


<b>Trường ĐH CNTP TP.HCM</b> <b>Khoa: CNTT</b> <b>Bộ môn: CNPM</b> <b>Môn: TH NMLT</b>	<b>BÀI 3. CẤU TRÚC LẶP (tt)</b>	
---	---------------------------------	---

### A. MỤC TIÊU:

- Xác định các bài toán cần sử dụng cấu trúc lặp
- Áp dụng được cấu trúc lặp while, do...while để xây dựng cách giải bài toán thực tế.
- Phân tích được công việc lặp của bài toán.
- Cài đặt được bài toán lặp sử dụng câu lệnh while, do...while
- Kết hợp cấu trúc lặp có lồng cấu trúc điều kiện if, if...else..., switch...case
- Phân biệt sự khác nhau khi sử dụng cấu trúc for, while, do...while...
- Áp dụng được các từ khóa break, continue lồng vào trong vòng lặp

### B. DỤNG CỤ - THIẾT BỊ THÍ NGHIỆM CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	– Máy tính (có thể là máy bàn hoặc laptop).	1	cái	

### C. VẬT LIỆU

- Visual C++.Net.
- Phần mềm hỗ trợ dạy học: NetOp School hoặc NetSupport School.

### D. NỘI DUNG THỰC HÀNH

#### I. Tóm tắt lý thuyết

##### 1. Cấu trúc lặp while

- Sử dụng khi bài toán có công việc lặp mà không biết số lần lặp là bao nhiêu, chỉ xác định được điều kiện dừng lặp của bài toán.

Ví dụ: Nhập vào dãy số nguyên cho tới khi người dùng nhập số 0 thì dừng. Trong ví dụ, công việc nhập số nguyên lặp đi lặp lại nhiều lần nhưng không biết là lặp bao nhiêu lần. Vòng lặp chỉ dừng khi người dùng nhập số 0.

- Cú pháp:

```
while (<điều kiện>)
{
    //công việc lặp
}
```

- Ví dụ:

```
while (x!=0)
{
```

```

printf("nhap x: ");
scanf("%d", &x);
}

```

- Trong câu lệnh while, nếu biểu thức điều kiện trả về giá trị đúng thì vòng lặp tiếp tục được thực hiện
- Để tránh vòng lặp vô tận, trong câu lệnh while phải có câu lệnh làm thay đổi giá trị của biểu thức điều kiện.

## 2. Cấu trúc lặp do...while

- Tương tự như cấu trúc lặp while, nhưng đoạn lệnh cần lặp thực hiện ít nhất một lần (while có thể không thực hiện lần nào nếu ngay mới lần đầu tiên biểu thức điều kiện trong while trả về giá trị sai)
- Cú pháp:

```

do
{
    //công việc cần lặp
} while (<điều kiện>);

```

- Ta có thể dùng từ khóa **break** để thoát ra khỏi vòng lặp

```

int i = 1;
do
{
    if(i == 5) break;
    printf("%d", i);
    i++;
}while(i < 10);

```

1 2 3 4

- Dùng từ khóa **continue** để chuyển sang bước lặp tiếp theo.

```

int i = 1;
do
{
    if(i == 5) continue;
    printf("%d", i);
    i++;
}while(i < 10);

```

2 3 4 6 7 8 9 10

## II. Bài tập mẫu có hướng dẫn

**Bài 1.** Nhập vào 2 số nguyên dương a và b. Tìm ước chung lớn nhất của 2 số nguyên dương đã nhập.

Bước 1: Phân tích bài toán

- Nhập 2 số nguyên bất kỳ
  - Xuất ra ước chung lớn nhất của 2 số vừa nhập
  - 02 biến số nguyên (int), 01 biến ucln (int)
  - Sử dụng thuật toán Euclid => số lần lặp của bài toán không xác định
- => sử dụng vòng lặp while
- Thuật toán Euclid tìm UCLN của a và b

Nếu  $a=b$  thì  $UCLN = a$ ;

Nếu  $a > b$  thì  $UCLN(a,b) = UCLN(a-b,b)$

Nếu  $a < b$  thì  $UCLN(a,b) = UCLN(a,b-a)$

Bước 2: Cài đặt chương trình

- Khai báo thư viện và viết hàm main
- Khai báo và nhập dữ liệu

```
int a,b;
printf("\n nhập vào 2 số a và b:");
scanf("%d%d", &a, &b);
```

- Cài đặt câu lệnh lặp

```
while (a != b)
{
    if(a > b)
        a = a - b;
    else
        b = b - a;
}
printf("UCLN là: %d", a);
```

Bước 3: Kiểm tra, sửa lỗi và chạy thử

**Bài 2.** Xây dựng một menu chương trình cho phép người dùng lựa chọn gồm các yêu cầu sau:

1. In ra màn hình từ hello
2. In ra màn hình câu Vòng lặp for
3. In ra màn hình câu Vòng lặp while
4. Thoát chương trình.

Sau khi lựa chọn chức năng, chương trình sẽ thực hiện theo đúng chức năng đã chọn rồi lại tiếp tục hiển thị menu. Chương trình chỉ dừng lại (kết thúc khi người dùng bấm phím 4).

Bước 1: Phân tích bài toán

- Xuất ra màn hình menu theo yêu cầu
- Sử dụng 01 biến luachon là số nguyên (int)

- Menu này xuất hiện ít nhất 1 lần (cho lần chọn đầu tiên) và lặp đi lặp lại nhiều lần => dùng cấu trúc lặp do...while

### Bước 2: Cài đặt chương trình

- Cài đặt chương trình

```
int luachon;
do
{
    printf("1. In ra màn hình tu Hello. \n");
    printf("2. In ra màn hình Vòng lặp for \n");
    printf("3. In ra màn hình Vòng lặp while \n");
    printf("4. Thoát chương trình \n");

    printf("Bạn chọn chức năng nào: \n");
    scanf("%d", &luachon);
    {
        case 1: printf("HELLO \n");
            break;
        case 2: printf("Vòng lặp FOR \n");
            break;
        case 3: printf("Vòng lặp WHILE \n");
            break;
        case 4: printf("Thoát chương trình\n");
            break;
    }

} while (luachon!=4);
```

### Bước 3: Kiểm tra, sửa lỗi và chạy thử

**Bài 3.** Viết chương trình nhập vào một số nguyên dương n. Tính tổng các chữ số có trong số nguyên dương n

### Bước 1: Phân tích bài toán

- Nhập vào số nguyên dương n
- Tính tổng các chữ số có trong số nguyên dương n
- Cách thực hiện, ví dụ với n=629

tong = 0;

n = 629 > 0:

- tong = tong + hàng đơn vị của n (số 9: n%10)
- Gán lại giá trị cho n: n = n/10

n = 62 > 0:

- tong=tong + hàng đơn vị của n (số 2: n%10)
- Gán lại giá trị cho n: n = n / 10

n = 6 > 0:

- $tong = tong + \text{hàng đơn vị của } n \text{ (số 6: } n \% 10)$
  - Gán lại giá trị cho  $n$ :  $n = n / 10$   
 $n = 0$ :dừng
- Ta thấy việc  $tong = tong + \text{hàng đơn vị của } n$  và  $n = n / 10$  được lặp lại nhiều lần nhưng không biết cụ thể bao nhiêu lần. Do đó ta phải dùng vòng lặp while.

#### Bước 2: Cài đặt chương trình

- Khai báo và nhập dữ liệu:

```
int n;
printf("\n Nhập vào 1 số nguyên dương:");
scanf("%d", &n);
```

- Cài đặt câu lệnh

```
int tong=0;
while (n>0)
{
    tong = tong + n%10;
    n=n/10;
}
printf("tong cac chu so trong n la:%d", tong);
```

#### Bước 3: Kiểm tra, sửa lỗi và chạy thử

```
nhap so nguyen duong n: 629
tong cac chu so trong n la:17
```

### **III. Bài tập thực hành trên lớp**

- Bài 4.** Viết chương trình kiểm tra xem  $n$  có phải là số nguyên tố hay không. Biết rằng số nguyên tố là số nguyên lớn hơn 1 và chỉ có 2 ước là 1 và chính nó.
- Bài 5.** Viết chương trình cho biết trong số nguyên  $n$  có bao nhiêu chữ số chẵn.
- Bài 6.** Viết chương trình tìm bội chung nhỏ nhất (BCNN) của 2 số nguyên dương  $a$  và  $b$  nhập vào từ bàn phím. Biết rằng  $BCNN(a,b) = a*b / UCLN(a,b)$ .
- Bài 7.** Viết chương trình tính tổng các số tự nhiên nhỏ hơn  $n$  (bằng cách sử dụng vòng lặp while).
- Bài 8.** Viết chương trình thực hiện việc lặp lại thao tác nhập và xuất một số nguyên bất kỳ từ bàn phím. Việc nhập kết thúc khi người dùng nhập vào giá trị 0. Hiển thị tổng các số vừa nhập.
- Bài 9.** Viết chương trình thực hiện việc lặp lại thao tác nhập và xuất một số nguyên có đúng 3 chữ số và các chữ số phải khác nhau từ bàn phím. Việc nhập kết thúc khi người dùng nhập vào giá trị vi phạm điều kiện trên. Hiển thị số lượng các số vừa nhập.
- Bài 10.** Viết chương nhập số nguyên lớn  $N$  (khai báo: long  $N$ ) có  $k$  chữ số.
1. Tìm chữ số hàng đầu tiên của  $N$ .

2. Tính tổng các chữ số của N.
3. Tìm chữ số lớn nhất trong k chữ số đó.
4. Đếm số chữ số của N.
5. N có phải là số có các chữ số khác nhau đôi một không?

*Gợi ý:* Áp dụng các phép toán / và % cho 10 để lấy lần lượt từng chữ số của N theo chiều từ chữ số hàng đơn vị đến các chữ số hàng chục, hàng trăm,...

Ví dụ:  $N = 347285108$ ,  $x = N \% 10 = 8$ ,  $y = N / 10 = 34728510$ . Lặp lại 2 phép toán chia này ta lần lượt lấy được từng chữ số của N từ hàng đơn vị.

Khi đó N có:

- Chữ số hàng đầu tiên là 3.
- Tổng các chữ số:  $3+4+7+2+8+5+1+0+8 = 38$
- Chữ số lớn nhất là 8.
- Số chữ số của N: 9

N không phải số có các chữ số khác nhau đôi một vì có 2 chữ số 8.

**Bài 11.** Bài toán cổ “Trăm trâu trăm cỏ” :

*Trâu đứng ăn năm*

*Trâu nằm ăn ba*

*Lụ khụ trâu già*

*Ba con một bó*

*Trăm con ăn cỏ*

*Trăm bó no nê.*

Hỏi có bao nhiêu trâu đứng, trâu nằm, trâu già ?

**Bài 12.** Giải và biện luận phương trình bậc một:  $ax+b=0$ .

#### IV. Bài tập về nhà:

**Bài 13.** Kiểm tra số nguyên dương n có các chữ số toàn là chữ số chẵn hay không?

**Bài 14.** Kiểm tra số nguyên dương n có các chữ số toàn là chữ số lẻ hay không?

**Bài 15.** Kiểm tra số nguyên dương n có các chữ số tăng dần từ trái qua phải không?

**Bài 16.** Kiểm tra số nguyên dương n có các chữ số giảm dần từ trái qua phải không?

**Bài 17.** Viết chương trình nhập một số nguyên dương n. Tìm số nguyên dương m lớn nhất sao cho  $1+2+3+\dots+m < n$ .

**Bài 18.** Viết chương trình nhập một số nguyên dương n. Tìm số nguyên dương m nhỏ nhất sao cho  $1+2+3+\dots+m > n$ .

**Bài 19.** Viết chương trình xuất số đảo của số nguyên dương n, n nhập từ bàn phím.

**Bài 20.** An gửi số tiền là n USD vào ngân hàng. An muốn số tiền của mình sẽ bằng hoặc lớn hơn 1000 USD. Giúp An viết chương trình tính số tháng cần gửi vào ngân hàng để tổng tiền vốn lẫn tiền lãi sẽ lớn hơn 1000USD. Biết lãi suất 0.7%/ tháng và số tiền vốn n được nhập từ bàn phím.

**Bài 21.** Giải và biện luận phương trình bậc hai:  $ax^2+bx+c=0$ .

**Bài 22.** Giải và biện luận phương trình trùng phương bậc bốn:  $ax^4+bx^2+c=0$

**Bài 23.** Nhập n và tính các tổng S sau:

$$1. S(n) = \sqrt{n + \sqrt{(n-1) + \sqrt{(n-2) + \cdots + \sqrt{1}}}} \text{ với } n > 0$$

$$2. S(n) = \sqrt{1 + \sqrt{2 + \sqrt{3 + \cdots + \sqrt{n}}}} \text{ với } n > 0$$