



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN  
HỆ CHÍNH QUY  
MÔN: **NHẬP MÔN LẬP TRÌNH**

## **HƯỚNG DẪN THỰC HÀNH TUẦN 04**

### **LẬP TRÌNH HÀM**

## MỤC LỤC

1	Các bước viết hàm .....	3
2	Cách gọi hàm.....	4
3	Truyền tham biến và tham trị .....	5
4	Bài tập luyện tập .....	7
5	Yêu cầu nộp bài .....	9

# 1 Các bước viết hàm

Xác định các thông tin sau:

- Tên hàm (bắt đầu bằng động từ, viết thường).
- Tên hàm mô tả công việc chính của hàm. Mỗi hàm chỉ thực hiện 1 công việc.
- Các tham số đầu vào (nếu có).
- Kết quả trả về (nếu có).

## Ví dụ 1:

Hàm nhập một số nguyên dương.

- **Tên hàm:** `nhapNguyenDuong1`
- **Công việc thực hiện:** nhập một số nguyên từ bàn phím và kiểm tra có phải số dương không? Nếu không thì bắt nhập lại.
- **Tham số đầu vào:** chỉ có 1 tham số là 1 biến kiểu số nguyên. Hàm này sẽ yêu cầu người dùng nhập 1 số nguyên từ bàn phím và lưu giá trị vào biến số nguyên này.
- **Kết quả trả về:** không có (**void**).

```
void nhapNguyenDuong1(int &n)
{
    do {
        cout << "Nhap mot so nguyen
duong: ";
        cin >> n;
    } while (n <= 0);
}
```

Tại sao có kí hiệu ‘&’ ở trước *n*?

## Ví dụ 2:

Hàm nhập một số nguyên dương.

- **Tên hàm:** `nhapNguyenDuong2`
- **Công việc thực hiện:** nhập một số nguyên từ bàn phím và kiểm tra có phải số dương không? Nếu không thì bắt nhập lại.
- **Tham số đầu vào:** không có.
- **Kết quả trả về:** một giá trị kiểu *int*

```
int nhapNguyenDuong2()
{
    int n;
    do {
        cout << "Nhap mot so nguyen
duong: ";
        cin >> n;
    } while (n <= 0);
    return n;
}
```

**Ví dụ 3:**

Tính tổng 3 số nguyên.

- **Tên hàm:** tinhTong3SoNguyen
- **Công việc thực hiện:** nhận vào 3 số nguyên và tính tổng.
- **Tham số đầu vào:** 3 số nguyên a, b, c đã có giá trị sẵn rồi, bây giờ chỉ đem đi tính tổng thôi.
- **Kết quả trả về:** 1 số nguyên là tổng của a, b, c.

```
int tinhTong3SoNguyen(int a, int b, int c)
{
    int kq = a + b + c;
    return kq;
}
```

**Lưu ý:** mỗi hàm chỉ thực hiện một công việc chính mà thôi. Chẳng hạn như các hàm tính toán thì chỉ thực hiện tính toán rồi trả kết quả về, ngoài ra không làm gì khác. Ví dụ hàm tính tổng thì chỉ tính tổng mà thôi, **KHÔNG** có tính hiệu, tính tích, hay xuất kết quả ra màn hình.

## 2 Cách gọi hàm

**Ví dụ 4:** Chú ý các đoạn mã nguồn nhập số nguyên được tô màu giống nhau.

Không dùng hàm	Dùng hàm
<pre>void main() {     //Ch.trình nhập 3 số nguyên a,b,c     int a, b, c;     do {         cout &lt;&lt; "Nhập một số nguyên dương: ";         cin &gt;&gt; a;     } while (a &lt;= 0);      do {         cout &lt;&lt; "Nhập một số nguyên dương: ";         cin &gt;&gt; b;     } while (b &lt;= 0);      do {         cout &lt;&lt; "Nhập một số nguyên dương: ";         cin &gt;&gt; c;     } while (c &lt;= 0); }</pre>	<pre>//giống như biến, hàm cũng phải khai báo //trước khi sử dụng (<b>prototype</b>)  //Hàm nhập trong ví dụ 1 void nhapNguyenDuong1(int &amp;n); //Hàm nhập trong ví dụ 2 int nhapNguyenDuong2();  void main() {     int a, b, c;     nhapNguyenDuong1(a);     nhapNguyenDuong1(b);     c = nhapNguyenDuong2(); }  //Sinh viên tự viết lại hàm nhập số nguyên đương trong ví dụ 1 và 2 vào sau hàm main.</pre>

**Ví dụ 5:** Chú ý các đoạn mã nguồn được tô màu giống nhau.

Không dùng hàm	Dùng hàm
<pre> void main () {     int a, b;     bool ktra = true;      cin &gt;&gt; a, b;     for (int i = 2 ; i &lt; a ; i++)         if (a%i == 0)             ktra = false;     if (ktra == true)         cout &lt;&lt; a &lt;&lt;" la SNT";      ktra = true;     for (int i = 2 ; i &lt; b ; i++)         if (b%i == 0)             ktra = false;     if (ktra == true)         cout &lt;&lt; b &lt;&lt;" la SNT"; } </pre>	<pre> bool ktSNT(int x);  void main () {     int a, b;     bool kq;      cin &gt;&gt; a, b;      kq = KtrSNT(a);      if (kq == true)         cout &lt;&lt; a &lt;&lt;" la SNT";      if (KtraSNT(b) == true)         cout &lt;&lt; b &lt;&lt;" la SNT"; }  //Sinh viên tự viết lại hàm kiểm tra số nguyên tố. </pre>

### 3 Truyền tham biến và tham trị

#### Ví dụ 6:

Chương trình	Màn hình
<pre> int nhap1(int n); int nhap2(int &amp;n); void main() {     int a = 3;      nhap1(a);     cout &lt;&lt; a;      nhap2(a);     cout &lt;&lt; a; } void nhap1(int m)//truyền tham trị {     m = 5; } void nhap2(int &amp;m)//truyền tham biến {     m = 5; } </pre>	<p>→ Kết quả bao nhiêu? Vì sao?</p> <p>→ Kết quả bao nhiêu? Vì sao?</p>

## Phạm vi các biến trong chương trình con

Chương trình	Chú thích về phạm vi các biến (các chú thích nhỏ phía trên)
<pre> bool kTraSNT(int n); void xuatSNT(int n); void nhapSoDuong(int &amp;n); void main() {     int n;     nhapSoDuong(n);     xuatSNT(n); } void nhapSoDuong(int &amp;n) {     //đoạn chtr nhập vào 1 giá trị     cho     //biến n và ktra giá trị nhập     //có phải là số dương không? } void xuatSNT(int n) {     int i;     for (int i = 0 ; i &lt; n ; i++)         if (KTraSNT(n))             cout &lt;&lt; i; } bool kTraSNT(int n) {     int i;     //đoạn chtr ktr n có phải số     ng.tổ? } </pre>	<pre> bool kTraSNT(int n<sup>KTraSNT</sup>); void xuatSNT(int n<sup>XuatsNT</sup>); void nhapSoDuong(int &amp;n<sup>(tùy vào hàm nào gọi đến nó)</sup>); void main() {     int n<sup>main</sup>;     nhapSoDuong(n<sup>main</sup>);     xuatSNT(n<sup>main</sup>); } void nhapSoDuong(int &amp;n<sup>main</sup>) {     //đoạn ch.trình nhập vào 1 giá trị cho     //biến n<sup>main</sup> và ktra giá trị nhập     //có phải là số dương không? } void xuatSNT(int n<sup>XuatsNT</sup>) {     int i<sup>XuatsNT</sup>;     for (int i<sup>XuatsNT</sup> = 0 ; i<sup>XuatsNT</sup> &lt; n<sup>XuatsNT</sup> ; i<sup>XuatsNT</sup>++)         if (KTraSNT(i<sup>XuatsNT</sup>))             cout &lt;&lt; i<sup>XuatsNT</sup>; } bool kTraSNT(int n<sup>KTraSNT</sup>) {     int i<sup>KTraSNT</sup>;     //đoạn ch.trình ktr n<sup>KTraSNT</sup> có phải số     //ng.tổ? } </pre>

## 4 Bài tập luyện tập

Thực hiện các bài tập sau bằng cách viết các chương trình con thực hiện các công việc: nhập (có kiểm tra điều kiện nhập), kiểm tra một số thỏa điều kiện số hoàn chỉnh, số đối xứng, tìm ước chung...)

**Bài 1:** Số  $n$  được gọi là *số hoàn chỉnh* khi tổng các ước số của  $n$  bằng với tích các ước số của nó (không kể số  $n$ ). Ví dụ: số 6 là số hoàn chỉnh vì nó có 3 ước số là 1, 2, 3 và chúng ta có  $6 = 1 + 2 + 3 = 1 \times 2 \times 3$ . Viết chương trình kiểm tra số nguyên dương  $n$  có là số hoàn chỉnh không? (với giá trị  $n$ :  $1 \leq n \leq 10.000.000$ ).

**Gợi ý:**

- Viết 2 hàm con: *nhapSoDuong*, *ktSoHoanChinh*
- Sử dụng biến  $S$  để tính tổng,  $P$  để tính tích các ước số của  $n$ . Khởi tạo:  $S=0$ ,  $P=1$ .
- Tính ước số của  $n$ : xem lại ví dụ 1 để biết cách tìm ước
- Trong vòng lặp tìm ước, cộng dồn  $S$  và nhân dồn  $P$ .
- Ra khỏi vòng lặp, so sánh  $S = P$ ?

**Bài 2:** Số  $n$  có *tính chất đối xứng* khi các chữ số của nó được viết ngược lại cũng chính là  $n$ . Ví dụ: số  $n = 34543$ ; 11; 4114 có tính chất đối xứng. Viết chương trình kiểm tra số nguyên dương  $n$  có tính chất đảo ngược không? (với giá trị  $n$ :  $10 \leq n \leq 10.000.000$ ).

**Gợi ý:**

- Viết 2 hàm con: *nhapSoDuong*, *ktSoDoiXung*
- Sử dụng tính chất sau  $N = a_k a_{k-1} \dots a_1 a_0 = a_k * 10^k + a_{k-1} * 10^{k-1} + \dots + a_0 * 10^0$
- Trong vòng lặp, tách từng chữ số của số  $N$ , rồi áp dụng công thức trên để tính ra tổng  $X$ . Trong lập trình C: dùng  $N \% 10$  để lấy chữ số cuối và  $N = N / 10$  để giảm số.
- Ra khỏi vòng lặp, so sánh  $N = X$ ?

**Bài 3:** Số  $n$  được gọi là *số hoàn hảo* khi tổng các ước số của  $n$  bằng 2 lần  $n$ . Ví dụ:  $n = 28$  thì  $28 * 2 = 1 + 2 + 4 + 7 + 14 + 28$ . Viết chương trình kiểm tra số nguyên dương  $n$  có phải là số hoàn hảo không? (với giá trị  $n$ :  $1 \leq n \leq 100.000.000$ ).

**Gợi ý:**

- Viết 2 hàm con: *nhapSoDuong*, *ktSoHoanHao*
- Tương tự bài 1

**Bài 4:** Viết chương trình tìm chữ số lớn nhất của số nguyên dương  $n$ . Ví dụ:  $n = 13483$  thì chữ số lớn nhất của số  $n$  là 8.

**Gợi ý:**

- Viết 2 hàm con: ***nhapSoDuong, timChuSoLonNhat***
- Gán MAX = ký số cuối cùng của số N.
- Trong vòng lặp, tách từng ký số của số N, rồi ngay lập tức so sánh từng ký số với số MAX để có kết luận.

**Bài 5:** Nhập vào các số thực, quá trình nhập sẽ ngừng khi người dùng nhập vào số 0, hãy đếm xem có bao nhiêu số thực âm trong các số vừa nhập?

**Gợi ý:**

- Viết 1 hàm con: ***demSoAm***
- Khởi tạo biến dem = 0.
- Dùng duy nhất 1 biến k để lưu đề giá trị các số nhập vào từ bàn phím
- Trong vòng lặp nhập số k, kiểm tra  $k < 0$  thì tăng biến dem lên 1.

**Bài 6:** Viết chương trình tìm ước chung lớn nhất của 2 số nguyên dương a và b

**Gợi ý:**

- Viết 2 hàm con: ***nhapSoDuong, timUCLN***
- Khái niệm: là số k lớn nhất mà cả a và b đều chia hết ( $a \% k = 0$  và  $b \% k = 0$ )

**Bài 7:** Viết chương trình tìm bội chung nhỏ nhất của 2 số nguyên dương a và b

**Gợi ý:**

- Viết 2 hàm con: ***nhapSoDuong, timBCNN***
- Khái niệm: là số k nhỏ nhất mà chia hết cả cho a và b ( $k \% a = 0$  và  $k \% b = 0$ )

**Bài 8:** Viết chương trình Hoán vị giá trị 2 số nguyên a và b.

**Gợi ý:**

- Viết 1 hàm con: ***hoanviSoNguyen***
- Cách 1: Sử dụng biến tạm tempt để gán qua lại giữa các giá trị a và b
- Cách 2: Sử dụng phép toán trừ (áp dụng công thức đã học ở cấp dưới)

**\* Bài 9:** Viết chương trình nhập vào 1 số nguyên n. Chuyển đổi n sang hệ nhị phân và xuất ra màn hình. (với giá trị n:  $-255 \leq n \leq 255$ )

- Viết 1 hàm con: ***chuyenNhiPhan***



\* **Bài 10:** Dãy số nguyên  $\{F_i\}$  được định nghĩa như sau: 
$$\begin{cases} F_0 = 1 \\ F_1 = 2 \\ F_n = 3.F_{n-1} - 2.F_{n-2}, \quad n > 1 \end{cases}$$

Cho số nguyên dương  $n$ . Viết chương trình tìm giá trị của  $F_n$ .

- Viết 2 hàm con: ***nhapSoDuong, tinhF***

## **5 Yêu cầu nộp bài**

Tất cả sv làm các bài tập luyện tập trên và nộp trên moodle môn học.