Chương 3
CÁC CẤU TRÚC ĐIỀU KHIỂN

Khoa Hệ thống thông tin quản lý

Hà Nội – 2015

Nội dung

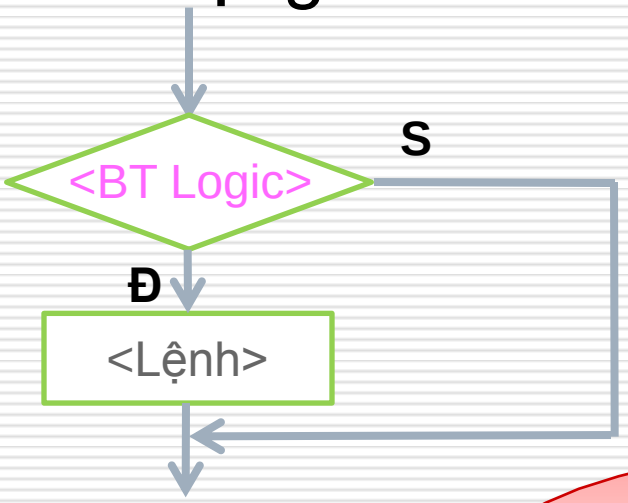
1. Cấu trúc rẽ nhánh
2. Cấu trúc lặp

1. Cấu trúc rẽ nhánh

1. Câu lệnh điều kiện if
2. Câu lệnh rẽ nhánh switch
3. Toán tử goto và nhãn
4. Bài tập thực hành

1.1 Câu lệnh điều kiện if

- Dạng thiếu




```
if (<BT Logic>)
    <Lệnh>;
```

Trong ( ), cho kết quả (sai = 0, đúng ≠ 0)

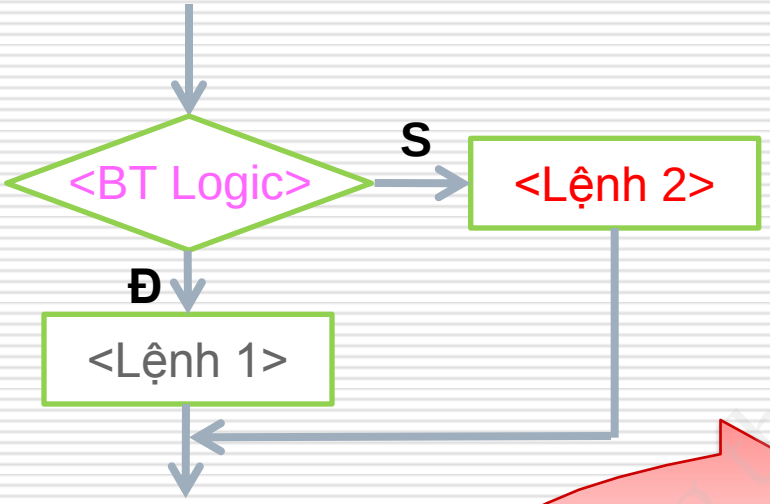
Câu lệnh đơn hoặc Câu lệnh phức (kẹp giữa { và })

Câu lệnh if (thiếu)

```
void main()
{
        if (a == 0)
                printf("a bang 0");

        if (a == 0)
        {
                printf("a bang 0");
                a = 2912;
        }
}
```

Câu lệnh if (đủ)




```
if (<BT Logic>)
        <Lệnh 1>;
else
        <Lệnh 2>;
```

Trong ( ), cho kết quả
(sai = 0, đúng ≠ 0)

Câu lệnh đơn hoặc
Câu lệnh phức (kẹp
giữa { và })

Câu lệnh if (đủ)

```
void main()
{
        if (a == 0)
                printf("a bang 0");
        else
                printf("a khac 0");

        if (a == 0)
        {
                printf("a bang 0");
                a = 2912;
        }
        else
                printf("a khac 0");
}
```

Câu lệnh if - Một số lưu ý

- Câu lệnh if và câu lệnh if… else là một câu lệnh đơn.

```
{
        if (a == 0)
                printf("a bang 0");
}
{
        if (a == 0)
        {
                printf("a bang 0");
                a = 2912;
        }
        else
                printf("a khac 0");
}
```

Câu lệnh if - Một số lưu ý

- Câu lệnh if có thể lồng vào nhau và else sẽ tương ứng với if gần nó nhất.

```
if (a != 0)
        if (b > 0)
                printf("a != 0 va b > 0");
        else
        printf("a != 0 va b <= 0");

if (a !=0)
{
        if (b > 0)
                printf("a != 0 va b > 0");
        else
                printf("a != 0 va b <= 0");
}
```

Câu lệnh if - Một số lưu ý

- Nên dùng else để loại trừ trường hợp.

```
if (delta < 0)
      printf("PT vo nghiem");
if (delta == 0)
      printf("PT co nghiem kep");
if (delta > 0)
      printf("PT co 2 nghiem");

if (delta < 0)
      printf("PT vo nghiem");
else  // delta >= 0
      if (delta == 0)
            printf("PT co nghiem kep");
      else
            printf("PT co 2 nghiem");
```

Câu lệnh if - Một số lưu ý

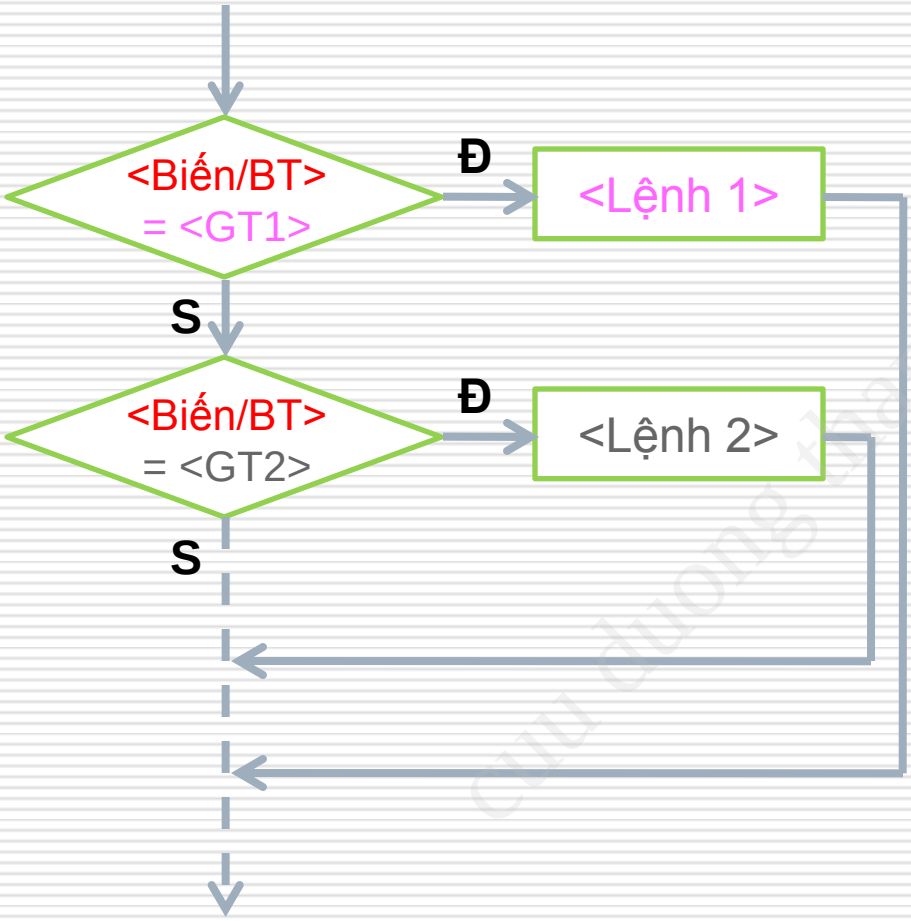
- Không được thêm ; sau điều kiện của if.

```
void main()
{
        int a = 0;
        if (a != 0)
                printf("a khac 0.");

        if (a != 0);
                printf("a khac 0.");

        if (a != 0)
        {
        };
        printf("a khac 0.");
}
```

1.2 Câu lệnh switch – Dạng thiếu




```
switch (<Biến/BT>)
{
    case <GT1>:<L1>;break;
    case <GT2>:<L2>;break;
    …
}
```

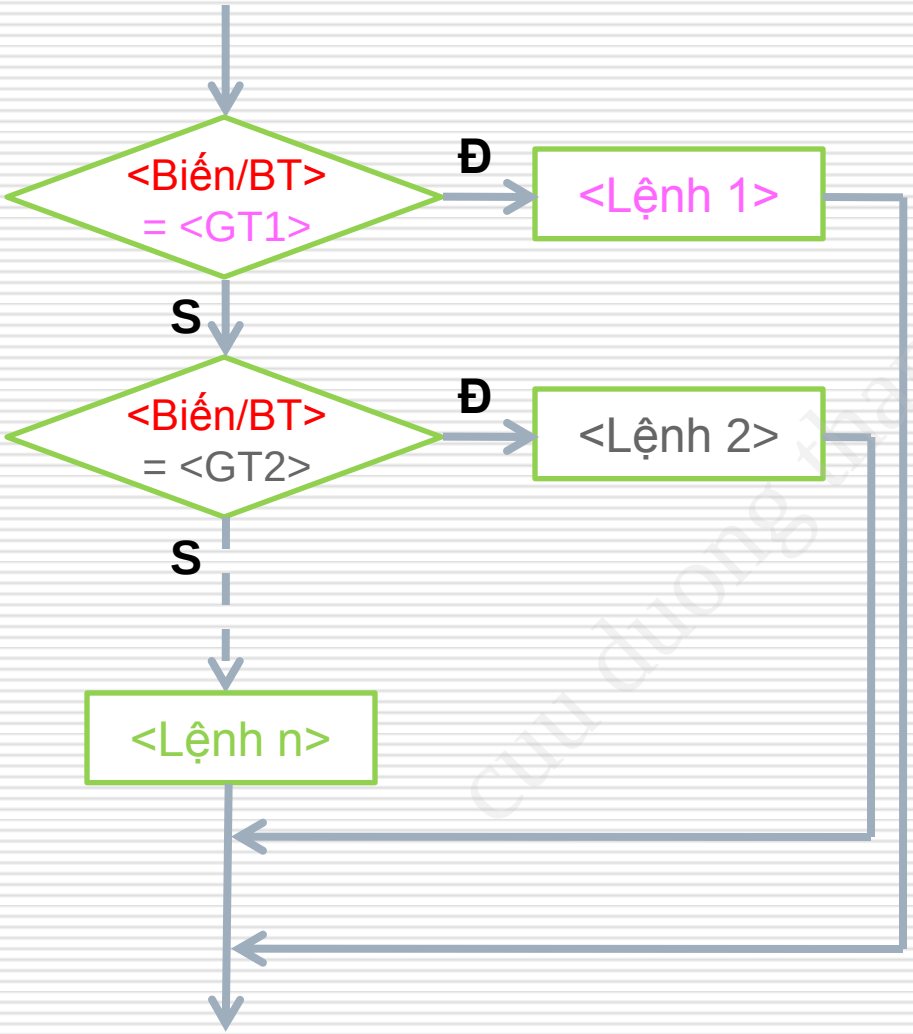
- <Biến/BT> là biến/biểu thức cho giá trị rời rạc.
- <Lệnh> : đơn hoặc khối lệnh {}

Câu lệnh switch (thiếu)

```
void main()
{
        int a;
        printf("Nhap a: ");
        scanf("%d", &a);

        switch (a)
        {
                case 1 : printf("Mot"); break;
                case 2 : printf("Hai"); break;
                case 3 : printf("Ba"); break;
        }
}
```

Câu lệnh switch (đủ)




```
switch (<Biến/BT>)
{
    case <GT1>:<Lệnh 1>;break;
    case <GT2>:<Lệnh 2>;break;
    …
    default:
        <Lệnh n>;
}
```

Câu lệnh switch (đủ)

```
void main()
{
        int a;
        printf("Nhap a: ");
        scanf("%d", &a);

        switch (a)
        {
                case 1 : printf("Mot"); break;
                case 2 : printf("Hai"); break;
                case 3 : printf("Ba"); break;
                default : printf("Ko biet doc");
        }
}
```

Câu lệnh switch - Một số lưu ý

- Câu lệnh switch là một câu lệnh đơn và có thể lồng nhau.

```
{
        switch (a)
        {
                case 1 : printf("Mot"); break;
                case 2 : switch (b)
                         {
                                case 1 : printf("A"); break;
                                case 2 : printf("B"); break;
                         } break;
                case 3 : printf("Ba"); break;
                default : printf("Khong biet doc");
        }
}
```

Câu lệnh switch - Một số lưu ý

- Các giá trị trong mỗi trường hợp phải khác nhau.

```
switch (a)
{
        case 1 : printf("Mot"); break;
        case 1 : printf("MOT"); break;
        case 2 : printf("Hai"); break;
        case 3 : printf("Ba"); break;
        case 1 : printf("1"); break;
        case 1 : printf("mot"); break;
        default : printf("Khong biet doc");
}
```

Câu lệnh switch - Một số lưu ý

- switch sẽ nhảy đến case tương ứng và thực hiện đến khi nào gặp break hoặc cuối switch sẽ kết thúc.

```
switch (a)
{
        case 1 : printf("Mot"); break;
        case 2 : printf("Hai"); break;
        case 3 : printf("Ba"); break;
}
```

Câu lệnh switch - Một số lưu ý

- switch nhảy đến case tương ứng và thực hiện đến khi nào gặp break hoặc cuối switch sẽ kết thúc.

```
switch (a)
{
        case 1 : printf("Mot"); break;
        case 2 : printf("Hai"); break;
        case 3 : printf("Ba"); break;
}
switch (a)
{
        case 1 : printf("Mot"); break;
        case 2 : printf("Hai"); break;
        case 3 : printf("Ba"); break;
}
```

Câu lệnh switch - Một số lưu ý

- Tận dụng tính chất khi bỏ break;

```
switch (a)
{
        case 1 : printf("So le"); break;
        case 2 : printf("So chan"); break;
        case 3 : printf("So le"); break;
        case 4 : printf("So chan"); break;
}
switch (a)
{
        case 1 :
        case 3 : printf("So le"); break;
        case 2 :
        case 4 : printf("So chan"); break;
}
```

1.3 Toán tử goto và nhãn

- Nhãn được viết như tên biến và có thêm dấu: (hai chấm) đứng sau, nhãn có thể được gán cho bất kì câu lệnh nào trong chương trình
- Lệnh nhảy goto có dạng:

  goto nhan;

  - Khi gặp lệnh này, máy nhảy đến nhãn viết sau từ khoá goto
- Ví dụ:

```
main()
{       int i;
        vaosl: printf("Nhap i: "); scanf("%d",&i);
        if (n<10) goto vaosl;
}
```

1.4 Bài tập thực hành

1. Nhập một số bất kỳ. Hãy đọc giá trị của số nguyên đó nếu nó có giá trị từ 1 đến 9, ngược lại thông báo không đọc được.
2. Nhập một chữ cái. Nếu là chữ thường thì đổi sang chữ hoa, ngược lại đổi sang chữ thường.
3. Giải phương trình bậc nhất $ax + b = 0$.
4. Giải phương trình bậc hai $ax^2 + bx + c = 0$.

Bài tập thực hành

5. Nhập 4 số nguyên a, b, c và d. Tìm số có giá trị nhỏ nhất (min).
6. Nhập 4 số nguyên a, b, c và d. Hãy sắp xếp giá trị của 4 số nguyên này theo thứ tự tăng dần.
7. Tính tiền đi taxi từ số km nhập vào. Biết:
   a. 1 km đầu giá 15000đ
   b. Từ km thứ 2 đến km thứ 5 giá 13500đ
   c. Từ km thứ 6 trở đi giá 11000đ
   d. Nếu trên 120km được giảm 10% tổng tiền.

Bài tập thực hành

8. Nhập vào tháng và năm. Cho biết tháng đó có bao nhiêu ngày.
9. Nhập độ dài 3 cạnh 1 tam giác. Kiểm tra đó có phải là tam giác không và là tam giác gì?

2. Cấu trúc lặp

1. Vòng lặp xác định for
2. Vòng lặp không xác định while
3. Vòng lặp không xác định do…while
4. Một số lưu ý
5. Bài tập thực hành

Đặt vấn đề

Bài toán 1. Bài toán gửi tiền tiết kiệm 1

Một người có một khoản tiền là a, đem gửi vào ngân hàng với lãi suất k mỗi tháng, sau n tháng người đó sẽ có được bao nhiêu tiền?

Đặt vấn đề (tt)

- Ví dụ:

  Số tiền gốc: 100

  Lãi suất tháng: 1%

  Số tháng gửi: 3

		1 tháng	2 tháng	3 tháng
Tiền gốc	100	101	102,01	103,0301
Tiền lãi		1	1,01	1,0201

Đặt vấn đề (tt)

- Phân tích bài toán 1:
  - Số tiền gốc ban đầu là: a
  - Tiền lãi sau tháng thứ nhất: a*k
  - ➔ Số tiền có được sau tháng thứ nhất: a + a*k
  - Số tiền này lại là số tiền gốc của tháng tiếp theo và cứ như vậy cho các tháng tiếp theo.
  - ➔ Thao tác này được lặp lại cho đến khi đủ n tháng
  - ➔ Số lần lặp là biết trước

Đặt vấn đề (tt)

Bài toán 2. Bài toán gửi tiền tiết kiệm 2

Một người có một khoản tiền là a, đem gửi vào ngân hàng với lãi suất k mỗi tháng, sau tối thiểu bao nhiêu tháng người đó sẽ có được số tiền là b đồng?

Đặt vấn đề (tt)

- Phân tích bài toán 2:
  - Việc tính tiền lãi hàng tháng vẫn lặp đi lặp lại giống như trong bài toán 1
  - Thao tác lặp sẽ dừng khi đạt được số tiền mong muốn.
  - ➔ Số lần lặp là không biết trước

2.1 Vòng lặp xác định for

- Cú pháp câu lệnh:

for ([Khởi tạo]**;** [Điều kiện]**;** [Thay đổi điều kiện]**)**

<Lệnh>**;**

- Trong đó:
  - [Khởi tạo], [Điều kiện], [Thay đổi điều kiện]: là biểu thức C bất kỳ
  - <Lệnh>: lệnh đơn hoặc khối lệnh nằm giữa { và }
  - [Khởi tạo] dùng để tạo giá trị ban đầu cho biến điều khiển
  - [Điều kiện] là biểu thức logic thể hiện điều kiện để tiếp tục vòng lặp
  - [Thay đổi điều kiện] thay đổi giá trị biến điều khiển

Cách thực hiện vòng lặp for

for ([Khởi tạo]; [Điều kiện]; [Thay đổi điều kiện])

<Lệnh>;

Bước 1. Thực hiện [Khởi tạo]

Bước 2. Xác định [Điều kiện]

Bước 3. Tuỳ thuộc vào [Điều kiện] máy lựa chọn một trong hai nhánh:

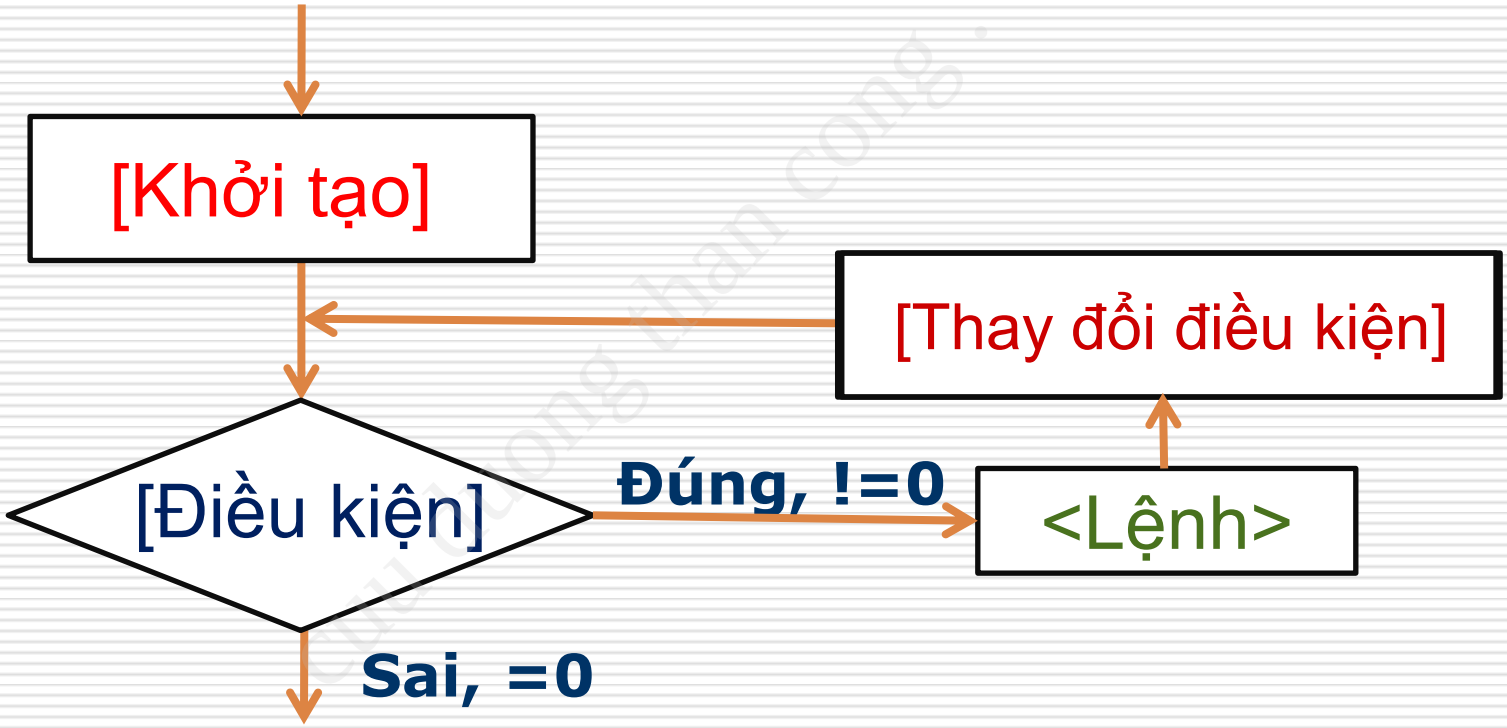
- Nếu [Điều kiện] = 0 (**Sai**), máy ra khỏi for và chuyển tới câu lệnh sau for.

- Nếu [Điều kiện]!=0 (**Đúng**), máy thực hiện <Lệnh>

Bước 4. Thực hiện [Thay đổi điều kiện] và quay lại bước 2 để bắt đầu một vòng lặp mới.

Cách thực hiện vòng lặp for (tt)

- Sơ đồ khối



Giải bài toán gửi tiền tiết kiệm 1

- Xác định INPUT/OUTPUT

```
INPUT: Số tiền ban đầu a, lãi suất k%, số tháng
cần gửi n
OUTPUT: Số tiền có được sau n tháng
```

- Thuật toán:

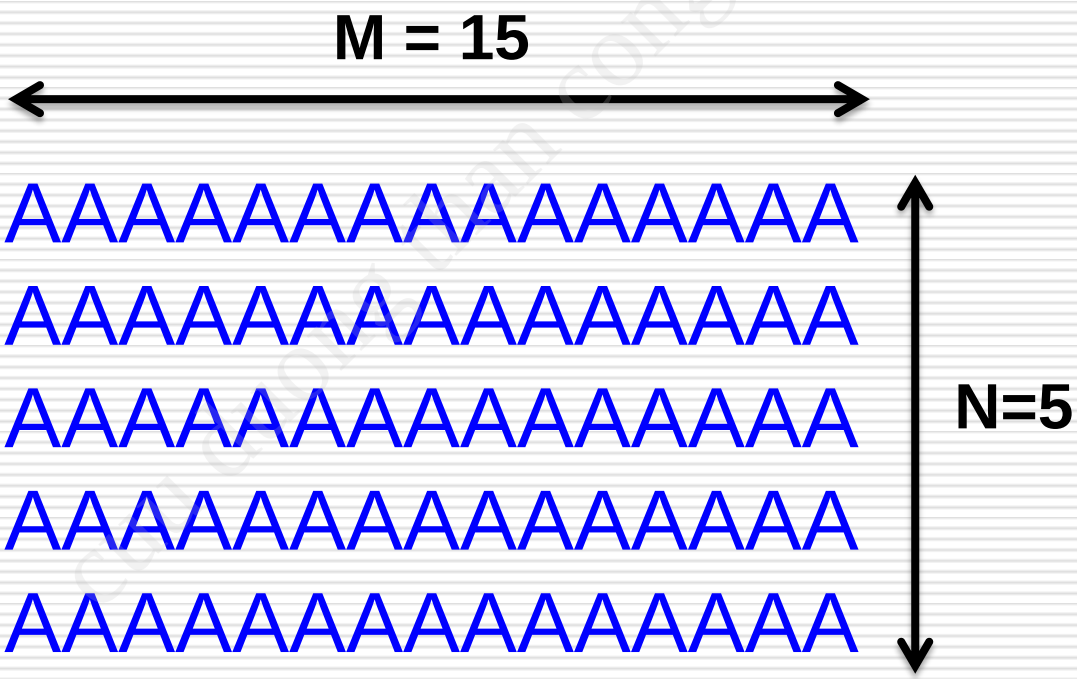
```
    c 1. Nhập số thực a, k; Số nguyên n
Bước 2. Gán thang = 1;
    c 3. Nếu thang <= n là sai,đưa ra a và kết thúc
    c 4. Gán a = a + a*k
Bước 5. Tăng số tháng: thang = thang + 1
Bước 6. Quay lại Bước 3
```

Giải bài toán gửi tiền tiết kiệm 1

```
#include <stdio.h>
#include <conio.h>
int main()
{
  float a,k;
  int thang,n;
  printf("Nhap so tien can gui: ");
  scanf("%f",&a);
  printf("Nhap lai suat: ");
  scanf("%f",&k);
  printf("Nhap so thang gui: ");
  scanf("%d",&n);
  for (thang=1;thang<=n;thang++)
       a=a+a*k;
  printf("So tien thu duoc sau %d thang la: %0.4f",n,a);
}
```

Bài toán

Bài toán 3. In hình chữ nhật gồm có kích thước N x M lên màn hình



Bài toán in hình chữ nhật

- Ý tưởng:
  - In lần lượt từng dòng, mỗi dòng in M chữ A bằng câu lệnh for

➔ Sử dụng 2 vòng for lồng nhau

```
printf("Nhap so dong: ");
scanf("%d",&N);
printf("Nhap so cot: ");
scanf("%d",&M);
for (int i = 1; i <= N ; i++)
{
        for (int j = 1; j <=M; j++)
                printf("A");
        printf("\n");
}
```

Chú ý

- Trong câu lệnh for, có thể sẽ không có phần [Khởi tạo]
- Ví dụ: Viết ra màn hình các số từ 1 đến 10

Cách viết 1:

```
int i;
for (i = 1; i <= 10; i++)
      printf("%d ", i);
```

Cách viết 2:

```
int i = 1;
for (; i <= 10; i++)
      printf("%d", i);
```

Chú ý (tt)

- Trong câu lệnh for, có thể sẽ không có phần [Điều kiện]
  - Khi đó [Điều kiện] là luôn luôn đúng
- Ví dụ: Viết ra màn hình các số từ 1 đến 10

Cách viết 1:

```
int i;
for (i = 1; i <= 10; i++)
        printf("%d ", i);
```

Cách viết 2:

```
for (i = 1; ; i++)
{
        if (i > 10) break;
        printf("%d", i);
}
```

Bài toán gửi tiền tiết kiệm 2

- Quay lại bài toán gửi tiền tiết kiệm
  - Giả sử một người gửi số tiền a với lãi suất k% thì sau bao nhiêu tháng có được số tiền là b?

➔ Không biết trước số lần lặp

➔ Sử dụng câu lệnh lặp không biết trước số lần

2.2 Vòng lặp không xác định while

- Cú pháp câu lệnh:

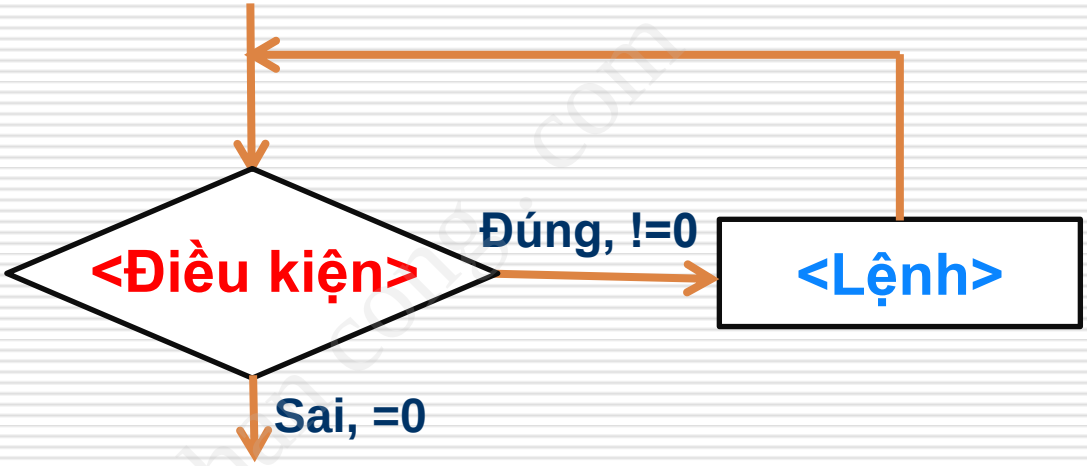
while (<Điều kiện>**)**

<Lệnh>**;**

- Trong đó:
  - <Điều kiện> Biểu thức C bất kỳ, thường là biểu thức quan hệ cho kết quả 0 (sai) và != 0 (đúng). Điều kiện có thể là một dãy biểu thức ngăn cách nhau bởi dấu phảy, tính đúng sai của <Điều kiện> là tính đúng sai của biểu thức cuối cùng trong dãy
  - <Lệnh> lệnh đơn hoặc khối lệnh nằm giữa { và }

Cách hoạt động của vòng lặp while

- Sơ đồ khối



- Cách hoạt động

Bước 1. Xác định giá trị **<Điều kiện>**

Bước 2. - Nếu **<Điều kiện>** bằng 0 (sai), máy ra ra khỏi vòng lặp.

- Nếu **<Điều kiện>** khác 0 (đúng) máy thực hiện **<Lệnh>** và trở lại **Bước 1**

Giải bài toán gửi tiền tiết kiệm 2

- Thuật toán

```
    c 1. Nhập số thực a, b và k;
Bước 2. Gán thang = 0;
    c 3. Nếu a>=b thì thông báo thang và kết
thúc;
    c 4. Gán a = a + a*k
Bước 5. Tăng số tháng: thang = thang + 1 và
quay lại Bước 3
```

Giải bài toán gửi tiền tiết kiệm

```
#include <stdio.h>
#include <conio.h>
int main()
{
  float a,b,k;
  int thang;
  printf("Nhap so tien can gui: "); scanf("%f",&a);
  printf("Nhap so tien mong muon: "); scanf("%f",&b);
  printf("Nhap lai suat: "); scanf("%f",&k);
  thang=0;
  while (a<b)
       {
         a=a+a*k;
         thang=thang+1;
       }
  printf("So thang can gui la: %d",thang);
  getch();
}
```

Chú ý

- Câu lệnh while có thể bị lặp vô tận (loop)

```
void main()
{
        int n;
        n = 1;
        while (n < 10) printf("%d", n);
}
```

```
void main()
{
        int n = 1;
        while (1)
                printf("%d", n);
}
```

  - Lặp vô hạn chỉ kết thúc được nhờ câu lệnh break

➔ Trong thân vòng lặp, cần có lệnh thay đổi giá trị <Điều kiện>

Bài toán

- Bài toán: Nhập số
  - Nhập vào một số nguyên cho đến khi số đó là số nguyên dương
- Nhận xét:
  - Công việc nhập được thực hiện lặp đi lặp lại chừng nào số nhập được còn chưa là số nguyên dương
  - Số lần lặp là không biết trước
- Thuật toán:

```
   c 1. Nhập số nguyên n
Bước 2. Nếu n>0 thông báo nhập đúng và kết thúc
   c 3. Nếu n<=0 quay lại Bước 1
```

2.3 Vòng lặp không xác định do…while

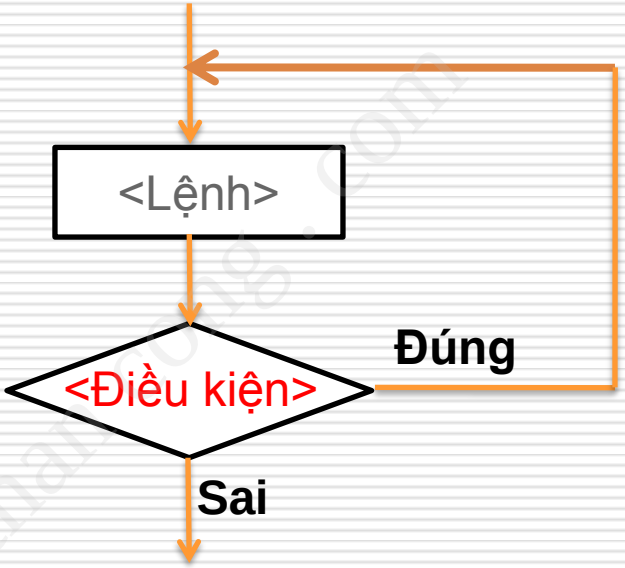
- Cú pháp

```
do
        <Lệnh>;
while (<Điều kiện>)
```

- Trong đó:
  - <Điều kiện> Biểu thức C bất kỳ, thường là biểu thức quan hệ cho kết quả 0 (sai) và != 0 (đúng). Điều kiện có thể là một dãy biểu thức ngăn cách nhau bởi dấu phảy, tính đúng sai của <Điều kiện> là tính đúng sai của biểu thức cuối cùng trong dãy
  - <Lệnh> lệnh đơn hoặc khối lệnh nằm giữa { và }

Vòng lặp không xác định do…while

- Sơ đồ khối



- Cách hoạt động

Bước 1. Thực hiện <Lệnh> trong thân vòng lặp

Bước 2. Xác định giá trị **<Điều kiện>**.

- Nếu **<Điều kiện>** bằng 0 (sai), máy ra ra khỏi vòng lặp.
- Nếu **<Điều kiện>** khác 0 (đúng) máy thực hiện **<Lệnh>** và trở lại **Bước 1**

Giải bài toán nhập số nguyên dương

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int n;
  do
      {
        printf("Hay nhap mot so >0: ");
        scanf("%d",&n);
        printf("Ban da nhap so %d\n «,n);
      }
   while (n<=0)
   printf("Dung so duong roi!");
   getch();
}
```

Chú ý

- Câu lệnh do… while có thể bị lặp vô tận (loop)

```
int n = 1;
do
        printf("%d", n);
while (n < 10);
```

```
void main()
{
        int n = 1;
        do
                printf("%d", n);
        while (1)
}
```

2.4 Một số lưu ý

- Cho phép ra khỏi for, while, do…while và switch
- Khi có nhiều chu trình lồng nhau, break đưa máy ra khỏi chu trình bên trong nhất chứa nó.
- Ví dụ: Thuật toán kiểm tra tính nguyên tố của n

```
int i,n, ng_to = 1;
for (i=2;i<=sqrt(n);i++)
   if (n%i==0)
      { ng_to=0;
        break;
      }
if (ng_to)
      printf("\n%d la so nguyen to",n);
else printf("\n%d la hop so",n);
```

Một số lưu ý (tt)

- continue dùng để bắt đầu một vòng mới của chu trình bên trong nhất chứa nó:
  - Trong thân toán tử for: máy sẽ chuyển tới bước khởi đầu lại
  - Trong thân của while hoặc do…while: máy chuyển tới xác định giá trị biểu thức (viết sau while), sau đó tiến hành kiểm tra điều kiện kết thúc chu trình
- Lưu ý: continue không áp dụng cho switch

Ví dụ câu lệnh continue

```
#include <stdio.h>
void main()
{ int i;
  for (i=1;i<=5;i++)
      { printf("\nBat dau %d",i);
        if (i<4) continue;
       printf("\nChao ban");
      }
}
```

- Kết quả:

```
Bat dau 1
Bat dau 2
Bat dau 3
Bat dau 4
Chao ban
Bat dau 5
Chao ban
```

2.5 Bài tập thực hành

1. Nhập một số nguyên dương n (n > 0).

   Hãy cho biết:

   a. Có phải là số đối xứng? Ví dụ: 121, 12321, …
   b. Có phải là số chính phương? Ví dụ: 4, 9, 16, …
   c. Có phải là số nguyên tố? Ví dụ: 2, 3, 5, 7, …
   d. Chữ số lớn nhất và nhỏ nhất?
   e. Các chữ số có tăng dần hay giảm dần không?

Bài tập thực hành

2. Nhập một số nguyên dương n. Tính:
   a. $S = 1 + 2 + \ldots + n$
   b. $S = 1^2 + 2^2 + \ldots + n^2$
   c. $S = 1 + 1/2 + \ldots + 1/n$
   d. $S = 1*2*\ldots*n = n!$
   e. $S = 1! + 2! + \ldots + n!$
3. Nhập 3 số nguyên a, b và n với a, b < n. Tính tổng các số nguyên dương nhỏ hơn n chia hết cho a nhưng không chia hết cho b.
4. Tính tổng các số nguyên tố nhỏ hơn n (0 < n < 50000)

Bài tập thực hành

5. Nhập một số nguyên dương n. Xuất ra số ngược lại. Ví dụ: Nhập 1706 ➔ Xuất 6071.
6. Tìm và in lên màn hình tất cả các số nguyên trong phạm vi từ 10 đến 99 sao cho tích của 2 chữ số bằng 2 lần tổng của 2 chữ số đó.
7. Tìm bội số chung lớn nhất của 2 số nguyên dương a và b nhập từ bàn phím.
8. Nhập n. In n số đầu tiên trong dãy Fibonacy biết

$F_0=F_1=1$

$F_n = F_{n-1} + F_{n-2}$