



TRƯỜNG ĐẠI HỌC THỦY LỢI
Khoa Công nghệ thông tin
Bộ môn Tin học và KTTT

NHẬP MÔN LẬP TRÌNH

INTRODUCTION TO COMPUTER PROGRAMMING

Giảng viên: TS.GVC Bùi Thị Thanh Xuân

Email: xuanbtt@tlu.edu.vn

Điện thoại: 0902001581

- ❖ Số tín chỉ: 3 (2-0-1)
- ❖ Số tiết: 45 tiết (30 LT và 15 TH)
- ❖ Học phần bắt buộc cho ngành: TH, HT, PM
- ❖ Điều kiện dự thi kết thúc HP: Nghỉ học không quá 20% số tiết (Nghỉ không quá 3 buổi)
- ❖ Nội dung tóm tắt: Giới thiệu về máy tính và cách máy tính làm việc; làm quen với lập trình cơ bản mình họa bằng ngôn ngữ C. Giải quyết và cài đặt thuật giải một số vấn đề đơn giản trong Toán học, Khoa học và Kỹ thuật.
- ❖ Trọng số điểm môn học:
 - Điểm quá trình: 50%
 - Điểm thi kết thúc HP: 50%



Chương 3.

Các lệnh lựa chọn và lặp

3.1. Lệnh khối

3.2. Lệnh rẽ nhánh

3.3. Lệnh lựa chọn

3.4. Các lệnh lặp

- Vòng lặp for
- Vòng lặp while và do while

3.5. Các lệnh nhảy

- Câu lệnh continue
- Câu lệnh break

❖ Lệnh đơn:

- Là biểu thức theo sau bởi dấu `;`
- Ví dụ: `x = 0; i++; printf("Hello");`

❖ Lệnh ghép (khối lệnh)

- Là tập hợp các câu lệnh (đơn và ghép) được đặt trong cặp ngoặc nhọn `{ }`
- C cho phép khai báo biến trong một khối lệnh
 - Phần khai báo phải nằm trước các câu lệnh
- Chú ý:
 - Lệnh ghép có thể đặt tại bất cứ chỗ nào mà cú pháp cho phép đặt 1 câu lệnh đơn
 - Không đặt dấu `;` sau một khối lệnh đặt trong ngoặc nhọn

❖ Trong ***lệnh ghép*** có thể ***chứa lệnh ghép khác***

❖ Có thể khai báo biến trong khối lệnh

```
{//Khai báo đối tượng cục bộ trong khối lệnh;
```

```
    {//Khai báo đối tượng cục bộ trong khối lệnh;
```

```
        ...
```

```
    }
```

```
    ...
```

```
}
```

Nếu các đối tượng được khai báo trùng tên nhau ?

```
1  #include <stdio.h>
2  int main(){ // ham main() cung la mot khoi lenh
3      int c = 10, d= 20;
4      printf("Bien ngoai khoi c = %d; d = %d", c,d);
5      {
6          int c = 10;
7          printf("\nBien trong khoi c = %d; d = %d", c,d);
8          printf("\nGia tri cua cac bien duoc them 10 don vi");
9          c = c + 10; d = d + 10;
10         printf("\nBien trong khoi c = %d; d = %d",c,d);
11     }
12     printf("\nBien ra ngoai khoi c = %d; d = %d \n",c,d);
13     return 0;
14 }
```

```
Bien ngoai khoi c = 10; d = 20
Bien trong khoi c = 10; d = 20
Gia tri cua cac bien duoc them 10 don vi
Bien trong khoi c = 20; d = 30
Bien ra ngoai khoi c = 10; d = 30
```

3.1. Lệnh khối

3.2. Lệnh rẽ nhánh

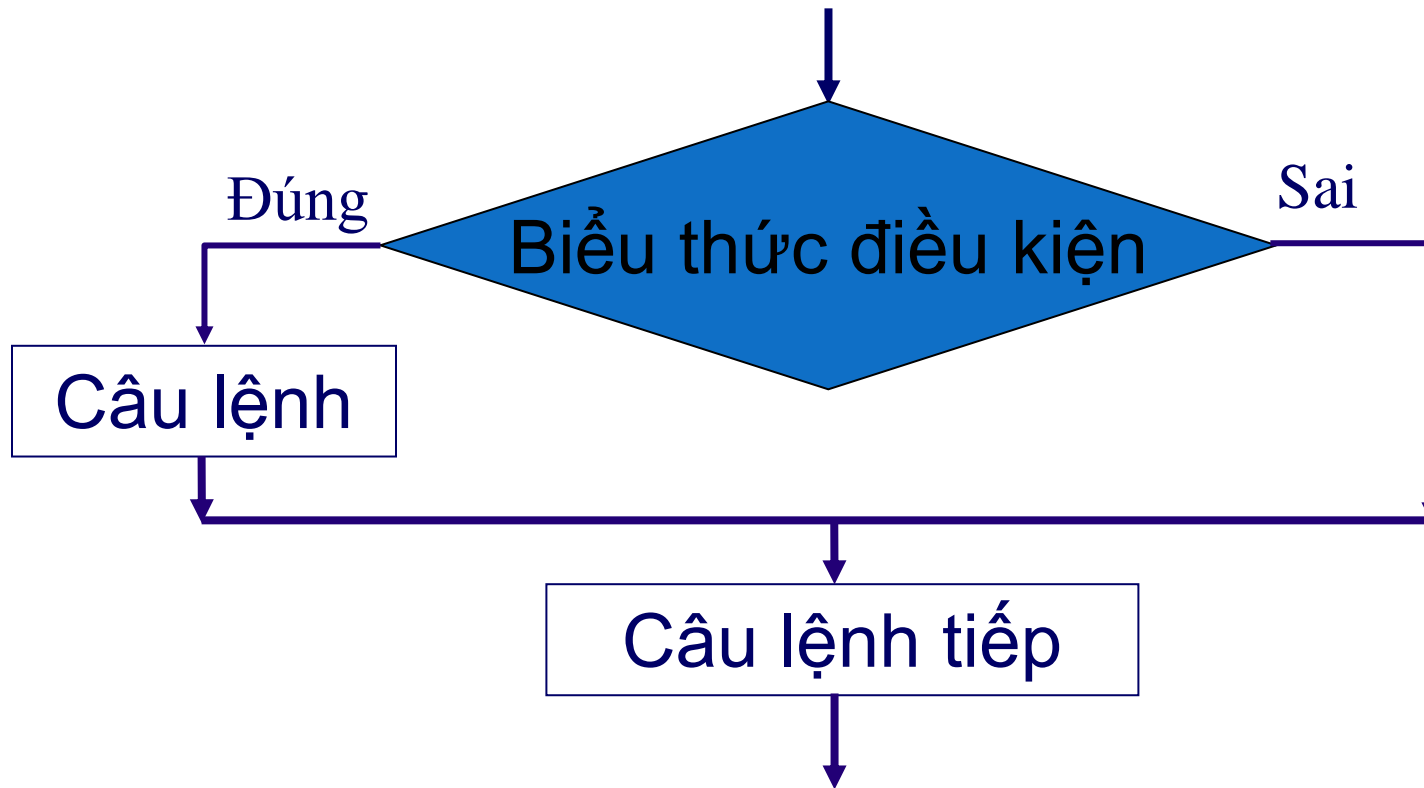
3.3. Lệnh lựa chọn

3.4. Các lệnh lặp

- Vòng lặp for
- Vòng lặp while và do while

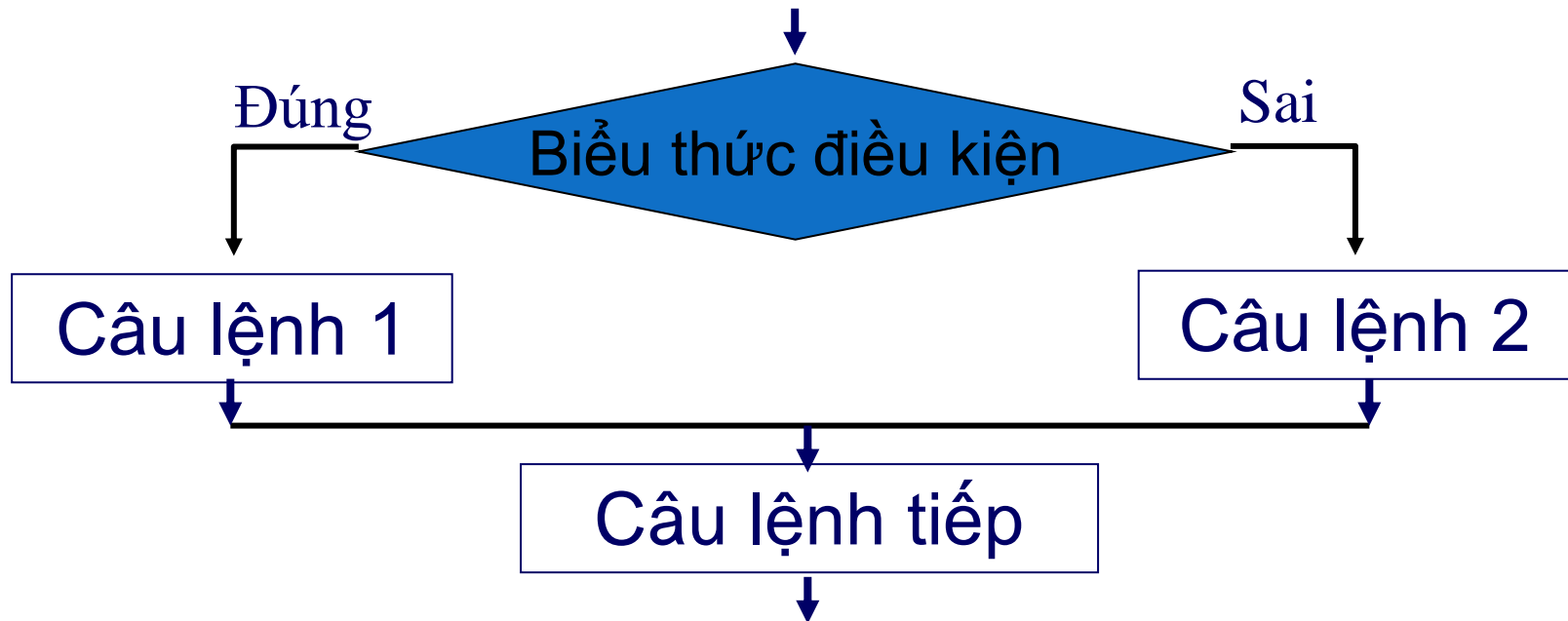
3.5. Các lệnh nhảy

- Câu lệnh continue
- Câu lệnh break



if (**Biểu thức điều kiện**)
Câu lệnh;
Câu lệnh kế tiếp;

if ($n \% 2 == 0$)
printf("so chan");



```
if (Biểu thức điều kiện)  
    Câu lệnh 1;  
else  
    Câu lệnh 2;  
Câu lệnh kế tiếp;
```

```
if (x > y)  
    z = x;  
else  
    z = y;  
    printf("max: %d", z);
```

Biểu thức điều kiện:

- Là biểu thức trả về giá trị logic đúng/sai
 - Giá trị logic đúng/True : khác 0
 - Giá trị logic sai/False: bằng 0

Ví dụ:

if (2+5) printf("Hello world! "); → Chấp nhận

Câu lệnh: Có thể là một khối lệnh (*Đặt trong cặp { }*)

```
1 #include <stdio.h>
2 int main(){
3     if(2)
4         printf("Xin chao!");
5     else
6         printf("Tam biet!");
7     return 0;
8 }
```

```
Xin chao!
-----
Process exited
Press any key to
```

```
1 #include <stdio.h>
2 int main(){
3     if(1-1)
4         printf("Xin chao!");
5     else
6         printf("Tam biet!");
7     return 0;
8 }
```

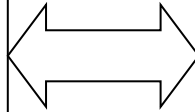
```
Tam biet!
-----
Process exited
Press any key to
```

Ví dụ: Tìm giá trị lớn nhất trong 2 số thực

```
1  #include<stdio.h>
2  #include<math.h>
3  int main()
4  {
5      float a, b, max; //khai bao bien
6      printf("Nhap a = "); scanf("%f", &a);
7      printf("Nhap b = "); scanf("%f", &b);
8      if (a < b)
9          max = b;
10     else
11         max = a;
12     printf("So lon nhat trong 2 so %.4f va %.4f la: %.4f", a, b, max);
13     return 0;
14 }
```

```
Nhap a = 4.5
Nhap b = 7.0
So lon nhat trong 2 so 4.5000 va 7.0000 la: 7.0000
-----
Process exited after 10.8 seconds with return value 0
Press any key to continue . . .
```

```
if(a < b)
    max = b;
else
    max = a;
```



```
max = a > b ? a : b;
```

Ví dụ: Giải phương trình $ax + b = 0$

```
1  #include <stdio.h>
2  float a, b, x;
3  int main(){
4      printf("Cho he so cua phuong trinh ax + b = 0.\n");
5      printf("a = "); scanf("%f",&a);
6      printf("b = "); scanf("%f",&b);
7      if(a == 0 && b == 0)
8          printf("PT vo so nghiem!");
9      if(a == 0 && b != 0)
10         printf("PT khong co nghiem!");
11     if(a != 0)
12         printf("PT co nghiem x = %f", -b/a);
13     return 0;
14 }
```

Ví dụ: Giải phương trình $ax + b = 0$

```
1  #include <stdio.h>
2  float a, b, x;
3  int main(){
4      printf("Cho he so cua phuong trinh ax + b = 0.\n");
5      printf("a = "); scanf("%f",&a);
6      printf("b = "); scanf("%f",&b);
7      if(a == 0 && b == 0)
8          printf("PT vo so nghiem!");
9      if(a == 0 && b != 0)
10         printf("PT khong co nghiem!");
11     if(a != 0)
12         printf("PT co nghiem x = %f", -b/a);
13     return 0;
14 }
```

```
1  #include <stdio.h>
2  int main(){
3      float a,b;
4      printf("Giai phuong trinh bac nhat ax + b = 0");
5      printf("\nNhap he so a b: "); scanf("%f %f", &a, &b);
6      if(a == 0)
7          if(b != 0)
8              printf("Phuong trinh vo nghiem");
9          else
10             printf("Phuong trinh vo so nghiem");
11     else
12         printf("Nghiem x = %g", -b/a);
13     return 0;
14 }
```


Ví dụ: Giải phương trình $ax + b = 0$

```
1  #include <stdio.h>
2  int main(){
3      float a,b;
4      printf("Giai phuong trinh bac nhat ax + b = 0");
5      printf("\nNhap he so a: "); scanf("%f", &a);
6      printf("Nhap he so b: "); scanf("%f", &b);
7      if(a == 0)
8          if(b != 0)
9              printf("Phuong trinh vo nghiem!");
10         else
11             printf("Phuong trinh vo so nghiem!");
12     else
13         printf("Nghiem x = %g", -b/a);
14     return 0;
15 }
```

```
Giai phuong trinh bac nhat ax + b = 0
Nhap he so a: 3.5
Nhap he so b: 7.0
Nghiem x = -2
-----
Process exited after 5.467 seconds with return value 0
Press any key to continue . . .
```

Ví dụ: Nhập số thực x và tính hàm

$$f(x) = \begin{cases} x^2 + \sin^4 2\pi x + 1 & \text{khi } x < 3 \\ 5 & \text{khi } x = 3 \\ \sqrt{x-3} + \log_{10}(x^2 - 3) & \text{khi } x > 3 \end{cases}$$

```
Nhap x = 3
Ket qua: 5.0000
-----
```

```
Nhap x = 2.5
Ket qua: 7.2500
-----
```

```
Nhap x = 3.5
Ket qua: 1.6732
-----
```

Ví dụ: Nhập số thực x và tính hàm

$$f(x) = \begin{cases} x^2 + \sin^4 2\pi x + 1 & \text{khi } x < 3 \\ 5 & \text{khi } x = 3 \\ \sqrt{x-3} + \log_{10}(x^2 - 3) & \text{khi } x > 3 \end{cases}$$

```
Nhap x = 3
Ket qua: 5.0000
```

```
Nhap x = 2.5
Ket qua: 7.2500
```

```
Nhap x = 3.5
Ket qua: 1.6732
```

```
1  #include <stdio.h>
2  #include <math.h>
3  int main()
4  {
5      float x, fx;
6      printf("Nhap x = "); scanf("%f",&x);
7      if(x < 3)
8          fx = x*x+pow(sin(2*M_PI*x),4)+1;
9      else
10         if(x==3)
11             fx = 5;
12         else
13             fx = sqrt(x-3) + log10(x*x-3);
14     printf("Ket qua: %.4f",fx);
15     return 0;
16 }
```

Viết chương trình nhập vào một số thực x. Tính và in ra màn hình giá trị của biểu thức

$$f(x) = x^2 + \sqrt[5]{x} + 3e^x \text{ nếu } x \neq 0,$$

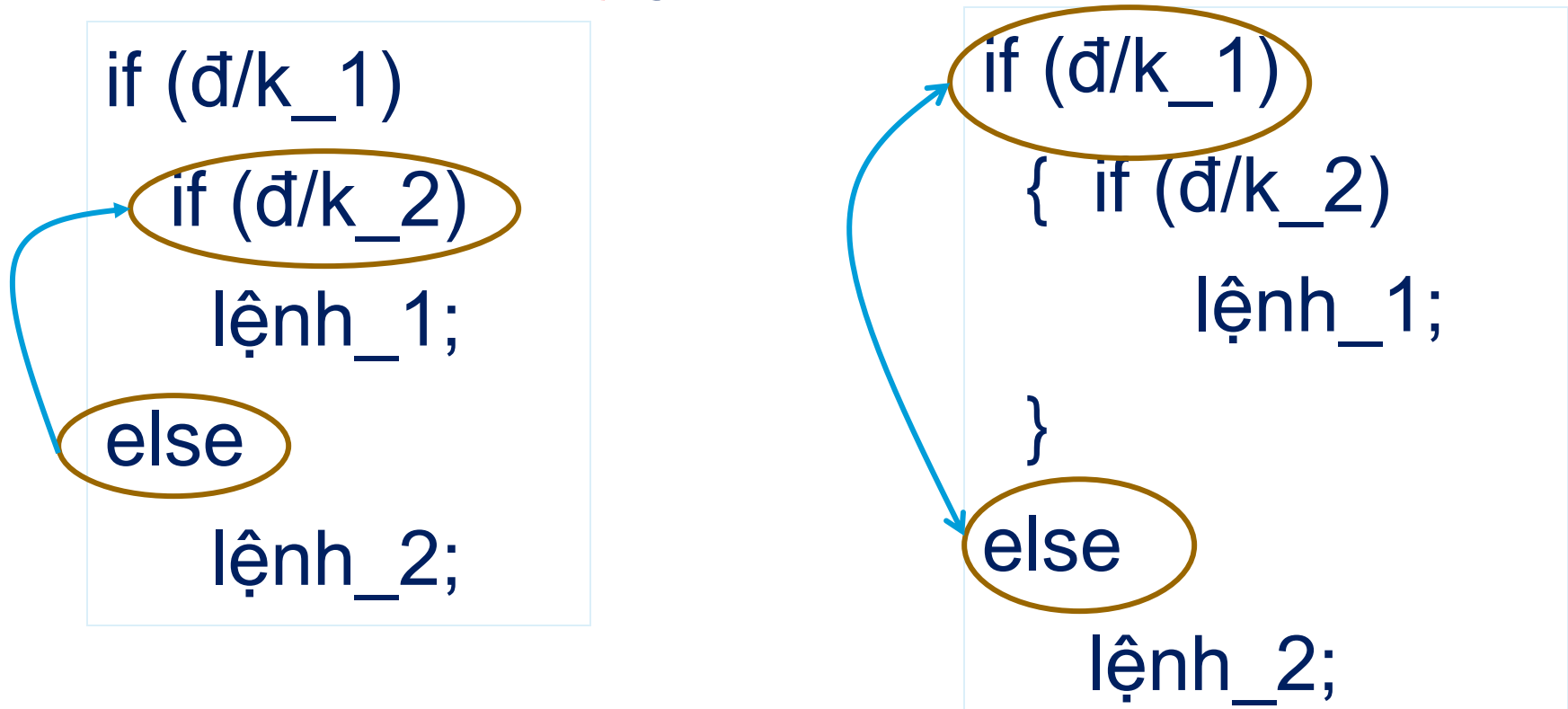
ngược lại $f(0) = 4$.

Viết chương trình nhập vào một số thực x. Tính và in ra màn hình giá trị của biểu thức

$f(x) = x^5 + \sqrt{x - 1} + 3e^x$ nếu x hợp lệ, ngược lại in ra thông báo “f(x) không xác định!”

Cấu trúc if.. và if ...else có thể lồng nhau

- Khi đó else sẽ tương ứng với *if (phía trên, chưa có else)* gần nhất



```
int a, b, c = 10;
```

```
if (a==0)
    if (b == 0)
        c = 20;
else
    c = 30;
```

$a \neq 0, b=? \rightarrow c = 10$

$a=0, b=0 \rightarrow c = 20$

$a=0, b \neq 0 \rightarrow c = 30$

```
if (a==0){
    if (b == 0)
        c = 20;
}else
    c = 30;
```

$a \neq 0, b=? \rightarrow c = 30$

$a=0, b=0 \rightarrow c = 20$

$a=0, b \neq 0 \rightarrow c = 10$

Viết chương trình nhập vào một số thực x . Tính và in ra màn hình giá trị của biểu thức làm tròn đến 3 chữ số sau dấu chấm thập phân

$$f(x) = \begin{cases} \frac{x^3 + x + 1}{x^2 - 1}, & x \neq \pm 1 \\ 3, & x = \pm 1 \end{cases}$$

Cho biểu giá bán lẻ điện sinh hoạt. Hãy viết chương trình nhập vào số kWh điện năng tiêu thụ của một gia đình trong 1 tháng và đưa ra thông báo số tiền điện phải trả?

| Bậc | Mức sử dụng | Giá điện (VNĐ) |
|-----|-----------------|----------------|
| 1 | 0-50 kWh | 1806 |
| 2 | 51-100 kWh | 1866 |
| 3 | 101-200 kWh | 2167 |
| 4 | 201-300 kWh | 2729 |
| 5 | 301-400 kWh | 3050 |
| 6 | 401 kWh trở lên | 3151 |

```
Nhap so dien tieu thu (kWh): 305
So tien dien phai tra la: 688450 VND.
-----
Process exited after 2.488 seconds with return value 0
Press any key to continue . . .
```

```
Nhap so dien tieu thu (kWh): -15
So lieu khong hop le!
-----
Process exited after 3.328 seconds with return value 0
Press any key to continue . . .
```

Viết chương trình tính cước Taxi theo công thức:
1 km đầu tiên có cước là 10000đ,
30 km tiếp theo có giá là 8000đ/1km
Các km sau đó có giá là 6000đ/1km.

```
Ban hay cho biet so km da di duoc: 0.5
So tien can tra = 10000 VND
-----
Process exited after 2.975 seconds with return value 0
Press any key to continue . . .
```

```
Ban hay cho biet so km da di duoc: 21
So tien can tra = 170000 VND
-----
Process exited after 4.385 seconds with return value 0
Press any key to continue . . .
```

```
Ban hay cho biet so km da di duoc: 33
So tien can tra = 262000 VND
-----
Process exited after 3.661 seconds with return value 0
Press any key to continue . . .
```

Ví dụ 1: Tính cước taxi (1)

```
1  #include <stdio.h>
2  #include <math.h> // dung ham ceil()
3  int main(){
4      unsigned long sotien;
5      float sokm;
6      printf("Ban hay cho biet so km da di duoc: "); scanf("%f", &sokm);
7      if (sokm <= 1.0)
8          sotien = 10000;
9      else
10         if (sokm <= 31.0)
11             sotien = 10000 + (ceil(sokm) - 1.0 ) * 8000;
12         else
13             sotien = 250000 + (ceil(sokm) - 31) * 6000;
14     printf("So tien can tra = %lu VND", sotien);
15     return 0;
16 }
```

Nhập số
âm vẫn
thực hiện

Kết quả:

```
Ban hay cho biet so km da di duoc: -2
So tien can tra = 10000 VND
-----
Process exited after 4.059 seconds with return value 0
Press any key to continue . . .
```

Ví dụ 1: Tính cước taxi (2)

```
1  #include <stdio.h>
2  #include <math.h> // dung ham ceil()
3  int main(){
4      unsigned long sotien;
5      float sokm;
6      printf("Ban hay cho biet so km da di duoc: "); scanf("%f", &sokm);
7      if(sokm>0){
8          if (sokm <= 1.0)
9              sotien = 10000;
10         else
11             if (sokm <= 31.0)
12                 sotien = 10000 + (ceil(sokm) - 1.0 )*8000;
13             else
14                 sotien = 250000 + (ceil(sokm) - 31)*6000;
15         printf("So tien can tra = %lu VND", sotien);
16     }
17     else
18         printf("So km phai duong!");
19     return 0;
20 }
```

Kết quả:

```
Ban hay cho biet so km da di duoc: -2.0
So km phai duong!
-----
Process exited after 3.401 seconds with return value 0
Press any key to continue . . .
```

Ví dụ 1: Tính cước taxi (3)

```
1 #include <stdio.h>
2 #include <math.h> // dung ham ceil()
3 int main(){
4     unsigned long sotien;
5     float sokm;
6     printf("Ban hay cho biet so km da di duoc: "); scanf("%f", &sokm);
7     if(sokm>0){
8         sotien= sokm <=1.0 ? 10000 : sokm <= 31? 10000 + (ceil(sokm) - 1.0 ) * 8000 : 250000 + (ceil(sokm) - 31) * 6000;
9         printf("So tien can tra = %lu VND", sotien);
10    }
11    else
12        printf("So km phai duong!");
13    return 0;
14 }
```

Kết quả:

```
Ban hay cho biet so km da di duoc: -2.0
So km phai duong!
-----
Process exited after 3.401 seconds with return value 0
Press any key to continue . . .
```

Ví dụ 2:

Giải phương trình bậc 2: $ax^2 + bx + c = 0$.
Minh họa giao diện nhập xuất như sau:

```
Nhap he so a, b, c: 0 1 2
PT suy bien thanh phuong trinh bac 1: 1.000000x + 2.000000 = 0
-----
```

```
Nhap he so a, b, c: 2 3 1
Phuong trinh co 2 nghiem phan biet:
x1 = -0.500000    x2 = -1.000000
-----
```

```
Nhap he so a, b, c: 1 2 3
Phuong trinh vo nghiem.
-----
```

```
Nhap he so a, b, c: 1 2 1
Phuong trinh co nghiem kep x1 = x2 = -1.000000
-----
```

Ví dụ 2: Giải phương trình bậc 2 $ax^2 + bx + c = 0$

```
1  #include <stdio.h>
2  #include <math.h> //thu vien de dung ham sqrt
3  int main(){
4      float a, b, c, delta;
5      printf("Nhap he so a, b, c: "); scanf("%f %f %f", &a, &b, &c);
6      delta = b*b - 4*a*c;
7      if(a==0)
8          printf("PT suy bien thanh phuong trinh bac 1 %fx + %f = 0",b,c);
9      else if(delta < 0)
10         printf("Phuong trinh vo nghiem.");
11     else if(delta == 0)
12         printf("Phuong trinh co nghiem kep x1 = x2 = %f",-b/(2*a));
13     else{
14         printf("Phuong trinh co 2 nghiem phan biet:\n");
15         printf("x1 = %f \t x2 = %f",(-b+sqrt(delta))/(2*a), (-b-sqrt(delta))/(2*a));
16     }
17     return 0;
18 }
```

```

1  #include <stdio.h>
2  #include <math.h> //thu vien de dung ham sqrt
3  int main(){
4      float a, b, c, delta;
5      printf("Nhap he so a, b, c: "); scanf("%f %f %f", &a, &b, &c);
6      delta = b*b - 4*a*c;
7      if(a==0)
8          printf("PT suy bien thanh phuong trinh bac 1: %gx + %g = 0",b,c);
9      else if(delta < 0)
10         printf("Phuong trinh vo nghiem.");
11     else if(delta == 0)
12         printf("Phuong trinh co nghiem kep x1 = x2 = %g",-b/(2*a));
13     else{
14         printf("Phuong trinh co 2 nghiem phan biet:\n");
15         printf("x1 = %g \t x2 = %g",(-b+sqrt(delta))/(2*a), (-b-sqrt(delta))/(2*a));
16     }
17     return 0;
18 }
```

```

Nhap he so a, b, c: 0 1.5 2.5
PT suy bien thanh phuong trinh bac 1: 1.5x + 2.5 = 0
-----
```

```

Nhap he so a, b, c: 1 3 2
Phuong trinh co 2 nghiem phan biet:
x1 = -1          x2 = -2
-----
```

```

Nhap he so a, b, c: 1 0 -6.25
Phuong trinh co 2 nghiem phan biet:
x1 = 2.5         x2 = -2.5
-----
```

```

Nhap he so a, b, c: 1.0 2.0 1.0
Phuong trinh co nghiem kep x1 = x2 = -1
-----
```

```

Nhap he so a, b, c: 1.5 2.5 4
Phuong trinh vo nghiem.
-----
```


Nhập số lượng và đơn giá (là các số nguyên dương) của một mặt hàng mà khách hàng chọn. Đưa ra tổng tiền (số thực) khách hàng phải trả, biết rằng:

- Nếu số lượng lớn hơn hoặc bằng 10 thì được giảm 30% tổng tiền.
- Nếu số lượng lớn hơn 4 và nhỏ hơn 10 thì được giảm 15% tổng tiền.
- Nếu số lượng nhỏ hơn hoặc bằng 4 thì không được giảm.

Nếu dữ liệu không hợp lệ (số lượng và đơn giá là số âm) thì đưa ra thông báo "So lieu khong hop le."

```
So luong: 23
Don gia: 15
So tien phai tra: 241.5
-----
```

```
1  #include<stdio.h>
2  #include<math.h>
3  int main()
4      int sl, dg;
5      float tien;
6      printf("So luong: "); scanf("%d", &sl);
7      printf("Don gia: "); scanf("%d", &dg);
8      if(sl <= 0 || dg <= 0)
9          printf("So lieu khong hop le.");
10     else {
11         if(sl >= 10)
12             tien = dg*sl*0.7;
13         else if(sl > 4)
14             tien = dg*sl*0.85;
15         else
16             tien = dg*sl;
17         printf("So tien phai tra: %g", tien);
18     }
19     return 0;
20 }
```

```
So luong: 23
Don gia: 15
So tien phai tra: 241.5
-----
```

```
So luong: 6
Don gia: 235
So tien phai tra: 1198.5
-----
```

Nhập Điểm trung bình (theo thang 4) của 1 sinh viên. Đưa ra thông báo xếp loại học lực của sinh viên đó. Biết rằng: ĐTB hợp lệ là trong đoạn $[0,4]$

Nếu $0 \leq \text{ĐTB} < 2$: Không đạt

Nếu $2 \leq \text{ĐTB} < 2.5$: Trung bình

Nếu $2.5 \leq \text{ĐTB} < 3.2$: Khá

Nếu $3.2 \leq \text{ĐTB} < 3.6$: Giỏi

Nếu $\text{ĐTB} \geq 3.6$: Xuất sắc

```
Nhap diem trung binh: 3.4
Gioi!
```

```
Nhap diem trung binh: 5
Diem nhap khong hop le!
```

Viết chương trình giải hệ phương trình bậc nhất:

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

```
1  #include <stdio.h>
2  int main()
3  {
4      float a1, b1, c1, a2, b2, c2, x, y, dx, dy, d;
5      printf("Nhap cac he so cua he PT:\n");
6      printf("a1, b1, c1 = "); scanf("%f%f%f", &a1, &b1, &c1);
7      printf("a2, b2, c2 = "); scanf("%f%f%f", &a2, &b2, &c2);
8      d  = a1 * b2 - a2 * b1;
9      dx = c1 * b2 - c2 * b1;
10     dy = a1 * c2 - a2 * c1;
11     if (d != 0)
12     {
13         x = dx/d;
14         y = dy/d;
15         printf("He PT co nghiem x = %f, y = %f.", x, y);
16     }
17     else
18         if (dx==0)
19             printf("He PT vo so nghiem!");
20         else
21             printf("He PT vo nghiem!");
22     return 0;
23 }
```

```
1 #include <stdio.h>
2 int main()
3 {
4     float a1, b1, c1, a2, b2, c2, x, y, dx, dy, d;
5     printf("Nhap cac he so cua he PT:\n");
6     printf("a1, b1, c1 = "); scanf("%f%f%f", &a1, &b1, &c1);
7     printf("a2, b2, c2 = "); scanf("%f%f%f", &a2, &b2, &c2);
8     d = a1 * b2 - a2 * b1;
9     dx = c1 * b2 - c2 * b1;
10    dy = a1 * c2 - a2 * c1;
11    if (d != 0)
12    {
13        x = dx/d;
14        y = dy/d;
15        printf("He PT co nghiem x = %f, y = %f.", x, y);
16    }
17    else
18        if (dx==0)
19            printf("He PT vo so nghiem!");
20        else
21            printf("He PT vo nghiem!");
22    return 0;
23 }
```

Nhap cac he so cua he PT:

a1, b1, c1 = 1 2 3

a2, b2, c2 = 2 4 6

He PT vo so nghiem!

Nhap cac he so cua he PT:

a1, b1, c1 = 1 3 5

a2, b2, c2 = 2 6 7

He PT vo nghiem!

Nhap cac he so cua he PT:

a1, b1, c1 = 1 2 3

a2, b2, c2 = 2 5 4

He PT co nghiem x = 7.000000, y = -2.000000.



Ví dụ 3 → Thực hiện chương trình



Nhap cac so:

a1, b1, c1 = 3 5 8

a2, b2, c2 = 2 1 9

He PT co nghiem $x = 5.285714$, $y = -1.571429$

Nhap cac so:

a1, b1, c1 = 1 2 3

a2, b2, c2 = 1 2 4

He phuong trinh vo nghiem!

Nhap cac so:

a1, b1, c1 = 1 2 3

a2, b2, c2 = 2 4 6

He PT co vo so nghiem!

3.1. Lệnh khối

3.2. Lệnh rẽ nhánh

➤ Cấu trúc if, if ... else

3.3. Lệnh lựa chọn

3.4. Các lệnh lặp

➤ Vòng lặp for

➤ Vòng lặp while và do while

3.5. Các lệnh nhảy

➤ Câu lệnh continue

➤ Câu lệnh break


```
switch (bieu_thuc)
```

```
{
```

```
    case gia_tri_1: lenh_1; [break;]
```

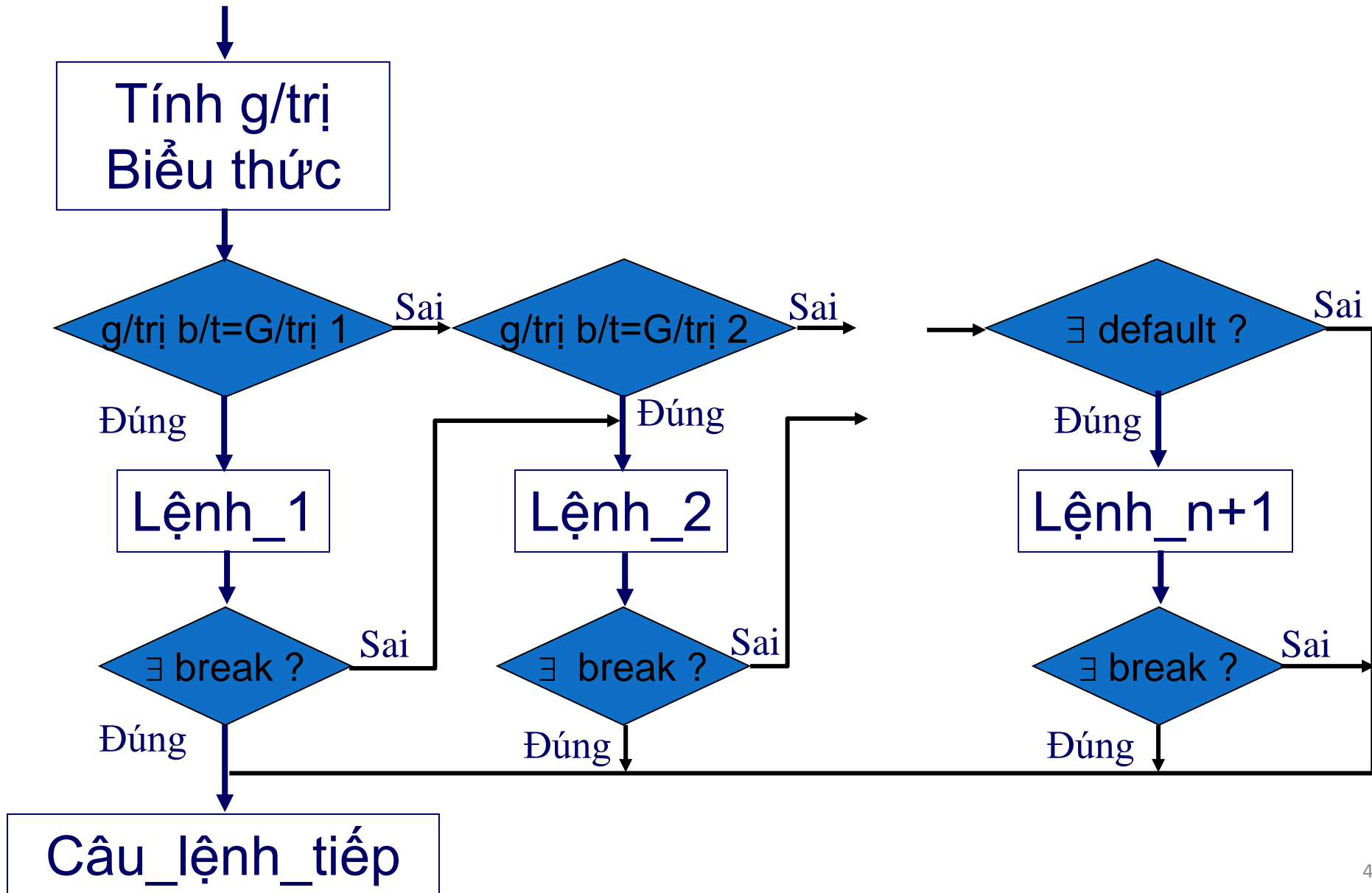
```
    case gia_tri_2: lenh_2; [break;]
```

```
    ...
```

```
    [default: lenh_n+1; [break;]]
```

```
}
```

```
Câu_lệnh_tiếp
```



Cơ chế hoạt động

- Tính giá trị của **biểu_thức**,
- So sánh giá trị của **biểu_thức** với các **giá_trị_k** (với $k = 1, 2, \dots, n$) nằm sau các từ khóa **case**
 - Xảy ra 2 khả năng:

❖ Tồn tại **giá_trị_i** bằng giá trị biểu thức.

□ Thực hiện **lệnh_i**

➤ Nếu tồn tại lệnh **break**,

- Nhảy tới tiếp tục thực hiện **Câu_lệnh_tiếp** nằm sau cấu trúc **switch**

➤ Nếu không tồn tại lệnh **break**

- Thực hiện các lệnh sau **lệnh_i** cho tới khi gặp **break** hoặc tới khi thoát khỏi cấu trúc **switch**
- Thực hiện **Câu_lệnh_tiếp**

- ❑ Không tồn tại **giá_trị_i** ($i = 1, 2, \dots, n$) nào bằng giá trị biểu thức
 - Nếu có nhãn **default**:
 - Chương trình sẽ thực hiện **lệnh_n+1**
 - Thực hiện **Câu_lệnh_tiếp** nằm ngay sau cấu trúc **switch**.
 - Nếu không có nhãn **default**:
 - Chương trình chuyển sang thực hiện lệnh tiếp theo nằm ngay sau cấu trúc **switch**: **Câu_Lệnh_tiếp**

Lập trình đọc từ bàn phím một số nguyên $1 \leq N \leq 10$ và đưa ra từ tiếng Anh tương ứng

```
Nhap mot gia tri so nguyen khong am: 4
4 -> Four
-----
Process exited after 1.652 seconds with return value 0
Press any key to continue . . .
```

```
Nhap mot gia tri so nguyen khong am: 11
Khong thoa man dieu kien [1..10]
-----
Process exited after 1.165 seconds with return value 0
Press any key to continue . . .
```

Cấu trúc lựa chọn switch: Ví dụ 1

```
1  #include <stdio.h>
2
3  int main(){
4      int N;
5      printf("Nhap mot gia tri so nguyen khong am: "); scanf("%d",&N);
6      switch(N){
7          case 1: printf("%d -> One",N); break;
8          case 2: printf("%d -> Two",N); break;
9          case 3: printf("%d -> Three",N); break;
10         case 4: printf("%d -> Four",N); break;
11         case 5: printf("%d -> Five",N); break;
12         case 6: printf("%d -> Six",N); break;
13         case 7: printf("%d -> Seven",N); break;
14         case 8: printf("%d -> Eight",N); break;
15         case 9: printf("%d -> Nine",N); break;
16         case 10: printf("%d -> Ten",N); break;
17         default:
18             printf("Khong thoa man dieu kien [1..10]");
19     }
20     return 0;
21 }
```



Cấu trúc lựa chọn switch → Ví dụ 2



Nhập vào số nguyên không âm, đưa ra ngày trong tuần tương ứng (theo số dư khi chia số đó cho 7, tức là nếu : 0 - Chủ nhật, 1 - Thứ Hai, 2 - Thứ 3,...

```
Nhap mot gia tri so nguyen khong am: 123  
Thu Nam
```

```
Nhap mot gia tri so nguyen khong am: 5  
Thu Sau
```



```
1  #include <stdio.h>
2  int main(){
3      int n;
4      printf("Nhap mot so nguyen khong am n = "); scanf("%d",&n);
5      if(n<0)
6          printf("So n am!");
7      else{
8          switch(n%7)
9          {
10             case 0: printf("Chu nhat"); break;
11             case 1: printf("Thu Hai"); break;
12             case 2: printf("Thu Ba"); break;
13             case 3: printf("Thu Tu"); break;
14             case 4: printf("Thu Nam"); break;
15             case 5: printf("Thu Sau"); break;
16             case 6: printf("Thu Bay");
17          }
18      }
19      return 0;
20 }
```

```
1  #include <stdio.h>
2  int main(){
3      int n;
4      printf("Nhap mot so nguyen khong am n = "); scanf("%d",&n);
5      if(n<0)
6          printf("So n am!");
7      else{
8          switch(n%7)
9          {
10             case 0: printf("Chu nhat"); break;
11             case 1: printf("Thu Hai"); break;
12             case 2: printf("Thu Ba"); break;
13             case 3: printf("Thu Tu"); break;
14             case 4: printf("Thu Nam"); break;
15             case 5: printf("Thu Sau"); break;
16             case 6: printf("Thu Bay");
17          }
18      }
19      return 0;
20 }
```

Nhap mot so nguyen khong am n = 13
Thu Bay

Nhap mot so nguyen khong am n = 345
Thu Ba

Có thể sử dụng đặc điểm: Không có lệnh break chương trình sẽ tự động chuyển xuống thực hiện các câu lệnh tiếp sau để viết chung mã lệnh cho các trường hợp khác nhau nhưng được xử lý như nhau.

Ví dụ: Trong một năm các tháng có 30 ngày là 4, 6, 9, 11 còn các tháng có 31 ngày là 1, 3, 5, 7, 8, 10, 12. Riêng tháng hai có thể có 28 hoặc 29 ngày. Hãy viết chương trình nhập vào 1 tháng, sau đó đưa ra kết luận tháng đó có bao nhiêu ngày

```
Nhap vao thang trong nam: 2
Thang 2 co 28 hoac 29 ngay!
-----
```

```
Nhap vao thang trong nam: 5
Thang 5 co 31 ngay!
-----
```

```
Nhap vao thang trong nam: 13
Khong co thang 13!
-----
```

```
Nhap vao thang trong nam: 9
Thang 9 co 30 ngay!
-----
```

Cấu trúc lựa chọn switch → Ví dụ

```
1  #include <stdio.h>
2  int main (){
3      int thang;
4      printf("Nhap vao thang trong nam: "); scanf("%d",&thang);
5      switch(thang){
6          case 1:
7          case 3:
8          case 5:
9          case 7:
10         case 8:
11         case 10:
12         case 12: printf("Thang %d co 31 ngay!",thang); break;
13         case 4:
14         case 6:
15         case 9:
16         case 11: printf("Thang %d co 30 ngay!",thang); break;
17         case 2: printf ("Thang %d co 28 hoac 29 ngay!",thang); break;
18         default :
19             printf("Khong co thang %d!",thang);
20     }
21     return 0;
22 }
```

```
#include <stdio.h>
int main (){
    int thang;
    printf("Nhap vao thang trong nam: "); scanf("%d",&thang);
    switch(thang){
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12: printf("Thang %d co 31 ngay!",thang); break;
        case 4:
        case 6:
        case 9:
        case 11: printf("Thang %d co 30 ngay!",thang); break;
        case 2: printf ("Thang %d co 28 hoac 29 ngay!",thang); break;
        default :
            printf("Khong co thang %d!",thang);
    }
    return 0;
}
```

```
Nhap vao thang trong nam: 2
Thang 2 co 28 hoac 29 ngay!
-----
```

```
Nhap vao thang trong nam: 13
Khong co thang 13!
-----
```

```
Nhap vao thang trong nam: 5
Thang 5 co 31 ngay!
-----
```

```
Nhap vao thang trong nam: 9
Thang 9 co 30 ngay!
-----
```

- Giá trị của biểu thức trong cấu trúc switch phải là số nguyên (*kiểu đếm được*)
 - Phải có kiểu dữ liệu là char, int, long
- Các giá trị sau từ khóa case (*gia_tri_1, gia_tri_2,..*) cũng phải là số nguyên

Điều kiện trong cấu trúc **if / if..else** cho phép làm việc với các kiểu dữ liệu khác số nguyên

1. Viết chương trình nhập vào một ký tự hệ hexa và đưa ra giá trị hệ 10 tương ứng
2. Lập trình đọc tọa độ 4 điểm A,B,C,M rồi kiểm tra xem điểm M nằm trong, nằm trên cạnh hay nằm ngoài tam giác ABC.
3. Lập trình đọc vào từ bàn phím 2 giá trị a, b rồi tính $y = 15x^2 + x + 7.2$ trong đó

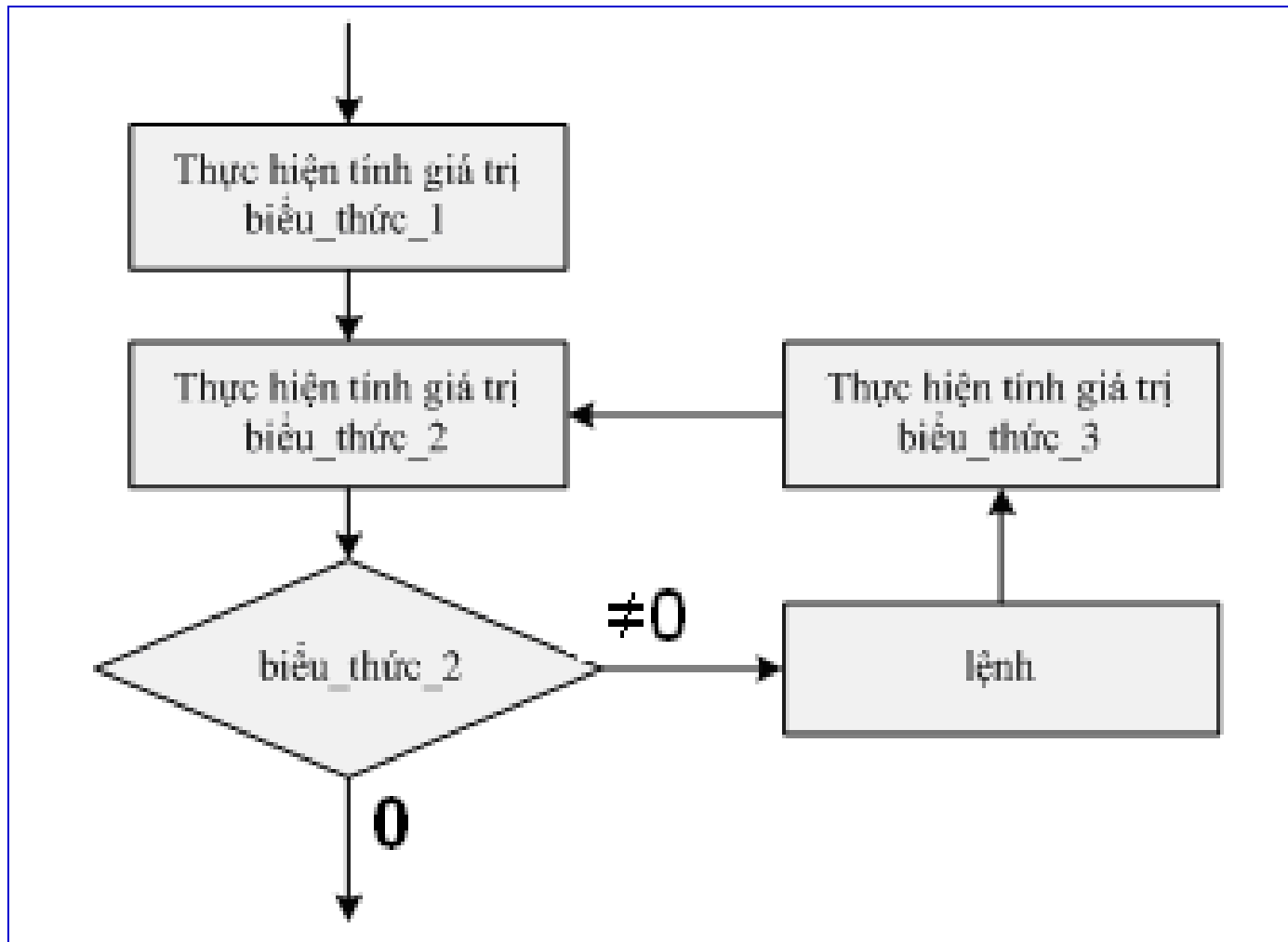
$$x = \begin{cases} \frac{a+b}{3} & \text{nê'u } a < b \\ 1.5172 & \text{nê'u } a = b \\ \frac{a-b}{a^2 + b^2} & \text{nê'u } a > b \end{cases}$$

Dùng để thực hiện nhiều lần một công việc

```
for([b.thuc_1];[b.thuc_2];[b.thuc_3])  
    Câu_Lệnh;
```

- **b.thuc_1**: Khởi tạo giá trị ban đầu cho vòng lặp
- **b.thuc_2**: Điều kiện tiếp tục vòng lặp
- **b.thuc_3**: Thay đổi biến điều khiển của vòng lặp
- **Câu_Lệnh** : Có thể là lệnh đơn lệnh kép hoặc lệnh rỗng

```
for([b.thuc_1];[b.thuc_2];[b.thuc_3]) Lệnh;
```



```
int i;  
for(i = 0; i < 100; i++) Câu_lệnh;
```

```
int i;  
for(i = 0; i < 100; i+=2) Câu_lệnh;
```

```
int i;  
for(i = 100; i > 0; i--) Câu_lệnh;
```

Tìm các số nguyên lẻ nhỏ hơn 100

```
#include <stdio.h>
int main(){
    int i;
    for(i = 1;i<100;i++)    {
        if(i%2 == 1) printf("%5d",i);
        if((i+1)%20 ==0) printf("\n");
    }
    return 0;
}
```

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 |
| 21 | 23 | 25 | 27 | 29 | 31 | 33 | 35 | 37 | 39 |
| 41 | 43 | 45 | 47 | 49 | 51 | 53 | 55 | 57 | 59 |
| 61 | 63 | 65 | 67 | 69 | 71 | 73 | 75 | 77 | 79 |
| 81 | 83 | 85 | 87 | 89 | 91 | 93 | 95 | 97 | 99 |

Tìm các số nguyên lẻ nhỏ hơn 100

```
#include <stdio.h>
int main(){
    int i;
    for(i = 99; i > 0; i-=2) {
        printf("%5d", i);
        if( (i-1) % 20 == 0) printf("\n");
    }
    return 0;
}
```

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 99 | 97 | 95 | 93 | 91 | 89 | 87 | 85 | 83 | 81 |
| 79 | 77 | 75 | 73 | 71 | 69 | 67 | 65 | 63 | 61 |
| 59 | 57 | 55 | 53 | 51 | 49 | 47 | 45 | 43 | 41 |
| 39 | 37 | 35 | 33 | 31 | 29 | 27 | 25 | 23 | 21 |
| 19 | 17 | 15 | 13 | 11 | 9 | 7 | 5 | 3 | 1 |

Nhập n và tính n!

```
1  #include<stdio.h>
2  #include<math.h>
3  int main()
4  {
5      int n;
6      long gt = 1;
7      printf("Cho n = "); scanf("%d",&n);
8      if(n<0)
9          printf("n la so am. Khong co giai thua!");
10     else
11     {
12         int i;
13         for(i=1;i<=n;i++)
14             gt =gt*i;
15         printf("%d! = %ld", n, gt);
16     }
17     return 0;
18 }
```

```
Cho n = 10
10! = 3628800
-----
Process exited after 3.489 seconds
Press any key to continue . . .
```

```
Cho n = -12
n la so am. Khong co giai thua!
-----
Process exited after 2.896 seconds
Press any key to continue . . .
```

Nhập n và tính tổng $S = 1 + 1/2 + \dots + 1/n$



Ví dụ



Nhập số tự nhiên n và tính tổng

$$S = 1 + 1/2 + \dots + 1/n$$

Nhập n và tính tổng $S = 1 + 1/2 + \dots + 1/n$

```
1  #include<stdio.h>
2
3  int main(){
4      int n;
5      float S = 0;
6      printf("Cho n = "); scanf("%d",&n);
7      if(n <= 0)
8          printf("n không thỏa mãn!");
9      else{
10         int i;
11         for(i=1;i<=n;i++)
12             S = S + (float)1/i; // S += 1.0/i;
13         printf("S = %g", S);
14     }
15     return 0;
16 }
```

Tìm số 3 chữ số thỏa mãn $abc = a^3 + b^3 + c^3$

Tìm số 3 chữ số thỏa mãn
 $abc = a^3 + b^3 + c^3$

Tìm số 3 chữ số thỏa mãn $abc = a^3 + b^3 + c^3$

```
#include <stdio.h>
int main()
{
    int i, a, b, c;
    for(i = 100; i < 1000; i++){
        a = i / 100;
        b = i % 100 / 10;
        c = i % 100 % 10;
        if(a*a*a + b*b*b + c*c*c == i)
            printf("%d \n", i);
    } //for
    return 0;
}
```

153
370
371
407

Tìm số 3 chữ số thỏa mãn $abc = a^3 + b^3 + c^3$

```
#include <stdio.h>
int main()
{
    int a, b, c;
    for(a = 1; a <= 9; a++)
        for(b = 0; b <= 9; b++)
            for(c = 0; c <= 9; c++)
                if(a*a*a + b*b*b + c*c*c == 100*a + 10*b + c)
                    printf("%d \n", 100*a + 10*b + c);
    return 0;
}
```

| |
|-----|
| 153 |
| 370 |
| 371 |
| 407 |

Không nhất thiết phải có đầy đủ các thành phần trong vòng lặp for

int getchar(): đọc ký tự từ vùng đệm bàn phím. Nếu vùng đệm rỗng, đợi người dùng gõ dãy ký tự (cho tới khi ấn phím Enter), sẽ trả về ký tự đầu

putchar(int c): đưa ký tự ra màn hình

1. Biểu thức khởi tạo

```
char c; int i=0;  
for( ; (c=getchar()) != '\n' ; i++)  
    putchar(c);  
printf("\nSo ky tu: %d",i);
```

```
Hello world  
Hello world  
So ky tu: 11
```

2. Biểu thức điều khiển

```
for(i=0 ; ; c=getchar(), i++)  
    if(c=='\n') break;  
printf("\nSo ky tu: %d",i);
```

```
Hello world  
  
So ky tu: 12
```

3. Thân vòng lặp

```
for(i=0 ; getchar() != '\n'; i++);  
printf("\nSo ky tu: %d",i);
```

```
Hello world  
So ky tu: 11
```

Nhập một số nguyên N , kiểm tra N có là số nguyên tố không?

```
Nhap gia tri N = 24
So 24 la hop so.
```

```
Nhap gia tri N = 13
So 13 la so nguyen to.
```

```
#include <stdio.h>
#include <math.h>
int main(){
    int N, i, OK = 1;
    printf("Nhap gia tri N = "); scanf("%d", &N);
    if (N<2) printf("So %d khong phai so nguyen to.", N);
    else {
        for (i=2; i < N; i++)
            if (N%i == 0) {
                OK = 0;
                break;
            }
        if (OK)
            printf("So %d la so nguyen to.", N);
        else
            printf("So %d la hop so.", N);
    }
    return 0;
}
```

```
Nhap gia tri N = 24
So 24 la hop so.
-----
```

```
Nhap gia tri N = 13
So 13 la so nguyen to.
-----
```

```
1  #include<stdio.h>
2
3  int main(){
4      int N, i, kt = 1;
5      printf("Cho so nguyen N = "); scanf("%d",&N);
6      if(N<2) kt = 0;
7      else{
8          for(i = 2; i<N; i++)
9              if(N%i == 0) kt = 0;
10     }
11     if(kt)
12         printf("%d la so nguyen to!",N);
13     else
14         printf("%d khong la so nguyen to!",N);
15     return 0;
16 }
```

```
Cho so nguyen N = 13
13 la so nguyen to!
-----
```

```
Cho so nguyen N = -12
-12 khong la so nguyen to!
-----
```

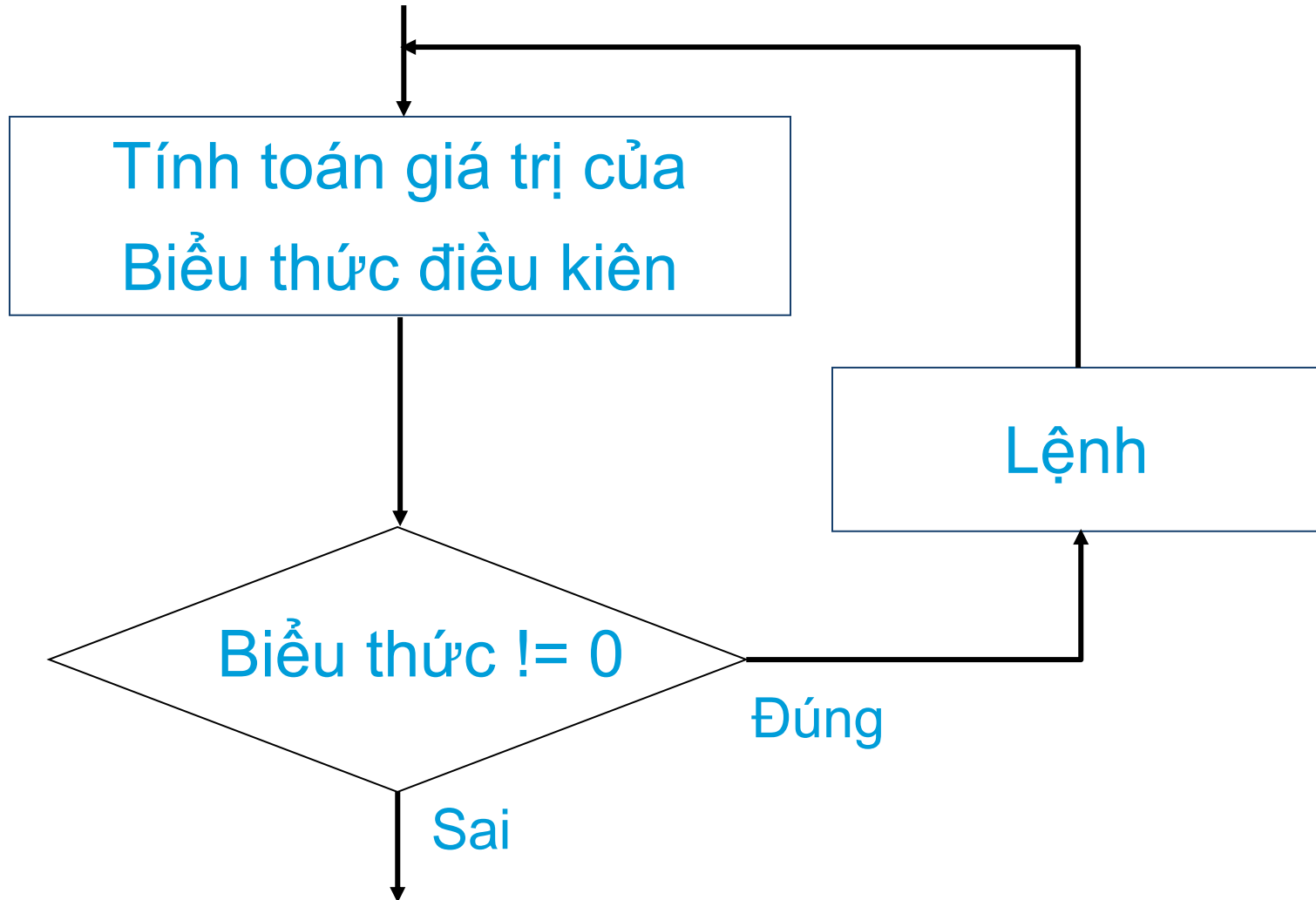

- ❖ Nhập số nguyên n ($n > 0$)
- ❖ Tính tổng các số chính phương $< n$
- ❖ Tính tổng các số nguyên tố $< n$

Dùng để thực hiện lặp đi lặp lại một công việc nào đó tới khi điều kiện không thỏa mãn

Cú pháp:

```
while (biểu_thức_điều_kiện)  
    lệnh;
```

- Chương trình kiểm tra điều kiện **trước** khi lặp
 - Giá trị của biểu thức điều kiện là đúng \Rightarrow thực hiện lệnh
- Các lệnh của vòng lặp có thể không được thực hiện lần nào nếu Biểu_thức_điều_kiện sai ngay từ đầu
- Nếu Biểu_thức_điều_kiện luôn đúng có thể gây lặp vô hạn



Nhập số tự nhiên n và in ra tổng của n số $1+2+..+n$

```
#include <stdio.h>
int main()
{
    long S = 0;
    int n;
    printf("Nhap n : ");scanf("%d",&n);
    while (n > 0)
    {
        S = S + n;
        n = n - 1;
    }
    printf("Ket qua là %ld \n",S);
    return 0;
}
```

```
Nhap n : 96
Ket qua là 4656
```



Ví dụ



Tìm số nguyên dương lớn nhất thỏa mãn
 $3n^5 - 317n < 5$

Tìm số nguyên lớn nhất thỏa mãn
 $3n^5 - 317n < 5$

```
#include <stdio.h>
#include <math.h>
int main()
{
    int n=0;
    while (3* pow(n,5) - 317*n < 5) n++;
    printf("%4d \n",n-1);
    return 0;
}
```

Kết quả
3

Cho biết kết quả thực hiện chương trình

```
#include <stdio.h>
int main()
{
    int i=3;
    while (i > 1){
        if(i % 2==0) i = i / 2;
        else i = i * 3 + 1;
        printf("%4d \n", i);
    }
    return 0;
}
```

Kết quả

10
5
16
8
4
2
1

Nhập một xâu và đếm số nguyên âm,
phụ âm, khoảng trắng

Nhập một chuỗi và đếm số nguyên âm, phụ âm, khoảng trắng

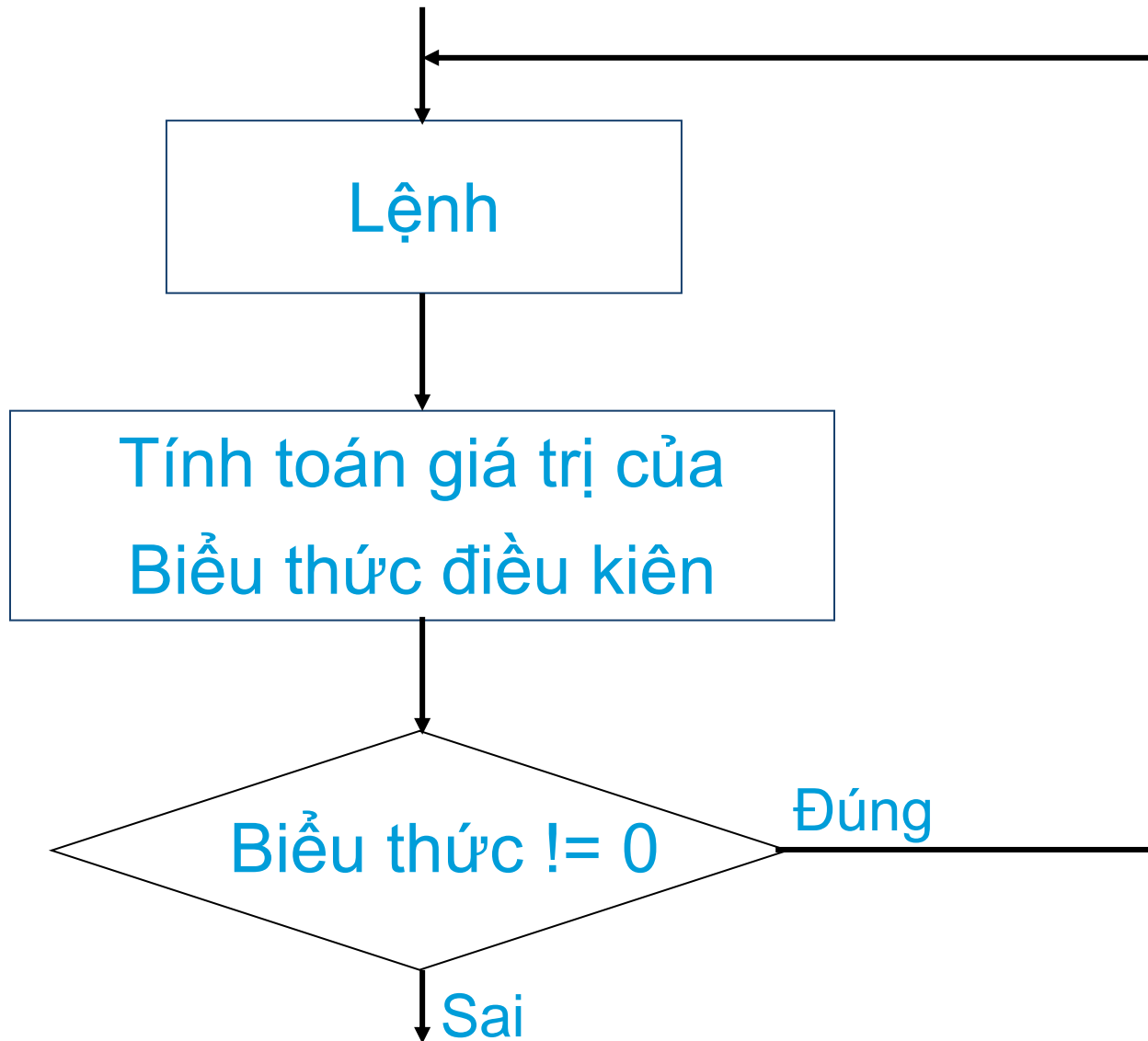
```
1  #include <stdio.h>
2  int main(){
3      int na, pa, kt;
4      char c;
5      na = pa = kt = 0;
6      printf(">");
7      while((c=getchar())!='\n')
8      {
9          switch(c){
10             case 'a':
11             case 'e':
12             case 'i':
13             case 'o':
14             case 'u':
15             case 'A':
16             case 'E':
17             case 'I':
18             case 'O':
19             case 'U': na++; break;
20             case ' ': kt++; break;
21             default : pa++;
22         }
23     }
24     printf("Chuoi co: %d nguyen am, %d phu am va %d khoang trang.",na,pa,kt);
25     return 0;
26 }
```

- Dùng để thực hiện lặp đi lặp lại một công việc nào đó

- Cú pháp:

```
do {  
    lenh;  
} while (bieu_thuc_dieu_kien) ;
```

- Chương trình kiểm tra điều kiện **sau** khi lặp
- Các `lenh` được thực hiện ít nhất một lần
- Biểu thức luôn đúng, lặp vô hạn lần



Nhập n và đưa tổng của n số nguyên đầu tiên

```
1  #include <stdio.h>
2  #include <conio.h>
3  void main(){
4      long S = 0;
5      int n;
6      printf("Nhap n = "); scanf("%d",&n);
7      do{
8          S = S + n;
9          n = n - 1;
10     }while (n>0);
11     printf("Ket qua la %ld",S);
12     getch();
13 }
```

do
S += n--;
while (n > 0);

- Nhập vào điểm của một sinh viên, nếu điểm đó không $\in [0, 10]$ thì thông báo cho người dùng nhập lại.
- Thực hiện:
 - Nếu dùng lệnh **if**
 - Chỉ kiểm tra được 1 lần
 - Sử dụng **for**
 - Chưa biết trước số lần lặp.
 - Sử dụng vòng lặp không cần xác định trước số lần lặp: **while** / **do while**

Dùng vòng lặp while

```
#include <stdio.h>
int main(){
    float diem;
    printf("Chương trình nhập điểm sinh viên\n");
    printf("Nhập điểm (0<=diem<=10):");
    scanf("%f",&diem);
    while (diem < 0 || diem > 10) {
        printf("\nBạn nhập không đúng!\n");
        printf("Bạn hãy nhập lại (0<=diem<=10):");
        scanf("%f",&diem);
    }
    printf("\nĐiểm bạn vừa nhập là: %.2f\n", diem);
    return 0;
}
```



Dùng vòng lặp while → Kết quả



```
Chương trình nhập điểm sinh viên  
Nhập điểm ( $0 \leq \text{diem} \leq 10$ ): 12
```

```
Bạn nhập không đúng!  
Bạn hãy nhập lại ( $0 \leq \text{diem} \leq 10$ ): -2
```

```
Bạn nhập không đúng!  
Bạn hãy nhập lại ( $0 \leq \text{diem} \leq 10$ ): 6.5
```

```
Điểm bạn vừa nhập là: 6.50
```

Dùng vòng lặp do...while

```
#include <stdio.h>
int main() {
    float diem;
    printf("Chương trình nhập diem sinh vien\n");
    do {
        printf("Nhap diem (0<=diem<=10):");
        scanf("%f",&diem);
        if (diem < 0 || diem > 10)
            printf("\nBan nhap khong dung!\n");
    } while (diem < 0 || diem > 10);
    printf("\nDiem ban vua nhap la: %.2f \n", diem);
    return 0;
}
```


Dùng vòng lặp do...while

```
#include <stdio.h>
int main(){
    float diem;
    printf("Chương trình nhập điểm sinh viên\n");
    printf("Nhập điểm (0<=diem<=10):");
    scanf("%f",&diem);
    while (diem < 0 || diem > 10) {
        printf("\nBạn nhập không đúng!\n");
        printf("Bạn hãy nhập lại (0<=diem<=10):");
        scanf("%f",&diem);
    }
    printf("\nĐiểm bạn vừa nhập là: %.2f\n", diem);
    return 0;
}
```

```
#include <stdio.h>
int main() {
    float diem;
    printf("Chương trình nhập điểm sinh viên\n");
    do {
        printf("Nhập điểm (0<=diem<=10):");
        scanf("%f",&diem);
        if (diem < 0 || diem > 10)
            printf("\nBạn nhập không đúng!\n");
    } while (diem < 0 || diem > 10);
    printf("\nĐiểm bạn vừa nhập là: %.2f \n", diem);
    return 0;
}
```

Nhập số nguyên dương n và phân tích số nguyên ra thừa số nguyên tố

Viết chương trình thực hiện công việc

- Nhập vào từ bàn phím 2 số nguyên
- Nhập vào từ bàn phím một ký tự bất kỳ;
 - Nếu đây là một toán tử số học thì đưa ra giá trị tương ứng với toán tử.
 - Nếu không phải thì đưa ra thông báo sai
- Chương trình thực hiện cho tới khi ký tự nhập vào là 'q' hoặc 'Q'

```
#include <stdio.h>
int main() {
    int a, b;
    char ch;
    int Fin = 0;
    printf("Nhap cac so a, b "); scanf("%d%d",&a,&b);
    do{
        printf("\nToan tu (+ ; - ; * ; / ; %) ");
        while ((getchar()) != '\n');
        ch = getchar();
        switch(ch){
            case '+': printf(" Co ket qua: %d\n",a+b); break;
            case '-': printf(" Co ket qua: %d\n",a-b); break;
            case '*': printf(" Co ket qua: %d\n",a*b); break;
            case '%': if (b==0) printf(" Chia cho 0\n");
                     else printf(" Co ket qua: %d\n",a%b);
                     break;
            case '/': if (b==0) printf(" Chia cho 0\n");
                     else printf(" Co ket qua: %d\n",a/b);
                     break;
            case 'q':
            case 'Q': Fin=1; break;
            default: printf(" khong co toan tu nay\n");
        }
    }while(Fin==0);
    printf("\nKet thuc, an mot phim... \n");
    return 0;
}
```

Nhap cac so a, b 145 60

Toan tu (+ ; - ; * ; / ; %) +
Co ket qua: 205

Toan tu (+ ; - ; * ; / ; %) -
Co ket qua: 85

Toan tu (+ ; - ; * ; / ; %) *
Co ket qua: 8700

Toan tu (+ ; - ; * ; / ; %) /
Co ket qua: 2

Toan tu (+ ; - ; * ; / ; %) %
Co ket qua: 25

Toan tu (+ ; - ; * ; / ; %) g
khong co toan tu nay

Toan tu (+ ; - ; * ; / ; %) q

Ket thuc, an mot phim...

Ví dụ: Nhập chuỗi, đếm số ký tự của chuỗi

❖ Khai báo:

```
int n = 0;
```

```
char c;
```

❖ Dùng vòng for

```
for(n = 0; ;c = getchar(), n++)
```

```
    if(c == '\n')
```

```
        break;
```

```
for(n=0; getchar() != '\n' ; n++) ;
```

Ví dụ: Nhập xâu, đếm số ký tự của xâu

❖ Dùng vòng lặp **while**

```
c = getchar();  
while (c != '\n'){  
    c = getchar();  
    n++;  
}
```

❖ Dùng vòng lặp **do... while**

```
do{  
    c = getchar();  
    n++;  
}while(c != '\n');
```

❖ Đưa kết quả ra: `printf("Chuoi chua %d ky tu", n);`

Nhập chuỗi ký tự cho đến khi gặp ký tự '*'.
Tính tần suất xuất hiện nguyên âm 'a'

Nhập chuỗi ký tự cho đến khi gặp ký tự '*'.

Tính tần suất xuất hiện nguyên âm 'a'

```
#include <stdio.h>
#include <ctype.h>
int main(){
    char c;
    int n, d;
    do{
        printf("Cho chuỗi: ");
        d = 0; n = 0;
        while( (c=getchar()) != '*'){
            n++;
            if (c=='a') d++;
        }
        if(n==0)
            printf("\nChuỗi ký tự rỗng\n");
        else
            printf("\nTần suất xuất hiện ký tự 'a' là: %5.2f%%\n", (float)100*d/n);
        printf("Tiếp tục <C/K>?: "); scanf("%c\n",&c);
    }while(toupper(getchar()) != 'K');
    return 0;
}
```


Ví dụ: Nhập một số nguyên N, kiểm tra là số nguyên tố không?

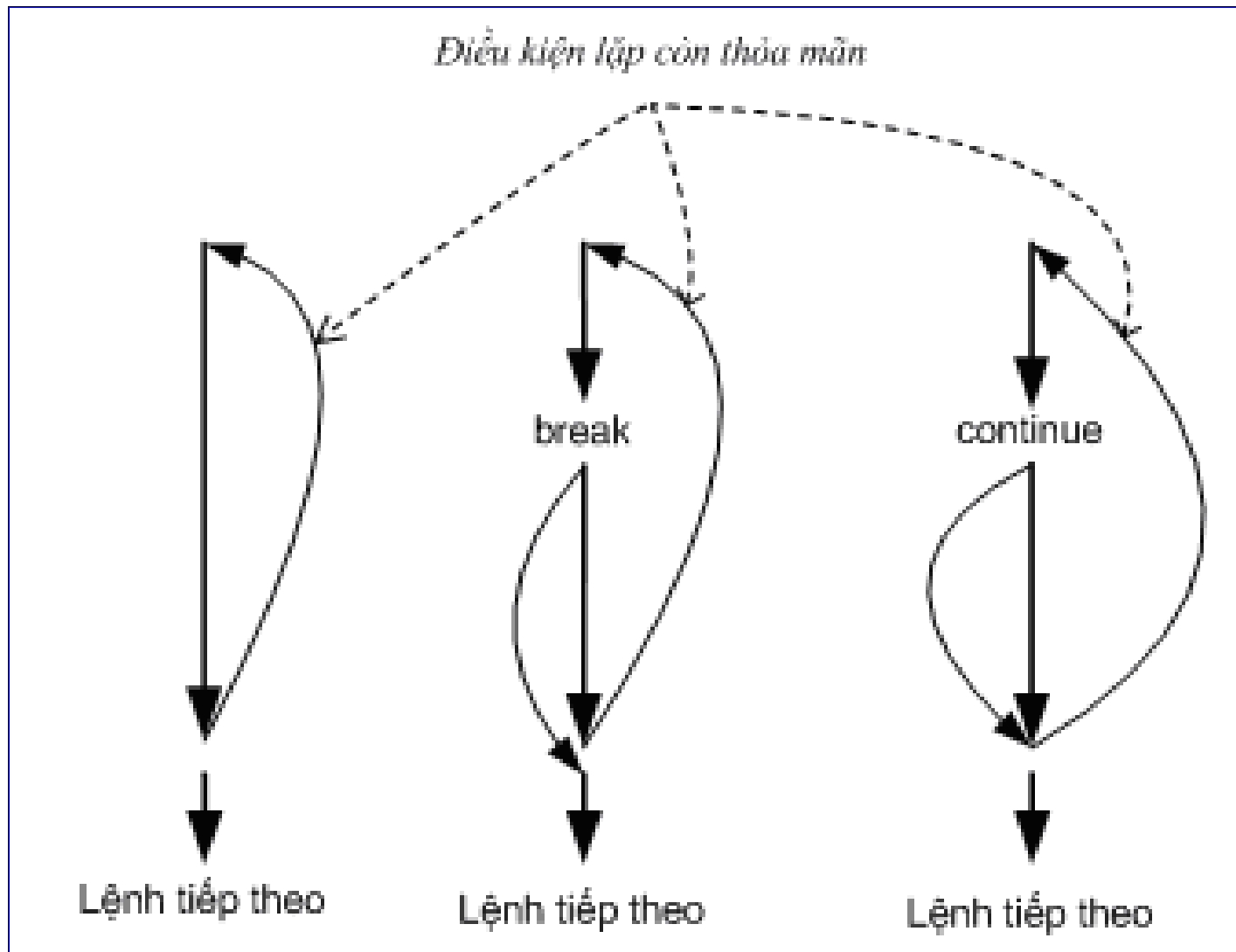
```
#include <stdio.h>
#include <math.h>
int main()
{ int N, i, OK = 1;
  printf("Nhap gia tri N = "); scanf("%d", &N);
  if (N<2) printf("So %d khong phai so nguyen to.", N);
  else {
    for (i=2; i < N; i++)
      if (N%i == 0) {
        OK = 0;
        break;
      }
    if (OK) printf("So %d la so nguyen to.", N);
    else printf("So %d la hop so.", N);
  }
  return 0;
}
```

$i \leq (\text{int})\text{sqrt}(N)$

```
i = 2;
while (N % i != 0) i++;
if (i == N) printf("so nguyen to")
```

- ❖ Các vòng lặp **while/ do ... while/ for** sẽ kết thúc quá trình lặp khi biểu thức điều kiện của vòng lặp không còn được thỏa mãn.
- ❖ Tuy nhiên trong lập trình đôi khi ta cũng cần thoát khỏi vòng lặp ngay cả khi biểu thức điều kiện của vòng lặp vẫn còn được thỏa mãn.
- ❖ Để hỗ trợ người lập trình làm việc đó, ngôn ngữ C cung cấp 2 câu lệnh là **continue** và **break**

continue > < break



- ❖ Bỏ qua việc thực hiện các câu lệnh nằm sau lệnh continue trong thân vòng lặp.
- ❖ Chuyển sang thực hiện một vòng lặp mới

In ra 100 số nguyên đầu tiên ngoại trừ các số chia hết cho 5

```
#include <stdio.h>
int main()
{
    int i;
    for(i = 1; i <= 100; i++)
    {
        if(i % 20 == 0)
            printf("\n");
        if(i % 5 == 0)
            continue;
        printf("%3d", i);
    }
    return 0;
}
```

```
1  2  3  4  6  7  8  9 11 12 13 14 16 17 18 19
21 22 23 24 26 27 28 29 31 32 33 34 36 37 38 39
41 42 43 44 46 47 48 49 51 52 53 54 56 57 58 59
61 62 63 64 66 67 68 69 71 72 73 74 76 77 78 79
81 82 83 84 86 87 88 89 91 92 93 94 96 97 98 99
```

Process exited after 0.09011 seconds with return value 0
Press any key to continue . . .

if (i % 5 != 0)
 printf("%3d", i);

In ra 100 số nguyên đầu tiên ngoại trừ các số chia hết cho 5

```
#include <stdio.h>
int main()
{
    int i;
    for(i = 1; i <= 100; i++)
    {
        if(i % 20 == 0)
            printf("\n");
        if (i % 5 != 0)
            printf("%3d", i);
    }
    return 0;
}
```

```
1  2  3  4  6  7  8  9 11 12 13 14 16 17 18 19
21 22 23 24 26 27 28 29 31 32 33 34 36 37 38 39
41 42 43 44 46 47 48 49 51 52 53 54 56 57 58 59
61 62 63 64 66 67 68 69 71 72 73 74 76 77 78 79
81 82 83 84 86 87 88 89 91 92 93 94 96 97 98 99
```

```
-----
Process exited after 0.09011 seconds with return value 0
Press any key to continue . . .
```

- ❖ Thoát khỏi vòng lặp ngay cả khi biểu thức điều kiện của vòng lặp vẫn còn được thỏa mãn.
- ❖ Chú ý:
 - break dùng để thoát ra khỏi khối lặp hiện tại
 - break cũng dùng để thoát ra khỏi lệnh rẽ nhánh switch

```
1  #include <stdio.h>
2  int main(){
3      int n;
4      do{
5          printf("\nEnter the number: ");
6          scanf("%d", &n);
7          if (n<0)
8              break;
9          if(n>10){
10             printf("Skip the value.");
11             continue;
12         }
13         printf("The number is: %d", n);
14     }while (n!=0);
15     return 0;
16 }
```

```
Enter the number: 1
The number is: 1
Enter the number: 12
Skip the value.
Enter the number: 5
The number is: 5
Enter the number: -1
-----
Process exited after 25.87 second
Press any key to continue . . .
```



```
#include <stdio.h>
int main()
{
    int i;
    for(i = 1; i<=10; i++)
    {
        if(i == 5) continue;
        printf("%5d \n", i);
        if(i==7) break;
    }
    return 0;
}
```

1
2
3
4
6
7

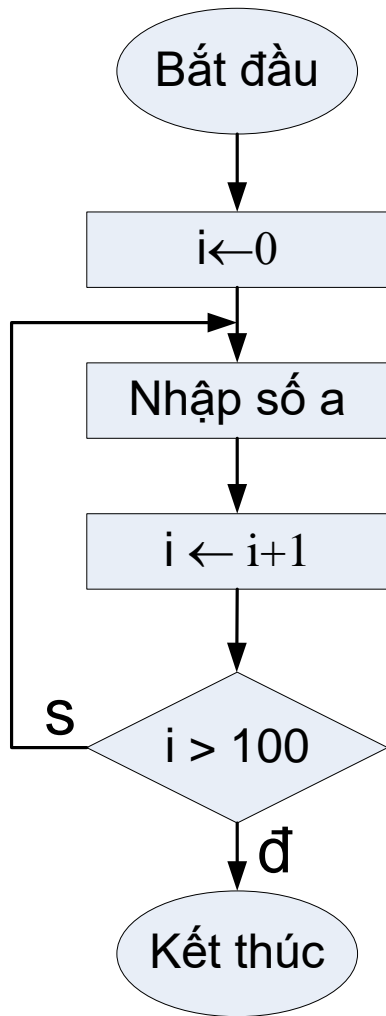
```
#include <stdio.h>
int main()
{
    int i, j;
    for(i = 0; i < 10; i++)
    {
        for(j = 0; j < 10; j++)
        {
            if(j > i)
                break;
            printf("i:%d j: %d\n", i, j);
        }
    }
    return 0;
}
```

| | |
|-----|------|
| i:0 | j:1 |
| i:1 | j:2 |
| i:2 | j:3 |
| i:3 | j:4 |
| i:4 | j:5 |
| i:5 | j:6 |
| i:6 | j:7 |
| i:7 | j:8 |
| i:8 | j:9 |
| i:9 | j:10 |

Viết chương trình thực hiện các công việc sau

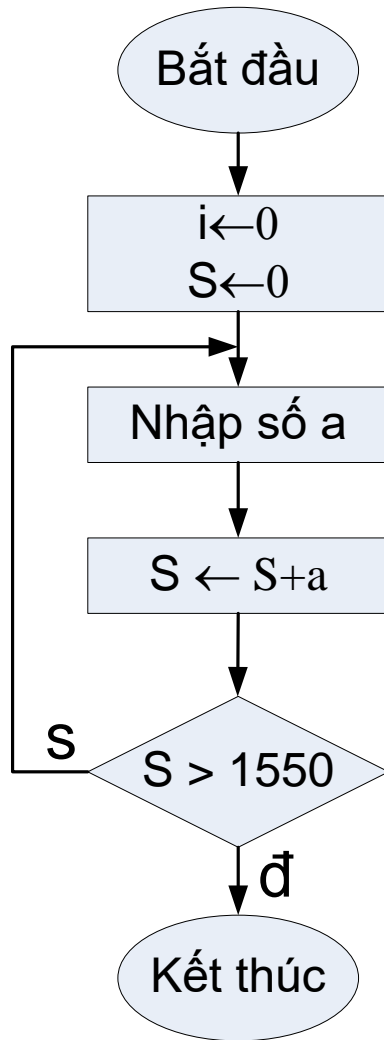
- Nhập vào một dãy số cho tới khi: Tổng của dãy lớn hơn 1550 hoặc là số phần tử trong dãy lớn hơn 100
- Đưa ra số lượng phần tử nằm trong khoảng (35, 70)
- Đưa ra trung bình cộng của các phần tử chia hết cho 7

Nhập một dãy số cho tới khi số phần tử trong dãy lớn hơn 100



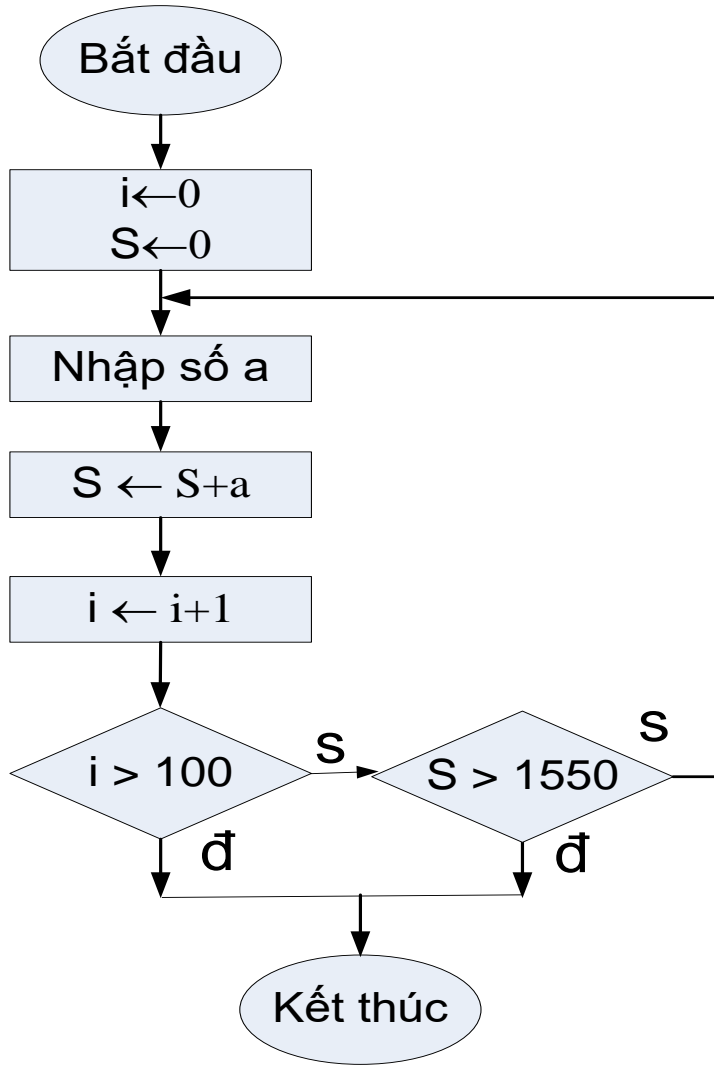
```
#include <stdio.h>
int main(){
    int a, i = 0;
    do{
        printf("Nhap vao so nguyen: ");
        scanf("%d",&a);
        i++;
    }while(i<=100);
    return 0;
}
```

Nhập một dãy số cho tới khi tổng của dãy lớn hơn 1550



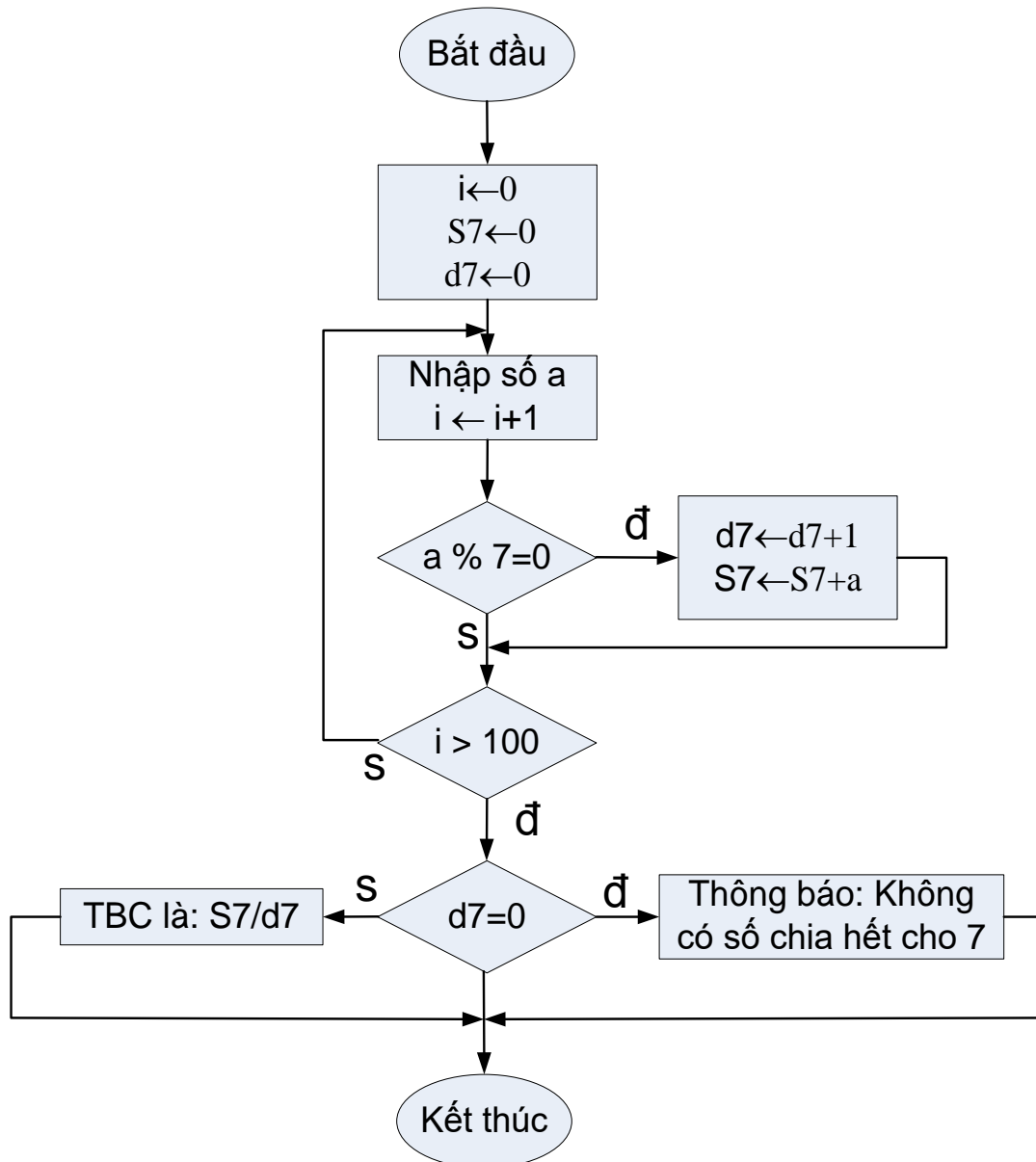
```
#include <stdio.h>
int main(){
    int a, S;
    S = 0;
    do{
        printf("Nhap vao so nguyen:");
        scanf("%d",&a);
        S+=a;
    }while (S <= 1550);
}
```

Nhập một dãy số cho tới khi thỏa mãn....



```
#include <stdio.h>
int main(){
    int a, i, S;
    S = 0; i = 0;
    do{
        printf("Nhap vao so nguyen:");
        scanf("%d",&a);
        S+=a;
        i++;
    }while((i<=100)&&(S <= 1550));
}
```

Đưa ra TBC của các phần tử chia hết cho 7(1)





Đưa ra TBC của các phần tử chia hết cho 7(1)



```
#include <stdio.h>
int main(){
    int a, i = 0, S7 = 0, d7 = 0;
    do{
        printf("Nhap vao so nguyen:"); scanf("%d",&a);
        i++;
        if(a%7==0){
            d7++;
            S7 += a;
        }
    }while (i <= 100);
    if(d7==0)
        printf("Khong co so chia het cho 7");
    else
        printf("Ket qua la %.4f",(float)S7/d7);
}
```


Đưa ra TBC của các phần tử chia hết cho 7(2)

```
#include <stdio.h>
int main(){
    int a, i = 0, S7 =0, d7 = 0, S = 0;
    do{
        printf("Nhap vao so nguyen:"); scanf("%d",&a);
        i++; S+=a;
        if(a%7==0){
            d7++;
            S7+=a;
        }
    }while ((i<=100)&&(S <= 1550));
    if(d7==0)
        printf("Khong co so chia het cho 7");
    else
        printf("Ket qua la %.4f",(float) S7/d7);
}
```



```
#include <stdio.h>
int main(){
    int a, i = 0, S7 = 0, d7 = 0, S = 0, d = 0;
    do{
        printf("Nhap vao so nguyen: "); scanf("%d",&a);
        i++;
        S+=a;
        if(a%7==0){
            d7++;
            S7 += a;
        }
        if((a>35) && (a<70) ) d++;
    }while ((i <=100)&&(S <= 1550));
    printf("So phan tu trong khoang (35,70) la %d \n",d);
    if(d7==0)
        printf("Khong co so chia het cho 7");
    else
        printf("TBC cac so chia het cho 7 %.4f",(float) S7/d7);
    return 0;
}
```

Cho hàm số $f(x) = x^5 + \sqrt[5]{x}$. Lập trình tính và đưa ra màn hình các cặp giá trị $\langle x, f(x) \rangle$ với x lấy dãy giá trị -10;-9.9;-9.8;; 4.9; 5.0.

```
#include <stdio.h>
#include <math.h>
int main(){
    float x, fx;
    for(x = -10.0; x <= 5.0; x += 0.1){
        if(x==0)
            fx = 0.0;
        else
            fx = pow(x,5)+x/fabs(x) * pow(fabs(x), 0.2);
        printf("<%4.1f,%7.2f>\n",x,fx);
    }
    return 0;
}
```



Bài tập



Đọc vào dãy số cho tới khi gặp một số dương chia hết cho 5. Tìm số lớn nhất của dãy và số lần xuất hiện các giá trị đó

Đọc vào dãy số cho tới khi gặp một số dương chia hết cho 5;
Tìm số lớn nhất của dãy và số lần xuất hiện các giá trị đó

```
#include <stdio.h>
#include <limits.h>

int main(){
    int a, d=0, max = INT_MIN;
    do {
        printf("Nhap mot so : "); scanf("%d",&a);
        if( a > max){
            max = a;
            d = 1;
        }else
            if (a == max) d++;
    }while ( a <=0 || a%5 != 0);
    printf("Max: %d va co %d gia tri.",max,d);
}
```

Đọc vào dãy số cho tới khi gặp một số dương chia hết cho 5;
Tìm số lớn nhất của dãy và số lần xuất hiện các giá trị đó

```
#include <stdio.h>
#include <limits.h>
int main(){
    int a, d = 0, max = INT_MIN;
    do {
        printf("Nhap mot so: "); scanf("%d",&a);
        if( a > max){
            max = a;
            d = 1;
        }
        else
            if (a == max) d++;
    }while (a<=0 || a%5!= 0);
    printf("Max: %d, va co %d gia tri.",max, d);
    return 0;
}
```

```
Nhap mot so: 7
Nhap mot so: 6
Nhap mot so: 4
Nhap mot so: 8
Nhap mot so: 9
Nhap mot so: 23
Nhap mot so: 46
Nhap mot so: 3
Nhap mot so: 46
Nhap mot so: 7
Nhap mot so: 46
Nhap mot so: 8
Nhap mot so: 9
Nhap mot so: 5
Max: 46, va co 3 gia tri.
-----
```


Viết chương trình đọc x và n vào từ bàn phím rồi tính

$$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Kết quả hiển thị:

```
Nhap vao so nguyen n: 20
Nhap vao so thuc x: 1
Ket qua la 2.71828198
-----
Process exited after 12.84 seconds
Press any key to continue . . .
```

Viết chương trình đọc x và số nguyên dương n vào từ bàn phím rồi tính

$$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

```
#include <stdio.h>
int main(){
    int n, i;
    float x, u = 1.0, S = 1.0;
    printf("Nhap vao so nguyen n: "); scanf("%d",&n);
    printf("Nhap vao so thuc x: "); scanf("%f",&x);
    for(i = 1; i <= n; i ++){
        u *= x/i;
        S += u;
    }
    printf("Ket qua la %.8f",S);
    return 0;
}
```

```
Nhap vao so nguyen n: 20
Nhap vao so thuc x: 1
Ket qua la 2.71828198
-----
Process exited after 12.84 seconds
Press any key to continue . . .
```

Nhập x và ε và tính với độ chính xác nhỏ hơn ε

$$S = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots + \frac{(-1)^n x^n}{n!} + \dots$$

Viết chương trình đọc x và n vào từ bàn phím rồi tính

$$S = \sqrt{x + \sqrt{x + \sqrt{x + \cdots + \sqrt{x}}}} \quad n \text{ dấu căn}$$

$$S = 1 + x + \frac{x^2}{2} + \frac{x^3}{3} + \cdots + \frac{x^n}{n}$$

$$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

$$S = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \cdots + \frac{(-1)^n x^n}{n!}$$

Đọc vào số thực x và epsilon (ϵ) từ bàn phím và tính biểu thức sau với độ chính xác nhỏ hơn epsilon ϵ :

$$S_1 = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots + \frac{(-1)^n x^n}{n!} + \dots$$

$$S_2 = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{(-1)^n}{(2n+1)!} x^{2n+1} + \dots // \sin(x)$$

$$S_3 = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{(-1)^n}{(2n)!} x^{2n} + \dots // \cos(x)$$

1. Câu lệnh khối

Đặt trong cặp ngoặc nhọn { }

2. Cấu trúc rẽ nhánh

- if (biểu_thức), if (biểu_thức) ... else

3. Cấu trúc lệnh lựa chọn

- switch (biểu_thức) {(case/break/default)}

4. Cấu trúc lặp

- for(biểu_thức_1; biểu_thức_2; biểu_thức_3) CâuLệnh;
- while (biểu_thức) CâuLệnh;
- do Câu_Lệnh while (biểu_thức);

5. Các lệnh thay đổi cấu trúc lập trình

- continue/ break

- 1) Cho số nguyên dương n ($0 < n \leq 100000$). Tính tổng các chữ số của n .
- 2) Cho số nguyên dương n ($0 < n \leq 10000$). Tìm chữ số lớn nhất và chữ số nhỏ nhất trong n .
- 3) Cho hai số nguyên dương a và b ($0 < a, b \leq 10000$). Tìm UCLN và BCNN của a và b .
- 4) Cho số nguyên dương n ($0 < n \leq 10000$). Kiểm tra xem n có phải là số nguyên tố hay không?
- 5) Biết rằng: số tự nhiên n được gọi là số hoàn hảo nếu tổng các ước số của n bằng đúng $2n$. Ví dụ: $n = 6$ có các ước là 1, 2, 3, 6 và có $1 + 2 + 3 + 6 = 12$. Cho hai số tự nhiên a và b ($1 \leq a < b \leq 10000$). Tìm tất cả các số hoàn hảo trong đoạn $[a, b]$

- 6) Nhập số nguyên n , in ra màn hình n số nguyên kể từ 100 trở đi.
- 7) Nhập số nguyên n , in ra màn hình các số chẵn nhỏ hơn n .
- 8) Nhập số n nguyên dương, tính và in các số chính phương từ 1 đến n^2 ra màn hình, mỗi số một dòng.
- 9) Nhập số n nguyên dương, in các số chính phương từ 1 đến n ra màn hình, mỗi số một dòng.

1. Nhập số nguyên n , in ra màn hình n số nguyên kể từ 100 trở đi.

2. Nhập số n nguyên dương, tính và in các số chính phương từ 1 đến n ra màn hình, mỗi số một dòng.

3. Nhập số n nguyên dương, tính giá trị số Y :

$$Y = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}.$$

4. Nhập số n và tính giá trị của số Z :

$$Z = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{1}{2n-1} - \frac{1}{2n}$$

5. Nhập số n nguyên dương, in các số nguyên tố trong khoảng từ 1 đến n ra màn hình.

6. Nhập số n nguyên dương ($n > 100$), in các số thừa số nguyên tố của số n .

1) Nhập số e dương đủ nhỏ, tính giá trị số S : $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$
dừng khi $\frac{1}{n} < e$.

2) Nhập số e và tính giá trị của số Y : $Y = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots$ dừng khi $\frac{1}{n} < e$.

3) Nhập số e dương đủ nhỏ, tính giá trị số S :

$$S = 1 - \frac{1}{2} + \frac{1}{3} - \dots + \frac{(-1)^{n-1}}{n}$$

dừng khi $\frac{1}{n} < e$.

4) Lập trình đọc x , e từ bàn phím rồi tính:

$$S = 1 + \frac{x}{2!} + \frac{x^2}{3!} + \dots + \frac{x^{n-1}}{n!}$$

dừng khi $\frac{1}{n!} < e$.

5) Lập trình đọc x , e từ bàn phím rồi tính:

$$S = 1 + \frac{x}{2} - \frac{x^2}{3} + \dots + \frac{(-1)^n x^{n-1}}{n}$$

dừng khi $\frac{1}{n} < e$.