KỸ THUẬT LẬP TRÌNH C/C++
Cấu trúc điều khiển

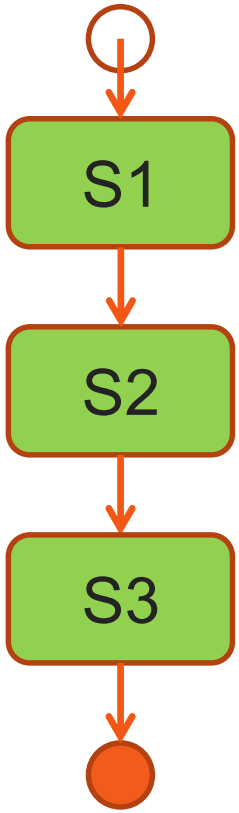
Thi-Lan Le
Thi-Lan.Le@mica.edu.vn; lan.lethi1@hust.edu.vn
Webpage: http://www.mica.edu.vn/perso/Le-Thi-Lan

Giới thiệu

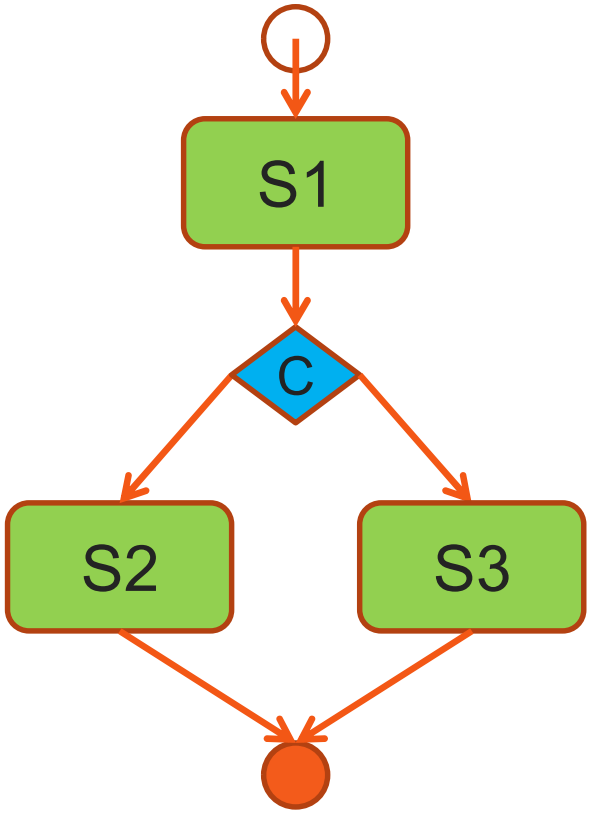
Có 3 loại cấu trúc điều khiển các lệnh cơ bản:

- **Cấu trúc tuần tự** : là cách tổ chức các lệnh thành từng khối. Phần cấu trúc khối lệnh đã được trình bầy trong chương 1.
- **Cấu trúc rẽ nhánh**: có các cấu trúc *if và switch.*
- **Cấu trúc lặp** : có các cấu trúc *for, while, do while.*

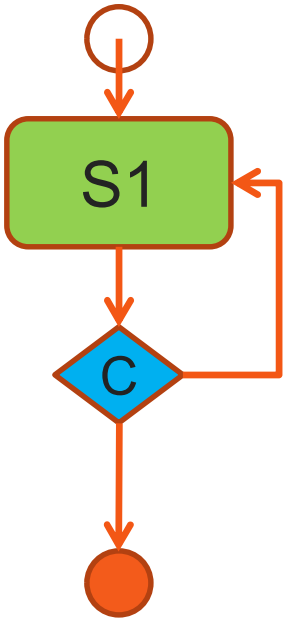
1. Giới thiệu: các cấu trúc điều khiển



Cấu trúc tuần tự



Cấu trúc rẽ nhánh



Cấu trúc lặp

2. Các cấu trúc rẽ nhánh
1. Giới thiệu

- Cấu trúc rẽ nhánh có thể chia làm hai loại:
- - Cấu trúc rẽ một trong hai nhánh : như cấu trúc *if*, *if..else* và lệnh *(? :)*.
- - Cấu trúc rẽ một, hai hoặc nhiều nhánh : cấu trúc *switch..case.*
- Trong hai cấu trúc này thì cấu trúc hai nhánh tổng quát hơn vì nó có thể áp dụng cho mọi loại biểu thức điều kiện rẽ nhánh và cấu trúc này cho phép lồng nhau để tạo thành các cấu trúc rẽ nhiều nhánh. Còn cấu trúc rẽ nhiều nhánh switch chỉ có thể áp dụng với biểu thức điều kiện rẽ nhánh kiểu số nguyên.

2. Các cấu trúc rẽ nhánh
1. Giới thiệu

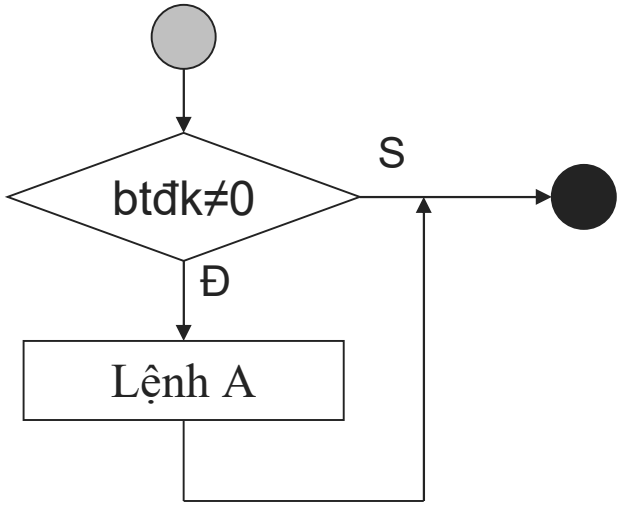
- Biểu thức điều kiện (*btđk)*
- Là biểu thức chứa các toán tử logic, biến/hằng logic
- Trả về kết quả 1 (true) hoặc 0 (false)
- Trong C, kiểu int có thể được ngầm hiểu là kiểu logic với việc chuyển đổi: 0 → false, khác 0 → true
  - Hệ quả: so sánh một số nguyên với 0 có thể bỏ qua trong trong các biểu thức logic:
  - `if (x != 0) …` → `if (x) …`
- Ví dụ:
  - `8*4 >= 10`
  - `x != y`
  - `b*b > 4*a*c`
  - `(a>2) && ((b<3) || (a>4)))`
  - `2-3 /* được ngầm chuyển thành true */`

2. Các cấu trúc rẽ nhánh
2.1 Lệnh *if*

- Dạng 1:
- Cú pháp:

if *(btđk) Lệnh A ;*

- Ý nghĩa:

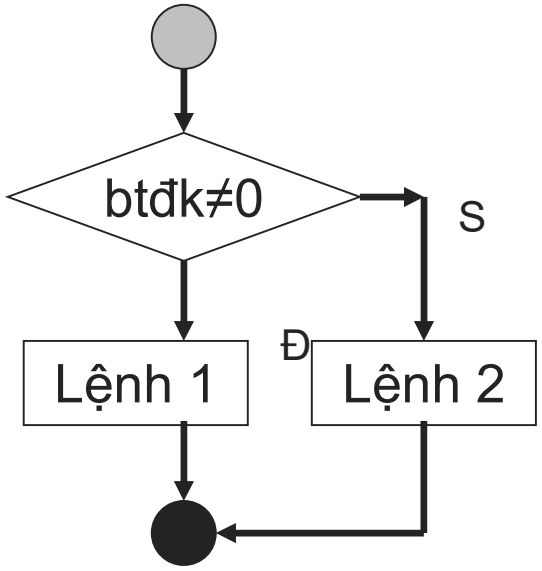


2. Các cấu trúc rẽ nhánh
2.1 Lệnh *if*

- Dạng 2:
  - Cú pháp:

    if *(btđk) lệnh 1 ;*

    else *lệnh 2 ;*
  - Ý nghĩa:



2. Các cấu trúc rẽ nhánh
2.1 Lệnh *if*

- Dạng 3: Hàm (? :)
- Cú pháp:

(btđk) **?** <bt1> **:** <bt2> ;

- Ý nghĩa: hàm trên tương đương với các lệnh:
  - *if (btđk) return bt1 ;*
  - *else return bt2 ;*

2. Các cấu trúc rẽ nhánh
2.1 Lệnh *if*

VD1: Viết chương trình tìm giá trị bé nhất của ba số a, b, c cho trước sử dụng cấu trúc if

2. Các cấu trúc rẽ nhánh
2.1 Lệnh *if*

VD1: Viết chương trình tìm giá trị bé nhất của ba số a, b, c cho trước.

```
#include <stdio.h>

void main(){
    int  a = 10, b = 15, c= 8;
    int  m;
```

```
//Cách 1
    m = a;
    if (b < m) m = b;
    if (c < m) m = c;
```

```
//Cách 2
if (a<b)
    if (a<c) m=a;
    else m=c;
else
    if (b<c) m=b;
    else m=c;
```

```
//Cách 3
m= (a<b) ? ((a<c) ?a:c) :
      ((b<c)? b : c);
```

```
printf ("Gia tri be nhat m = %d", m);
}// end main
```

2. Các cấu trúc rẽ nhánh
2.1 Lệnh *switch..case*

- Cú pháp:

```
switch (biểu thức điều kiện) {
   case hằng số 1 : câu lệnh 1
   case hằng số 2 : câu lệnh 2
   …
   [default : câu lệnh nhánh default]
}
```

- **Biểu thức điều kiện** là một biểu thức số học nhận giá trị nguyên.
- Hằng số 1, hằng số 2,… là các hằng số chọn kiểu số nguyên khác nhau, tương ứng cho các nhánh chọn *case* khác nhau. Đây là các hằng số mà giá trị biểu thức điều kiện có thể nhận.
- Nhánh *default* là nhánh lựa chọn mặc định khi không có nhánh nào khác được chọn. Nhánh này là không bắt buộc phải có.

2. Các cấu trúc rẽ nhánh
2.1 Lệnh *switch..case*

Ý nghĩa:

- Bước 1 : tính giá trị biểu thức điều kiện
- Bước 2 : so sánh giá trị này với các hằng số trong các nhánh *case*. Nếu giá trị này bằng với hằng số chọn trong nhánh case nào thì câu lệnh trong nhánh đấy được thực hiện. Nếu không có nhánh case nào được thực hiện và có nhánh *default* thì câu lệnh nhánh *default* sẽ được thực hiện.

Lưu ý:

- Để kết thúc việc thi hành của một nhánh chọn của cấu trúc *switch*, ta phải có lệnh *break* ở cuối của nhánh đó.
- Nếu không có lệnh *break*, chương trình sẽ tiếp tục được thi hành ở nhánh kế tiếp. Điều này được áp dụng khi có nhiều giá trị của biểu thức điều kiện cùng áp dụng cho một trường hợp.

2. Các cấu trúc rẽ nhánh
2.1 Lệnh *switch..case*

- Ví dụ:

```
#include <stdio.h>
#include <conio.h>
void main() {
        char ch;
        printf("Nhap gia tri ch=");
        scanf("%c",&ch);
        switch (ch){
                case 'a': printf("Ki tu a da duoc nhap");break;
                case 'b': printf("Ki tu b da duoc nhap");break;
                default: printf("Ki tu khac a va b da duoc nhap");
        }
        getch();
}
```

3. Các cấu trúc lặp

- Lệnh ***for***
- Lệnh ***while***
- Lệnh ***do … while***

Lệnh *for*

- Cú pháp:

for (lệnh 1; btđk; lệnh 2) lệnh 3 ;

- Trong đó:

+ Lệnh 1: lệnh khởi tạo giá trị ban đầu cho biến chạy

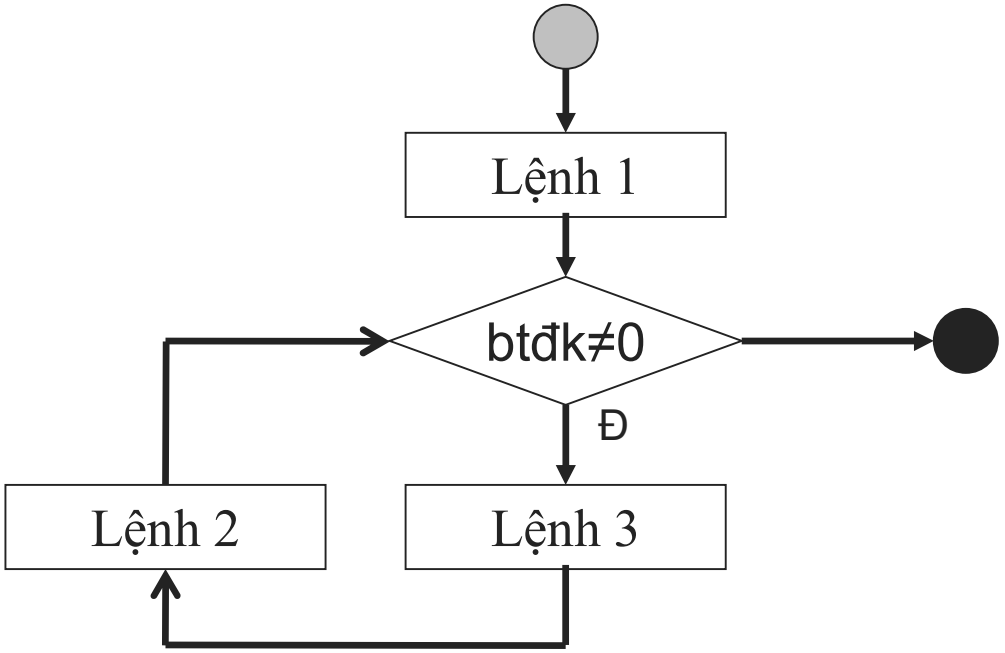
+ btđk: là biểu thức logic xác định điểm dừng của vòng lặp. Chừng nào btđk<>0 thì còn thi hành vòng lặp.

+ lệnh 2: là lệnh thay đổi giá trị biến chạy

+ lệnh 3: lệnh thân vòng lặp.

Lệnh *for*

- **Ý nghĩa hoạt động**
  - **for (lệnh 1; btđk; lệnh 2) lệnh 3 ;**



Lệnh ***for*** – Ví dụ 1: Tính căn bậc 2 của 10 số nguyên dương đầu tiên

```
#include <stdio.h>
#include <conio.h>
#include <math.h> /*Thu vien toan hoc, chua ham sqrt tinh can bac 2*/
void  main()
{  int  i;
   float  kqua;

   for (i=1 ; i<=10 ; i++) {
      kqua = sqrt(i);
      printf("Can bac 2 cua %d = %f \n", i, kqua);
   }
   getch();
}            //Ket thuc chuong trinh
```

Lệnh ***for*** – Ví dụ 2: Tính căn bậc 2 của một số nhập từ bàn phím

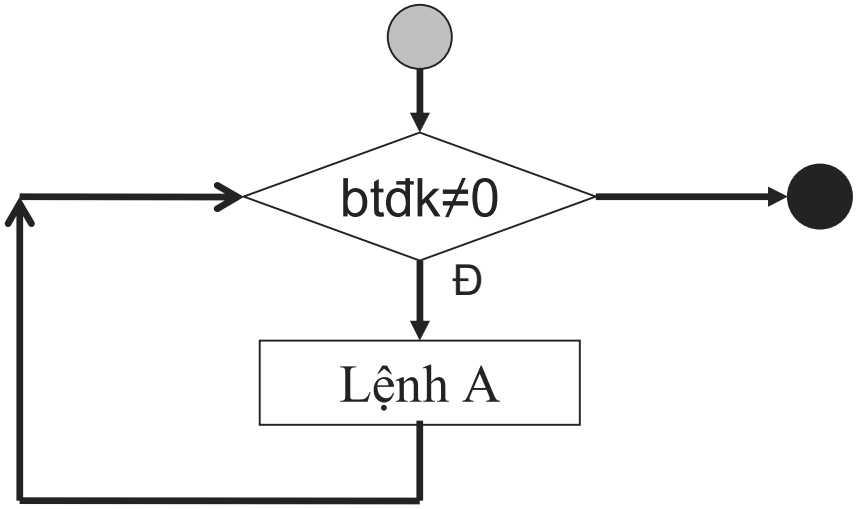
```
#include <stdio.h>
#include <math.h> /*Thu vien toan hoc, chua ham tinh can bac 2*/
main()
{  int  x;
   float  ketqua;
   int i;
   char  ch;
   for (; ;) {
      printf("\nHay nhap mot so nguyen :");
      scanf("%d",&x);
      ketqua = pow(x,1.0/2);
      printf("Can bac 2 cua %d = %f \n", x, ketqua);
      printf("Tiep tuc khong? Y/N");
      ch= getche();
      if (ch == 'n' || ch == 'N') break;
   }
} //Ket thuc chuong trinh
```

Lệnh ***while***

- Cú pháp:

while (btđk) lệnh A ;

- Nguyên tắc hoạt động



Lệnh ***while***
Ví dụ: tính BSCNN của 2 số

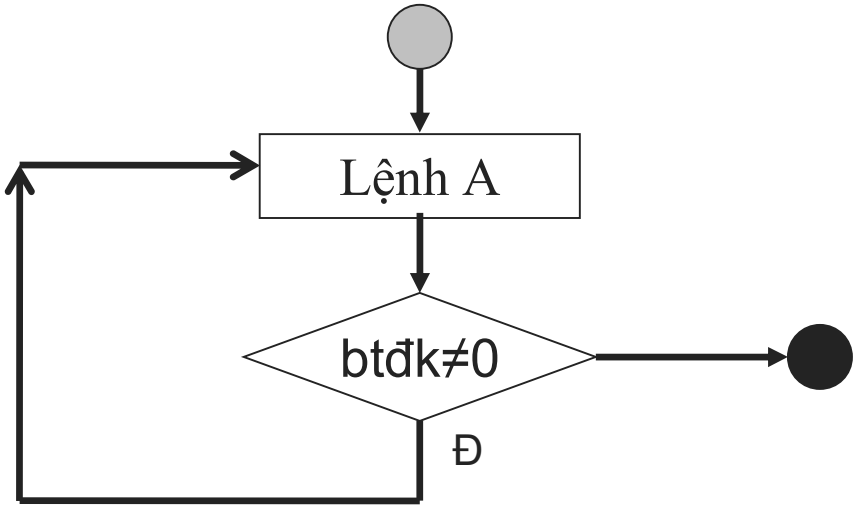
```
#include <stdio.h>
#include <conio.h>
void main()             {
   unsigned int a,b,x,y;
   printf("nhap 2 so x, y : ");
   scanf("%u%u", &x,&y);
   a = x;   b = y;
   if (a*b==0)
      printf("Khong tim bsc cua 0");
   else {
      while (a!=b)
         if (a>b) a -= b;
         else b -= a;
      printf("boi so chung nho nhat la %u", x*y/a);
   }
   getch();
}
```

Lệnh *do … while*

- Cú pháp:

do lệnh A **while** (btđk) ;

- Nguyên tắc hoạt động



Lệnh ***do … while*** - Ví dụ: Tính căn bậc 2 của 10 số nguyên dương đầu tiên

```
#include <stdio.h>
#include <conio.h>
#include <math.h> /*Thu vien toan hoc, chua ham sqrt tinh can bac 2*/
main()
{  int  i=1;
   float  kqua;
   do {
      kqua = sqrt(i);
      printf("Can bac 2 cua %d = %f \n", i, kqua);
      i++;
   } while (i<=10);
   getch();
}            //Ket thuc chuong trinh
```

Lệnh ***break*** và ***continue***

- Lệnh **break**: cho phép kết thúc vòng lặp ngay lập tức (bỏ qua bước kiểm tra điều kiện)

- Lệnh **continue**: cho phép bỏ qua các lệnh phía sau lệnh này và ngay lập tức quay lại đầu vòng lặp.

Tóm tắt về các cấu trúc điều khiển

- Cấu trúc tuần tự: khối lệnh {…}
- Cấu trúc rẽ nhánh:
  - 1 nhánh: **if**
  - 2 nhánh: **if** .. **else** ..
  - 3 nhánh trở lên: **switch** .. **case** ..
- Cấu trúc lặp:
  - Số bước lặp xác định trước: **for** (..) ..
  - Số bước lặp không xác định trước:
    - Số bước lặp > 0: **do** .. **while** ..
    - Số bước lặp >= 0: **while** .. **do** ..

Xin cảm ơn!