)
- Dữ liệu hàm trả về: không có
- Thuật giải:

Duyệt cạnh lần, mỗi lần duyệt:

- Duyệt canh lần, mỗi lần xuất “*”
- Xuống dòng

Định nghĩa hàm

```
void xuatHinhVuong(int canh)
{
    for (int i = 1; i <= canh; i++)
    {
        for (int j = 1; j <= canh; j++)
            cout << "*";
        cout << endl;
    }
}
```

- Sinh viên tự xây dựng hàm main để gọi thực thi hàm vừa xây dựng.

6. Hướng dẫn giải bài 6:

- Mục đích: tính số đảo ngược của số nguyên dương n
- Dữ liệu nhận vào: số nguyên dương n
- Dữ liệu hàm trả về: số đảo ngược (cũng là số nguyên dương)
- Thuật giải: sinh viên tham khảo cách tính số đảo ngược ở bài tập 5 – bài thực hành 4.

7. Hướng dẫn giải bài 7

- Mục đích: kiểm tra 1 số nguyên có phải là số nguyên tố
- Dữ liệu nhận vào: số nguyên n
- Dữ liệu hàm trả về: đúng/ sai (kiểu bool)
- Thuật giải: sinh viên tham khảo cách kiểm tra 1 số có phải là số nguyên tố hay không ở bài thực hành 4.