



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

NHẬP MÔN LẬP TRÌNH

INTRODUCTION TO COMPUTER PROGRAMMING

CSE102

Giảng viên: TS.Nguyễn Quỳnh Diệp

Email: diepnq@tlu.edu.vn



Chương 3

CÁC LỆNH LỰA CHỌN VÀ LẶP



NỘI DUNG CHÍNH



1. Các câu lệnh lựa chọn

- Lệnh if
- Lệnh switch

2. Các câu lệnh lặp

- Lệnh for
- Lệnh while
- Lệnh do while

3. Lệnh nhảy

- break
- continue

4. Kết hợp lặp và lựa chọn



1. Các câu lệnh lựa chọn

▪ Lệnh đơn:

- Là lệnh bên trong nó không chứa lệnh nào
- Kết thúc 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 { }

- 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 ?

▪ Kết quả thực hiện chương trình

Bien ngoai khoi c = 10; d = 20

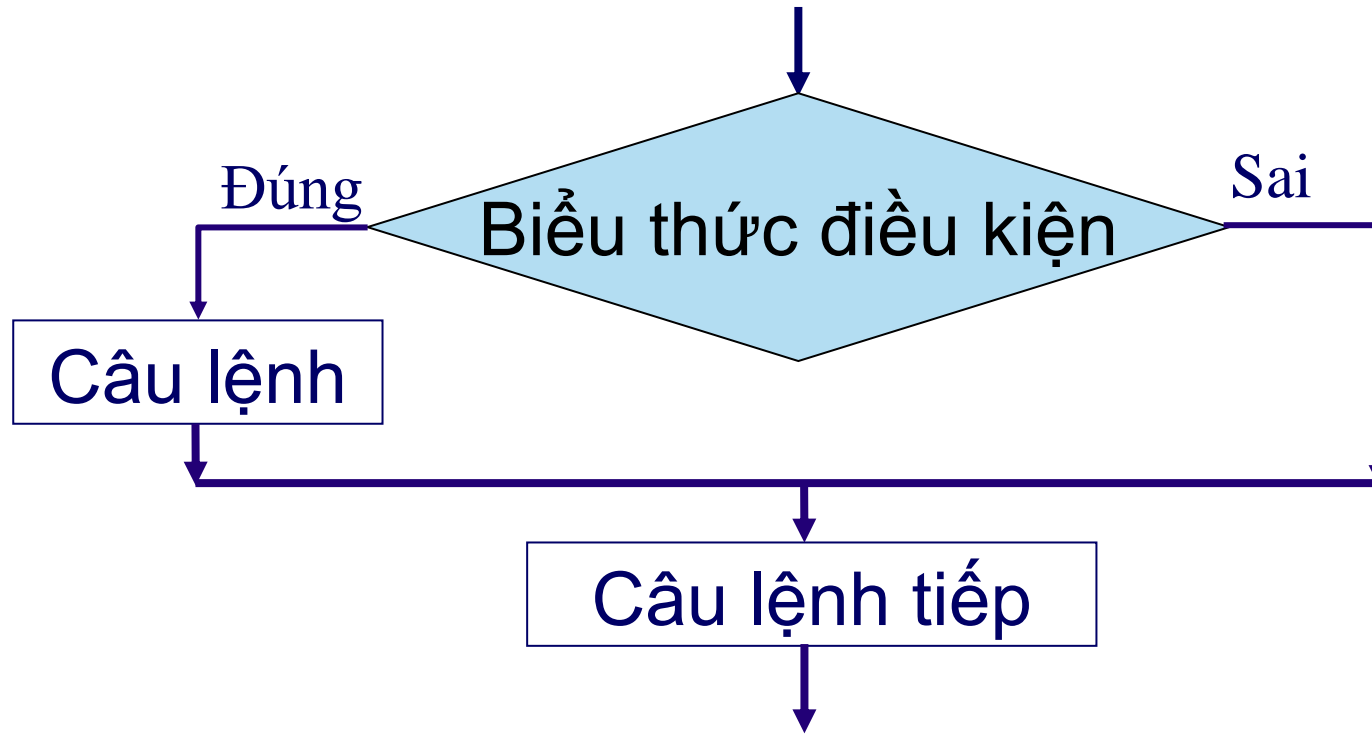
Bien trong khoi c = 10; d = 20

Gia tri cua cac bien duoc them 10 don vi

Bien trong khoi c = 20; d = 30

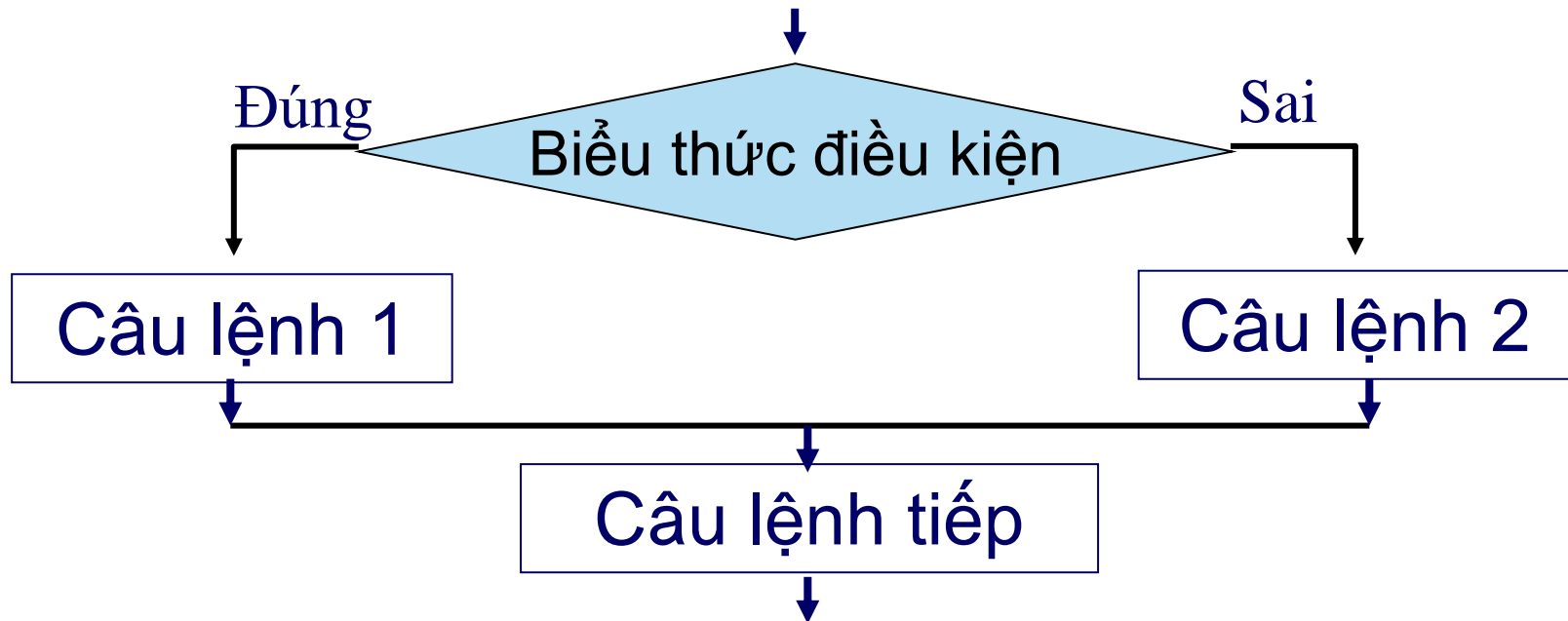
Bien ra ngoai khoi c = 10; d = 30

Lưu ý: Biến địa phương / Biến toàn cục



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

- `x=5; y=3; if (x>y) printf("x>y")`

- `if (2+5) printf("Hello world! ");` -> Chấp nhận

■ Câu lệnh:

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

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 %0.4f la: %0.4f", a, b, max);
13     return 0;
14 }
```

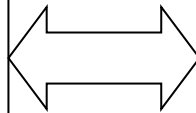
```
Nhap a = 4.5
Nhap b = 7.0
So lon nhat trong 2 so 4.5000 va 7.0000 la: 7.0000
-----
Process exited after 10.8 seconds with return value 0
Press any key to continue . . .
```

Ví dụ: So sánh 2 số thực được nhập vào

Nhap gia tri a va b: 12.345 54.321

So lon nhat trong 2 so 12.3450 va 54.3210 la 54.3210

```
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  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 }
```

```
Giai phuong trinh bac nhat ax + b = 0
Nhap he so a: 0
Nhap he so b: 3.5
Phuong trinh vo nghiem!
-----
```

```
Giai phuong trinh bac nhat ax + b = 0
Nhap he so a: 0
Nhap he so b: 0
Phuong trinh vo so nghiem!
-----
```

```
Giai phuong trinh bac nhat ax + b = 0
Nhap he so a b: 3.5 7.0
Nghiem x = -2.0000
-----
```

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 }
```

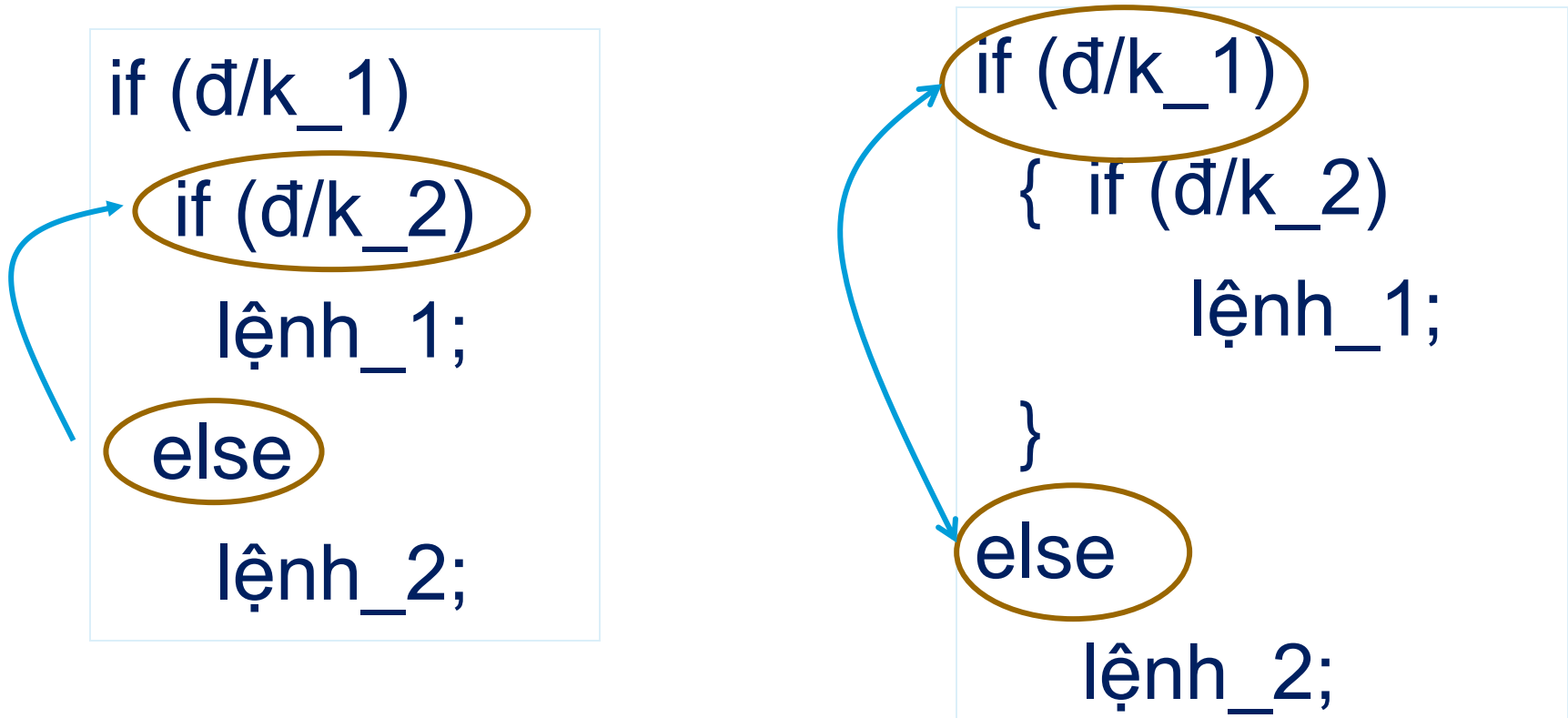
```
Giai phuong trinh bac nhat ax + b = 0
Nhap he so a: 3.5
Nhap he so b: 7.0
Nghiem x = -2
```

```
-----
Process exited after 5.467 seconds with return value 0
Press any key to continue . . .
```

- Nhập số thực x và tính hàm

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

- 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



Lập trình nhập từ bàn phím một số nguyên $1 \leq N \leq 5$ và đưa ra từ tiếng Anh tương ứng

```
1  #include<stdio.h>
2
3  void main(){
4      int x;
5      printf (" Nhap 1 so x bat ki tu 1 den 5 : ");
6      scanf("%d",&x);
7      if(x==1)
8          printf("\nEnglish : one ");
9      if(x==2)
10         printf("\nEnglish : two ");
11     if(x==3)
12         printf("\nEnglish : three ");
13     if(x==4)
14         printf("\nEnglish : four ");
15     if(x==5)
16         printf("\nEnglish : five ");
17
18 }
```



```
switch (bieu_thuc)
```

```
{
```

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

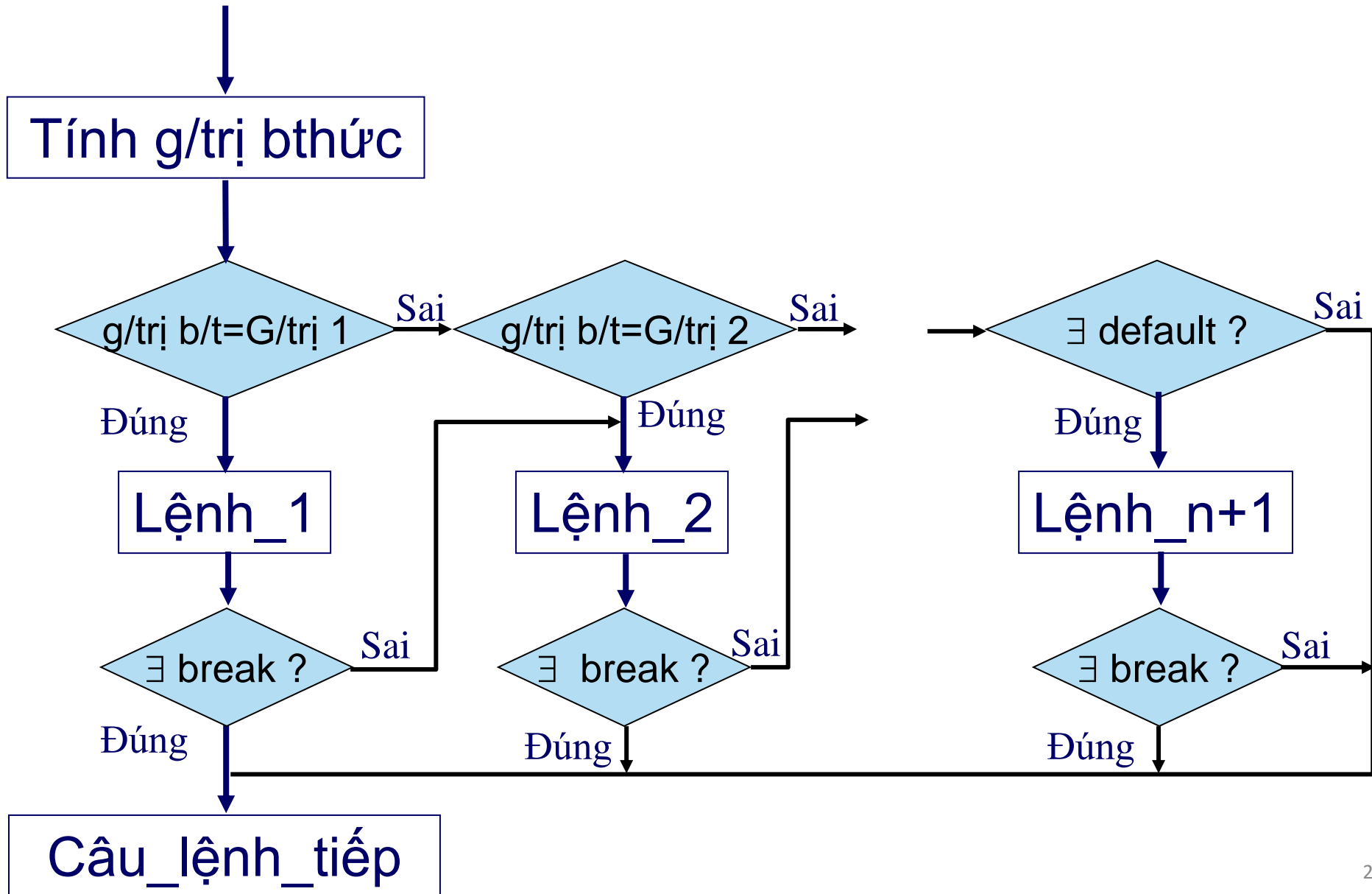
```
    case gia_tri_2: lenh_2; [break];
```

```
    ...
```

```
    [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***,
 - ✓ Kết thúc switch
 - ✓ 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**:
 - Kết thúc cấu trúc switch
 - Chuyển sang thực hiện ***Câu_Lệnh_tiếp*** nằm ngay sau cấu trúc switch

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

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

▪ Kết quả thực hiện chương trình

```
Nhap mot gia tri so nguyen khong am: 7  
7 -> Seven
```

```
Nhap mot gia tri so nguyen khong am: 3  
3 -> Three
```

```
Nhap mot gia tri so nguyen khong am: -6  
Khong thoa man dieu kien [1..10]
```

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

Cấu trúc lựa chọn switch

- Có thể lợi 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 2 có thể có 28 hoặc 29 ngày.

Hãy viết chương trình nhập vào 1 tháng, sau đó đưa ra thông báo số ngày của tháng đó.

```
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 }
```


■ Kết quả thực hiện

```
#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, char

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

4. Viết chương trình tính cước Taxi theo công thức:
 - 1 km đầu tiên có cước là 10.000đ,
 - 30 km tiếp theo có giá là 8.000đ/1km
 - Các km sau đó có giá là 6.000đ/1km.
 - (Chú ý: tính cước theo số chẵn km, không tính km lẻ)
5. Viết chương trình giải phương trình bậc hai $ax^2+bx+c=0$
6. 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 #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

```
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 . . .
```

```
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 }
```

```
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 . . .
```

- 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 }
```

- 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. Lập trình nhập vào từ bàn phím 2 giá trị a, b nguyên rồi tính $y = 15x^2 + x + 7.2$, trong đó

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

2. Viết chương trình nhập vào một ký tự hệ hexa và đưa ra giá trị hệ 10 tương ứng

3. Lập trình nhập tọa độ 4 điểm A, B, C, M rồi kiểm tra xem điểm M , nằm trên cạnh, nằm trong tam giác hay nằm ngoài tam giác ABC .



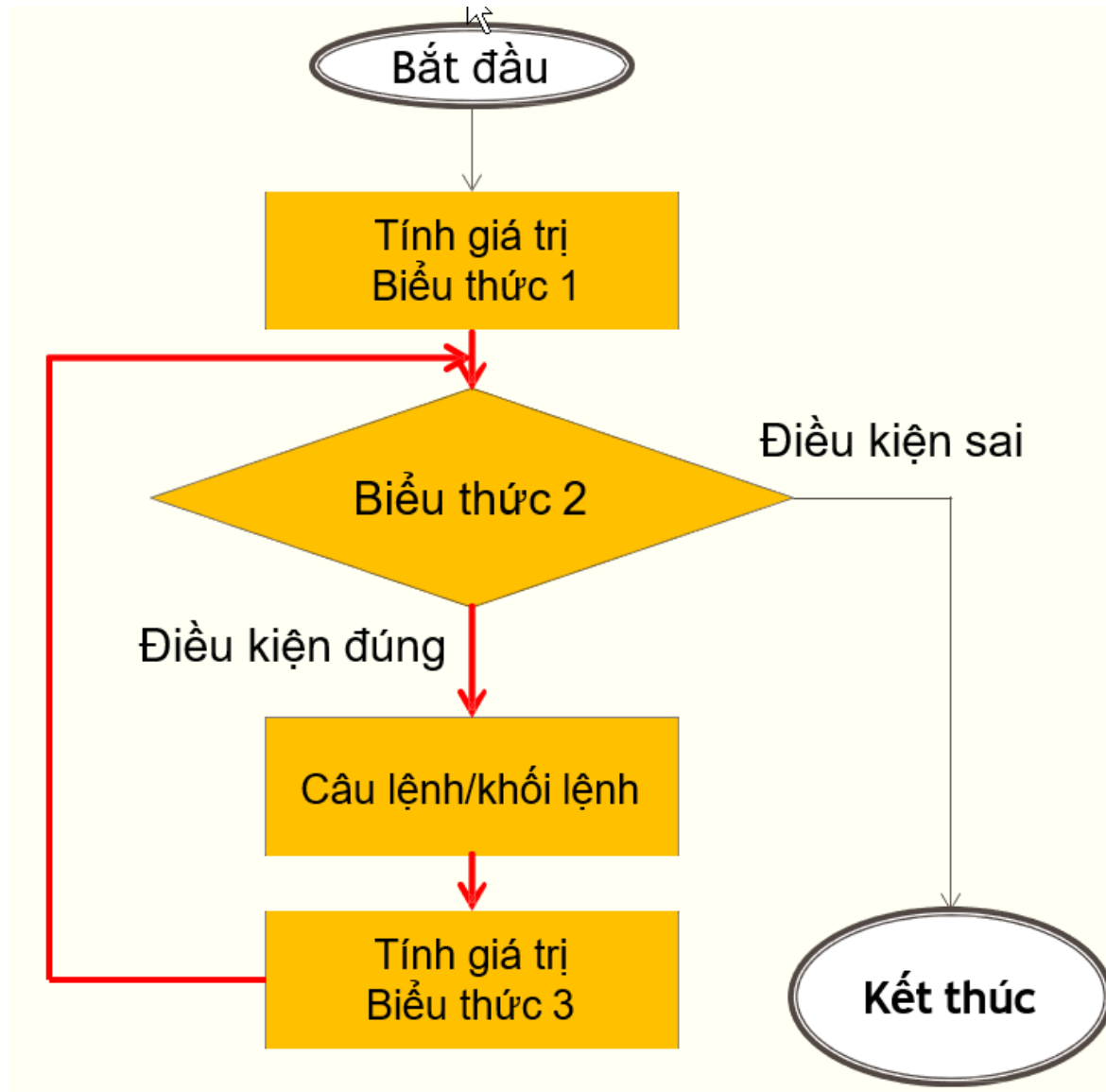
2. Các câu lệnh lặp

Cấu trúc lặp for

- Tác dụng: Dùng để thực hiện nhiều lần một công việc
- Cú pháp:

```
for([b.thuc_1];[b.thuc_2];[b.thuc_3])  
    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: biểu thức thay đổi biến điều khiển của vòng lặp
- Lệnh: Có thể là lệnh đơn lệnh kép



```
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;
```

Ví dụ 1

- Tìm các số nguyên lẻ, nhỏ hơn 100

Cách 1

```
#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

Ví dụ 1

- Tìm các số nguyên lẻ, nhỏ hơn 100

Cách 2

```
#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 số nguyên n ($n > 0$)

a. Tính tổng các số nguyên liên tiếp từ 1 đến n

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

b. Tính tổng các số chính phương $< n$

c. Tính tổng các số nguyên tố $< n$

VD3: Nhập n và tính tổng $S = 1 + 1/2 + \dots + 1/n$

```
#include <stdio.h>
int main()
{
    float S = 0.0;
    int n, i;
    printf("Nhap n : ");
    scanf("%d",&n);
    for(i = 1; i <= n; i++)
        S = S + (float)1/i; //S+=1.0/i;
    printf("Ket qua là %7.4f \n",S);
    return 0;
}
```

- VD4: Tìm số 3 chữ số thỏa mãn $abc = a^3 + b^3 + c^3$

Cách 1

```
#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);
    }
    return 0;
}
```

153
370
371
407

Cách 2

```
#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



- Trong vòng for không nhất thiết phải có đầy đủ các thành phần
- **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(c)**: đưa ký tự ra màn hình

1. Thiếu 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. Thiếu 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. Thiếu thân vòng lặp

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

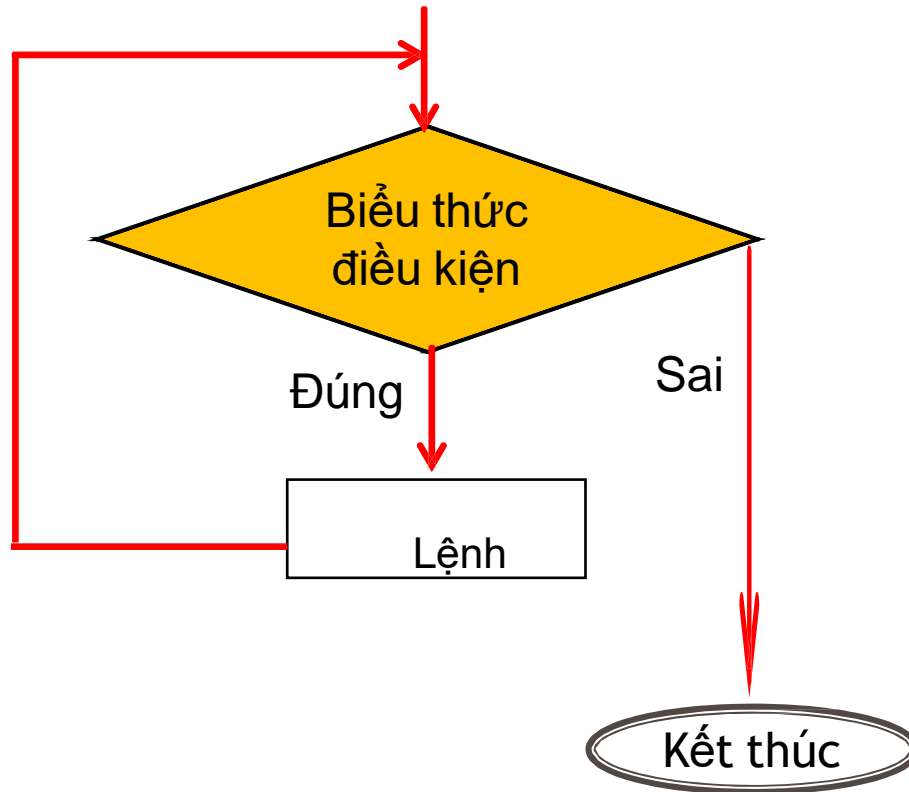
```
Hello world  
So ky tu: 11
```

Cấu trúc lặp while

- Tác dụng: Dùng để thực hiện lặp đi lặp lại một công việc nào đó tới khi điều kiện không thỏa mãn
- Cú pháp:

```
while(bieu_thuc_dieu_kien)  
    lenh;
```

- Thực hiện:
 - Chương trình kiểm tra (bieu_thuc_dieu_kien) **trước** khi lặp
 - Giá trị của (bieu_thuc_dieu_kien) là đúng \Rightarrow thực hiện lệnh
 - Các lệnh của vòng lặp có thể không được thực hiện lần nào nếu (bieu_thuc_dieu_kien) sai ngay từ đầu
 - Nếu (bieu_thuc_dieu_kien) luôn đúng có thể dẫn đến lặp vô hạn



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

Cách 1:

```
#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

Cách 2:

$$S=1+2+3+4+5$$

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

```
int main()
```

```
{ long S=0;
```

```
  int n;
```

```
  printf("Nhap n = "); scanf("%d", &n);
```

```
  int i=1;
```

```
  while (i<=n)
```

```
  {   S=S+i;
```

```
      i++;
```

```
  }
```

```
  printf("Tong S = %ld", S);
```

```
  return 0;
```

```
}
```

- Tìm số nguyên dương 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;
}
```

- 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          switch(c){
9              case 'a':
10             case 'e':
11             case 'i':
12             case 'o':
13             case 'u':
14             case 'A':
15             case 'E':
16             case 'I':
17             case 'O':
18             case 'U': na++; break;
19             case ' ': kt++; break;
20             default : pa++;
21         }
22     printf("Chuoi co: %d nguyen am, %d phu am va %d khoang trang.",na,pa,kt);
23     return 0;
24 }
```

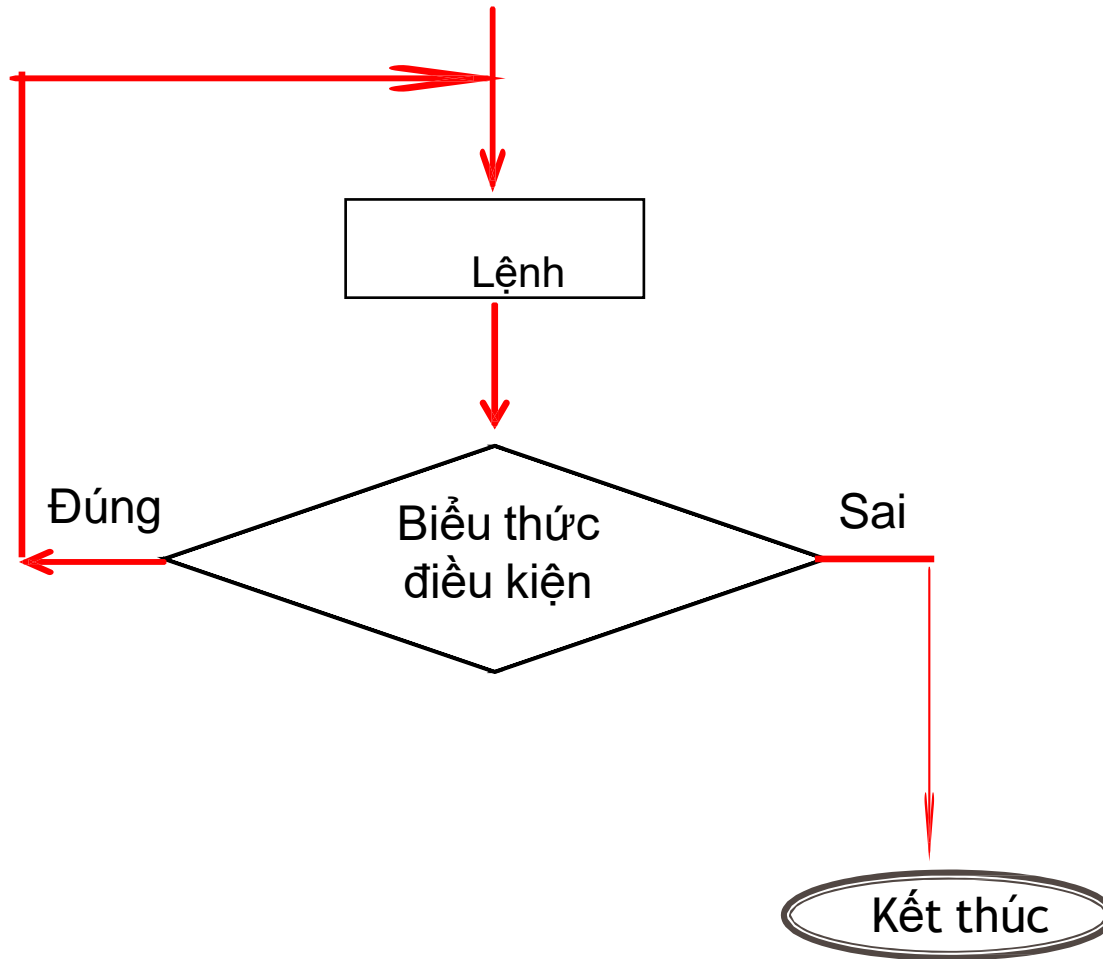
```
>Dai hoc Bach Khoa Ha Noi
Chuoi co: 9 nguyen am, 10 phu am va 5 khoang trang.
-----
Process exited after 13.08 seconds with return value 0
Press any key to continue . . .
```

Cấu trúc lặp do.. while..

- Tác dụng: Dùng để thực hiện lặp đi lặp lại một công việc nào đó
- Cú pháp:

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

- Thực hiện:
 - Chương trình thực hiện **lenh**
 - Sau đó kiểm tra (bieu_thuc_dieu_kien). Nếu đúng thì quay lại vòng lặp. Nếu sai thì kết thúc lặp
 - Các **lenh** được thực hiện ít nhất một lần, kể cả khi (bieu_thuc_dieu_kien) sai ngay từ đầu.
 - Nếu (bieu_thuc_dieu_kien) luôn đúng thì lặp vô hạn



- Nhập các số từ bàn phím và tính tổng cho đến khi nhập số 10

Cách 1:

```
S=0;
do { printf("Nhap n = ");
      scanf("%d", &n);
      S=S+n;
} while (n!=10)
```

```
n=3; S=0+3=3
n=5; S=3+5=8
n=10; S=3+5+10=18
```

Cách 2:

```
S=0;
printf("Nhap n = ");
scanf("%d", &n);
while (n!=10){
    s=s+n;
    scanf("%d", &n);
}
```

```
n=3; S=0+3=3
n=5; S=3+5=8
n=10;
```

- 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 thuộc $[0, 10]$ 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 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<=diem<=10):12
```

```
Bạn nhập không đúng!  
Bạn hãy nhập lại (0<=diem<=10):-2
```

```
Bạn nhập không đúng!  
Bạn hãy nhập lại (0<=diem<=10):6.5
```

```
Điểm bạn vừa nhập là: 6.50
```

```
#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("\nBan nhập không đúng!\n");
    } while (diem < 0 || diem > 10);
    printf("\nĐiểm bạn vừa nhập là: %.2f \n", diem);
    return 0;
}
```

- Nhập 1 số nguyên và phân tích số đó ra thừa số nguyên tố

```
#include <stdio.h>
#include <ctype.h>
int main(){
    int N, i;
    char ch;
    do{ printf("\n\nNhap vao so nguyen duong "); scanf("%d",&N);
        printf("%d = ",N);
        i = 2;
        while (i < N ){
            if (N % i == 0){
                printf("%d x ",i);
                N = N/i;
            } else i++;
        }
        printf("%d \n",N);
        printf("Tiep tuc <C/K>?");
        scanf ("%s",&ch);
    }while(toupper(ch) != 'K');
    return 0;
}
```

```
#include <stdio.h>
#include <ctype.h>
int main(){
    int N, i;
    char ch;
    do{
        printf("\n\nNhap vao so nguyen duong ");
        scanf("%d",&N);
        printf("%d = ",N);
        i = 2;
        while (i < N ){
            if (N % i == 0){
                printf("%d x ",i);
                N = N/i;
            } else    i++;
        }
        printf("%d \n",N);
        printf("Tiep tục <C/K>?"); scanf("%s", &ch);
    } while(toupper(ch) != 'K');
    return 0;
}
```

Ví dụ 10 → Kết quả

Nhap vao so nguyen duong 48

$48 = 2 \times 2 \times 2 \times 2 \times 3$

Tiep tục <C/K>?c

Nhap vao so nguyen duong 1024

$1024 = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$

Tiep tục <C/K>?c

Nhap vao so nguyen duong 1001

$1001 = 7 \times 11 \times 13$

Tiep tục <C/K>?c

Nhap vao so nguyen duong 73

$73 = 73$

Tiep tục <C/K>?k

—

Viết chương trình thực hiện công việc:

- Nhập vào từ bàn phím 2 số nguyên a, b
- Nhập vào từ bàn phím một ký tự bất kỳ c;
 - Nếu c 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'


```
1  #include <stdio.h>
2  int main()
3  {
4      int a,b;
5      char ch;
6      printf("Nhap 2 so a va b: "); scanf("%d%d",&a,&b);
7      int tiep=0;
8      do{
9          fflush(stdin); //xoa bo nho dem
10         printf("Nhap cac toan tu +,-,*,/,%%: ");scanf("%c",&ch);
11         switch(ch)
12         {
13             case '+': printf("a + b = %d\n",a+b); break;
14             case '-': printf("a - b = %d\n",a-b); break;
15             case '*': printf("a * b = %d\n",a*b); break;
16             case '/': if (b==0) printf("Khong chia duoc cho 0\n");
17                     else printf("a / b = %d\n",a/b);
18                     break;
19             case '%': if (b==0) printf("Khong chia duoc cho 0\n");
20                     else printf("a %% b = %d\n",a%b);
21                     break;
22             case 'q':
23             case 'Q': tiep=1; break;
24             default: printf("Khong co toan tu nay\n");
25         }
26     } while (tiep==0);
27     return 0;
28 }
```

Ví dụ 11 → Kết quả

C:\Users\84904\OneDrive - Thuyloi University\NHAPMONLAPTRINH\COI

```
Nhap 2 so a va b: 5      2
Nhap cac toan tu +,-,*,/,%: +
a + b = 7
Nhap cac toan tu +,-,*,/,%: -
a - b = 3
Nhap cac toan tu +,-,*,/,%: *
a * b = 10
Nhap cac toan tu +,-,*,/,%: /
a / b = 2
Nhap cac toan tu +,-,*,/,%: %
a % b = 1
Nhap cac toan tu +,-,*,/,%: $
Khong co toan tu nay
Nhap cac toan tu +,-,*,/,%: q
```

```
-----
Process exited after 30.91 seconds with return value 0
Press any key to continue . . .
```

Nhập vào 1 xâu và đếm số kí tự của xâu đó

- 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++) ;
```

- 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);`



Ví dụ 13



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

- Nhập một số nguyên N, kiểm tra n có 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;
}
```

```
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;
}
```

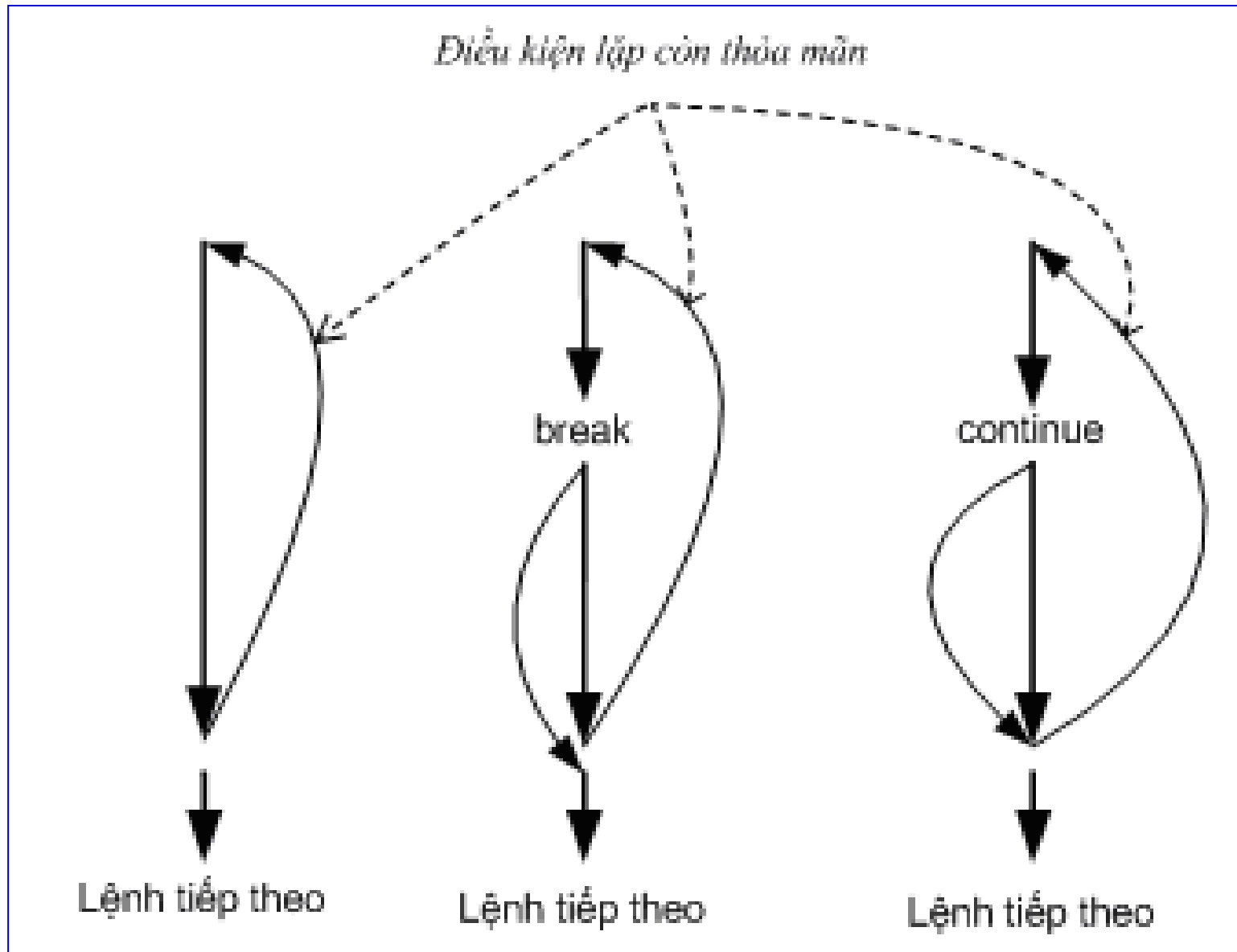
$i \leq (\text{int})\text{sqrt}(N)$

$i = 2;$
 $\text{while } (N \% i \neq 0) \ i++;$
 $\text{if } (i == N) \ \text{printf}(\text{" so nguyen to"})$



3. Các lệnh nhảy

- Các vòng lặp **while/ do ... while/ for** sẽ kết thúc quá trình lặp khi biểu thức điều kiện của vòng lặp không còn đúng nữa.
- 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 đúng.
- Để hỗ trợ người lập trình làm việc đó, ngôn ngữ C cung cấp 2 câu lệnh là **continue** và **break**



- Lệnh nằm trong vòng lặp
- Khi gặp lệnh continue, chương trình 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 % 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 đúng**
- 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

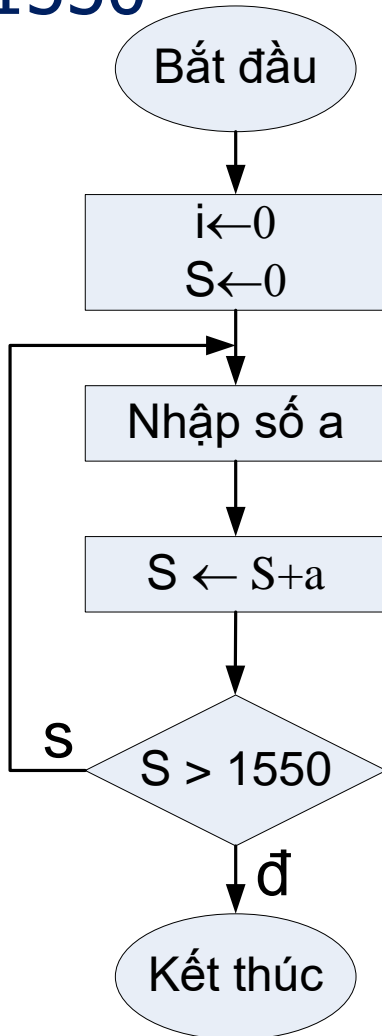

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i = 0; i < 10; i++)
    {
        for(j = 0; j < 10; j++)
        {
            if(j > i)
                break;
        }
        printf("i:%d j: %d\n", i, j);
    }
    return 0;
}
```

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 nhập vào một dãy số cho tới khi

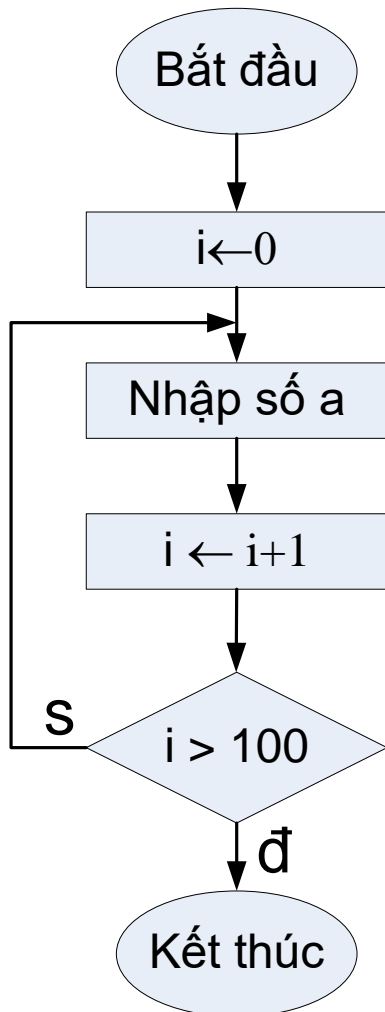
- a. Tổng của dãy lớn hơn 1550 hoặc là
- b. Số phần tử trong dãy bằng 101

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

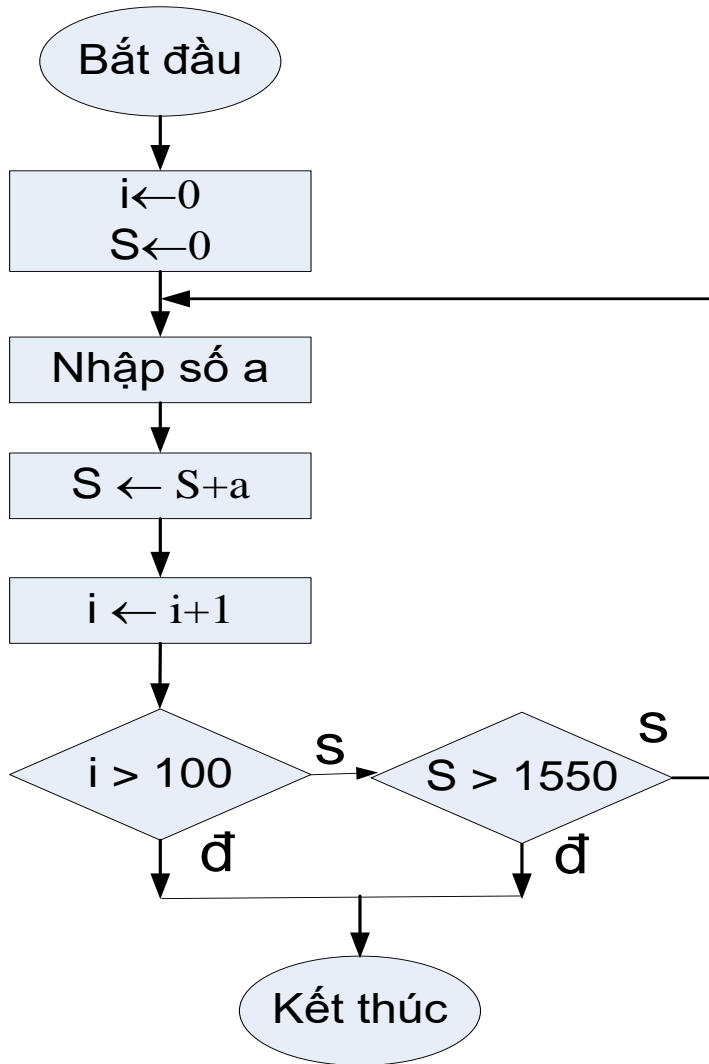


```
#include <stdio.h>
int main(){
    int a, S;
    S = 0;
    do{
        printf("Nhap vao so nguyen:");
        scanf("%d",&a);
        S+=a;
    }while (S <= 1550);
}
```

- Nhập một dãy số cho tới khi số phần tử trong dãy bằng 101



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



```
#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));
}
```

- 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 . . .
```

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. Nhập số n nguyên dương, in ra màn hình n số nguyên kể từ 100 trở đi.
2. Nhập số n nguyên dương, in ra màn hình các số chẵn nhỏ hơn n .
3. Nhập số n nguyên dương, tính giá trị
$$Y = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$
4. Nhập số n nguyên dương, tính giá trị Z :
$$Z = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{1}{2n-1} - \frac{1}{2n}$$
5. Cho số nguyên dương n ($0 < n \leq 10000$). Tính tổng các chữ số của n .
6. Cho hai số nguyên dương a và b ($0 < a, b \leq 10000$). Tìm UCLN và BCNN của a và b .

7. Cho số nguyên dương n ($0 < n \leq 10000$). Kiểm tra xem n có phải là số nguyên tố hay không?
8. Nhập số n nguyên dương, in các số chính phương từ 1 đến n ra màn hình.
9. Nhập số n nguyên dương ($n > 100$), in các số thừa số nguyên tố của số n .
10. Nhập số n nguyên dương, in các số nguyên tố trong khoảng từ 1 đến n ra màn hình.
11. 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 $[a, b]$