

# Bài 7: CÂU LỆNH LẶP

## I/ Các công việc phải thực hiện nhiều lần:

- Trong cuộc sống hàng ngày, nhiều hoạt động được thực hiện lặp đi lặp lại nhiều lần.
- Có những hoạt động mà chúng ta thực hiện lặp với số lần nhất định và biết trước, và những công việc và số lần không biết trước.

VD

+ số lần lặp biết trước: Các ngày trong tuần các em đều lặp đi lặp lại hoạt động buổi sáng đến trường và buổi trưa trở về nhà.

+ Số lần lặp không biết trước:

Trong một trận cầu lông các em lặp đi lặp lại công việc đánh cầu cho đến khi kết thúc trận cầu.

- Khi viết chương trình máy tính cũng vậy, trong nhiều trường hợp ta cũng phải viết lặp lại nhiều lần câu lệnh chỉ để thực hiện 1 phép tính nhất định.

## II/ Câu lệnh lặp – một lệnh thay cho nhiều câu lệnh:

- Cấu trúc lặp trong thuật toán được dùng để mô tả thực hiện lặp lại nhiều lần một hoặc một nhóm các hoạt động.
- Mọi ngôn ngữ lập trình đều có câu lệnh lặp để thể hiện cấu trúc lặp, trong đó một hoặc nhiều lệnh được viết một lần nhưng lại được điều khiển để thực hiện lặp lại một số hữu hạn lần. Các lệnh điều khiển quá trình lặp lặp như vậy gọi là câu lệnh lặp.

## III/ Ví dụ về câu lệnh lặp:

Cú Pháp câu lệnh lặp với số lần biết trước trong Pascal.

**For** <biến đếm>:= <giá trị đầu> **to** <giá trị cuối> **do** <câu lệnh>;

trong đó:

+ for, to, do là các từ khóa

+ *biến đếm* là biến đơn có kiểu nguyên

+ *giá trị đầu* và *giá trị cuối* là các biểu thức có cùng kiểu với *biến đếm* và *giá trị cuối* phải lớn hơn *giá trị đầu*

+ *câu lệnh* có thể là câu lệnh đơn giản hay câu lệnh ghép (câu lệnh ghép begin...end;)

- Số lần lặp: giá trị cuối – giá trị đầu +1

- Khi câu lệnh thành phần của lệnh lặp for...do là một câu lệnh lặp for...do khác thì ta nói rằng các câu lệnh lặp lồng nhau.

- Thoát khỏi câu lệnh lặp dùng lệnh **Break**

Vd 1: Chương trình in ra màn hình thứ tự lần lặp.

Program lap;

var i:integer;

begin

for i:= 1 to 20 do

writeln('Day la lan lap thu',i);

readln;

end.

Vd2: chương trình ghi nhận vị trí 10 chữ O rơi từ trên xuống.

```
uses crt;  
var i:integer;  
begin  
    clrscr;  
    for i:= 1 to 20 do  
        begin  
            writeln('O');  
            delay(200);  
        end;  
    readln;  
end.
```

(Delay (200) là hàm khai báo thời gian rơi nhanh hay chậm của chữ O)

\*Lưu ý: Câu lệnh đơn giản Writeln('O') và Delay(200) được đặt trong từ khoá BEGIN và AND để tạo thành câu lệnh ghép trong PASCAL

#### **IV/ Tính tổng và tích bằng câu lệnh lặp:**

Vd 1: chương trình tính tổng N số tự nhiên đầu tiên, với N là số tự nhiên được nhập từ bàn phím.

$$S = 1+2+3+ \dots + N$$

```
program Tinh_tong;  
var N,i:integer;  
    S:longint;  
begin  
    write('Nhap so N = ');  
    readln(N);  
    S:= 0;  
    for i:= 1 to N do  
        S:= S+i;  
    writeln('Tong cua ', N, ' so tu nhien dau tien S = ', S);  
    readln;  
end.
```

\*Kiểu longint có phạm vi từ  $-2^{31}$  đến  $2^{31} - 1$ .

Vd 2: chương trình tính tích N số tự nhiên, với N là số tự nhiên được nhập từ bàn phím.

$$N! = 1.2.3 \dots N$$

```
program Tinh_Giai_Thua;  
var N,i:integer;  
    P:longint;  
begin  
    write('Nhap so N = ');  
    readln(N);  
    P:= 1;  
    for i:= 1 to N do  
        P:= P*i;  
    writeln(N, '! = ', P);  
    readln;  
end.
```

## Bài tập câu lệnh lặp For...do

### - Các bài tập trong bài thực hành 5

1. Viết chương trình tính  $S=1+2+3+\dots+n$

Program BT1;

Var n,s,i:integer;

begin

write('nhap n= ');readln(n);

s:=0;

for i:=1 to n do s:=s+i;

write('tong',n:4,'so nguyen duong dau tien la',s:8);

readln;

end.

2. Viết chương trình tính  $S=1-2+3-\dots+/-n$

3. Viết chương trình tính  $S=2+4+6+\dots+n$

4. Viết chương trình tính  $S=1+3+5+\dots+n$

5. Viết chương trình tính  $S=1+1/2+1/3+\dots+1/n$

6. Viết chương trình tính  $S=1+1/3+1/5+\dots+1/n$

7. Viết chương trình tính  $S=1+1/2+1/4+\dots+1/n$

8. Viết chương trình tính  $S=1-1/2+1/3-$

9. Viết chương trình giải bài toán cổ sau: “ Vừa gà vừa chó, bó lại cho tròn, 36 con, 100 chân chẵn”

10. Viết chương trình Tính  $P=1.2.3\dots n$

program BT10;

var i,p,n:longint;

begin

writeln('tinh giai thua');

write('nhap n= ');readln(n);

p:=1;

for i:=1 to n do

p:=p\*i;

write('tich',n:4,'so dau tien la',p:9);

readln;

end.

11. Viết chương trình tính  $P=1.3.5\dots n$

12. Viết chương trình tính  $P=1.2.4\dots n$

13. Viết chương trình tính  $P=1.1/2.1/3\dots 1/n$

14. Viết chương trình tính  $p=1.1/3.1/5\dots 1/n$

15. Viết chương trình  $p=1.1/2.1/4\dots 1/n$

# Bài 8: LẶP VỚI SỐ LẦN CHƯA BIẾT TRƯỚC

## 1. Các hoạt động lặp với số lần chưa biết trước

a/ Ví dụ 1 (sgk).

b/ Ví dụ 2: Nếu cộng lần lượt  $n$  số tự nhiên đầu tiên ( $n = 1, 2, 3, \dots$ ), Cần cộng bao nhiêu số tự nhiên đầu tiên để ta nhận được tổng  $T_n$  nhỏ nhất lớn hơn 1000?

**Giải :**

Kí hiệu  $S$  là tổng cần tìm và ta có thuật toán như sau:

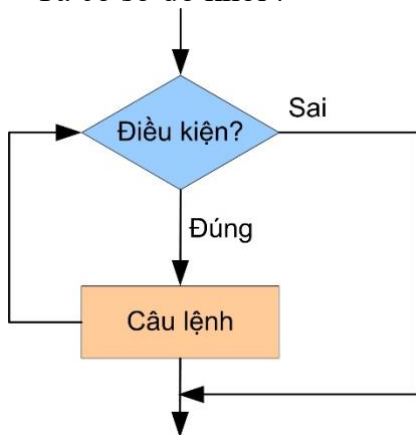
+ Bước 1.  $S \leftarrow 0, n \leftarrow 0$ .

+ Bước 2. Nếu  $S \leq 1000, n \leftarrow n + 1$ ; ngược lại chuyển tới bước 4.

+ Bước 3.  $S \leftarrow S + n$  và quay lại bước 2.

+ Bước 4. In kết quả :  $S$  và  $n$  là số tự nhiên nhỏ nhất sao cho  $S > 1000$ . Kết thúc thuật toán.

\* Ta có sơ đồ khối :



\* Nhận xét : Để viết chương trình chỉ dẫn máy tính thực hiện các hoạt động lặp như trong các ví dụ trên, ta có thể sử dụng câu lệnh có dạng *lặp với số lần chưa biết trước*

## 2. Ví dụ về lệnh lặp với số lần chưa biết trước

Trong Pascal câu lệnh lặp với số lần chưa biết trước có dạng:

**while** <điều kiện> **do** <câu lệnh>;

trong đó:

- *điều kiện* thường là một phép so sánh;
- *câu lệnh* có thể là câu lệnh đơn giản hay câu lệnh ghép.

Câu lệnh lặp này được thực hiện như sau:

Bước 1 : Kiểm tra *điều kiện*.

Bước 2 : Nếu *điều kiện* SAI, *câu lệnh* sẽ bị bỏ qua và việc thực hiện lệnh lặp kết thúc. Nếu *điều kiện* đúng, thực hiện *câu lệnh* và quay lại bước 1.

**Ví dụ 3.**

Với giá trị nào của  $n$  ( $n > 0$ ) thì  $\frac{1}{n} < 0.005$  hoặc  $\frac{1}{n} < 0.003$ ? Chương trình dưới đây tính

số  $n$  nhỏ nhất để  $\frac{1}{n}$  nhỏ hơn một sai số cho trước :

**uses crt;**

**var x: real;**

```

n: integer;
const sai_so=0.003;
begin
  clrscr;
  x:=1; n:=1;
  while x>=sai_so do begin n:=n+1; x:=1/n end;
  writeln('So n nho nhat de 1/n < ',sai_so:5:4, 'la ',n);
  readln
end.

```

**Ví dụ 4.** Chương trình Pascal dưới đây thể hiện thuật toán tính số  $n$  trong ví dụ 2:

```

var S,n: integer;
begin
  S:=0; n:=1;
  while S<=1000 do
begin n:=n+1; S:=S+n end;
  writeln('So n nho nhat de tong > 1000 la ',n);
  writeln('Tong dau tien > 1000 la ',S);
end.

```

**Ví dụ 5.** Viết chương trình tính tổng  $T = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{100}$

Giải :

Để viết chương trình tính tổng  $T = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{100}$  ta có thể sử dụng lệnh lặp với số lần lặp biết

trước **for...do**:

```

T:=0;
for i:=1 to 100 do T:=T+1/i;
writeln(T);

```

Nếu sử dụng lệnh lặp **while...do**, đoạn chương trình dưới đây cũng cho cùng một kết quả:

```

T:=0;
i:=1;
while i<=100 do begin T:=T+1/i; i:=i+1 end;
writeln(T);

```

\* Nhận xét : Ví dụ này cho thấy rằng chúng ta có thể sử dụng câu lệnh **while...do** thay cho câu lệnh **for...do**.

### 3. Lặp vô hạn lần – Lỗi lặp trình cần tránh

Khi viết chương trình sử dụng cấu trúc lặp cần chú ý tránh tạo nên vòng lặp không bao giờ kết thúc.

## Bài 9 : LÀM VIỆC VỚI DÃY SỐ

### 1. Dãy số và biến mảng

- Ví dụ 1: Nhập 50 điểm.
- Ví dụ 2: Tìm số lớn nhất trong dãy số.
- Nên gán các giá trị điểm bằng một biến và khai báo kiểu mảng dữ liệu. Biến đó được gọi là biến mảng. Giá trị của biến mảng là một dãy số nguyên hoặc số thực.

### 2. Ví dụ về biến mảng

#### a) Khai báo biến mảng

- Ví dụ: SGK
- Cấu trúc khai báo biến mảng Tên mảng:  
array [< chỉ số đầu> .. < chỉ số cuối > ] of < kiểu dữ liệu >;

#### Trong đó:

- Chỉ số đầu và chỉ số cuối là hai số nguyên.
- Chỉ số đầu <= chỉ số cuối.
- Kiểu dữ liệu có thể là số thực hoặc số nguyên.

#### b) Truy cập mảng

#### c) Nhập giá trị cho biến mảng.

- Ví dụ: Có thể gán trực tiếp hoặc nhập điểm cho 50 phần tử.  
Diem[1]:=9  
Diem[2]:=8...  
Hoặc sử dụng 50 lệnh read hoặc readln( ) để nhập điểm.  
- Ví dụ: For i:=1 to 50 do If diem[i] > 8.0 then writeln ( \_ gioi );

### 3. Tìm giá trị lớn nhất, nhỏ nhất.

\* Thuật toán tìm số lớn nhất:

Bước 1. Nhập N và dãy A1,..., An

Bước 2. Max ← A1

Lần lượt gán giá trị từ 2 đến N cho i. Với mỗi giá trị của i thì thực hiện Max←Ai

Bước 4. Đưa ra màn hình giá trị Max rồi kết thúc

### 4. Thực hành ví dụ 3:

- Ví dụ 3: SGK/78
- Ví dụ 4: Nhập 10 số nguyên tìm số lớn nhất

### **Bài 1: Nhập dãy, sắp xếp theo chiều tăng dần.**

```
Program bt;
```

```
Var
```

```
  a: array [1..10] of integer;
```

```
  p,i,j,n:integer;
```

```
  begin
```

```
writeln('nhap n= '); readln( n);
```

```
writeln( 'Nhap day so:');
```

```
for i:=1 to n do
```

```
begin
```

```
writeln( ' nhap so a [ ', i, ' ] = '); readln(a[i]);
```

```
end;
```

```
for i:=1 to n-1 do
```

```
  for j:=i+1 to n do
```

```
  if a[i] > a[j] then
```

```
  begin
```

```
    p:=a[i];
```

```
    a[i]:=a[j];
```

```
    a[j]:=p
```

```
  end;
```

```
  writeln(' Day da xep lai:');
```

```
  for i:=1 to n do write(a[i]:5);
```

```
  readln;
```

```
end.
```

### **Bài 2: Nhập dãy, sắp xếp theo chiều giảm dần. (học sinh tự làm vô tập)**

# **Bài 10: LÀM QUEN VỚI GIẢI PHẪU CƠ THỂ NGƯỜI BẰNG PHẦN MỀM ANATOMY**

## **1. Cùng làm quen với phần mềm Anatomy:**

-Mục đích của phần mềm:

+Quan sát các hệ giải phẫu cơ thể người như hệ xương, hệ cơ, hệ thần kinh,..

+Khám phá chức năng của một số bộ phận cơ thể người.

-Phần mềm có hai nút lệnh Learn (học) và Exercises(bài tập)

-Tám biểu tượng tương ứng với 8 chủ đề.

## **2. Hệ xương:**

Nháy chuột vào biểu tượng có dòng chữ SKELETAL SYSTEM để tìm hiểu về hệ xương của con người.

a) Các thao tác trực tiếp trên mô hình mô phỏng:

-Dịch chuyển

- Xoay mô hình

- Phóng to, thu nhỏ

b)Bổ sung thêm các hệ khác vào hình mô phỏng

Có thể hiển thị thêm các hệ khác.

c) Quan sát chi tiết các hệ giải phẫu cơ thể người.

- Nháy chuột vào bộ phận muốn quan sát, bộ phận này sẽ đổi màu.

- Muốn hủy nháy đúp chuột bên ngoài khu vực có mô phỏng

-Có thể ẩn bộ phận này khỏi mô hình

## **3/ Hệ cơ**

Nháy chuột vào biểu tượng có dòng chữ MUSCULAR SYSTEM để tìm hiểu về hệ cơ.

Cơ bám vào xương có chức năng co, giãn để làm cho xương chuyển động

## **4/ Hệ tuần hoàn:**

- Nháy chuột vào biểu tượng có dòng chữ CIRCULATORY SYSTEM để tìm hiểu về hệ tuần hoàn của con người.

- Chức năng giúp lưu thông máu đi khắp cơ thể để nuôi từng tế bào.

## **5/ Hệ hô hấp**

-Nháy chuột vào biểu tượng có dòng chữ RESPIRATORY SYSTEM để tìm hiểu về hệ hô hấp.

-Hệ hô hấp có chức năng đặc biệt là làm giàu oxi trong máu thông qua trao đổi chất với bên ngoài, ví dụ hít thở không khí. Thông qua histt thở, hệ hô hấp lấy Oxi đưa vào máu và sau đó lấy CO<sub>2</sub> trong máu để thải ra ngoài.

## **6/ Hệ tiêu hoá**

-Nháy chuột vào biểu tượng có dòng chữ DIGESTIVE SYSTEM để tìm hiểu hệ tiêu hoá.

- Chức năng là tiếp quản thức ăn từ miệng và tiêu hoá, hấp thụ, biến thức ăn thành năng lượng đi nuôi cơ thể.

## **7/ Hệ bài tiết**

-Nháy chuột vào biểu tượng có dòng chữ EXCRETOR SYSTEM để tìm hiểu hệ bài tiết.

-Chức năng thải các chất độc ra bên ngoài cơ thể.

## **8/ Hệ thần kinh:**

-Nháy chuột vào biểu tượng có dòng chữ NERVOUS SYSTEM để tìm hiểu về hệ thần kinh của con người.

- Các bộ phận chính của hệ thần kinh

\*\*\* **Dặn dò:**

- Học bài kết hợp SGK

- Xem và thực hành lại trên máy tính (Nếu HS ở nhà có máy).

## **BÀI 11: GIẢI TOÁN VÀ VẼ HÌNH PHẪNG VỚI GEOGEBRA**

1. Các phép tính trên đa thức
2. Các phép tính trên phân thức đại số
3. Giải phương trình và bất phương trình một ẩn
4. Quan hệ toán học và các công cụ tạo quan hệ toán học trong Geogebra
5. Các công cụ biến đổi hình học trong Geogebra
6. Công cụ đường tròn và cách vẽ một số hình đặc biệt  
(Học sinh xem SGK trang 93 →105)

## **BÀI 12: VẼ HÌNH KHÔNG GIAN VỚI GEOGEBRA**

1. Làm quen với cửa sổ không gian 3D
  2. Điểm và di chuyển điểm trong không gian
  3. Xoay hình trong không gian
  4. Vẽ hình hộp chữ nhật và lập phương
  5. Vẽ hình lăng trụ đứng
  6. Vẽ hình chóp  
(Học sinh xem SGK trang 106 →118)
- Xem và thực hành lại trên máy tính (Nếu HS ở nhà có máy).