



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

TIN HỌC ĐẠI CƯƠNG

Phần 2. Lập trình

Bài 4. Các cấu trúc lập trình trong C

Nội dung



4.1. Cấu trúc lệnh khối

4.2. Cấu trúc rẽ nhánh

4.2.1. Cấu trúc if, if ... else

4.2.2. Cấu trúc lựa chọn switch

4.3. Cấu trúc lặp

4.3.1. Vòng lặp for

4.3.2. Vòng lặp while

4.4. Các lệnh thay đổi cấu trúc lập trình

4.4.1. continue

4.4.2. break

4.1. Cấu trúc lệnh khối

- Thể hiện cấu trúc tuần tự
- Lệnh khối là dãy các câu lệnh được đặt trong cặp dấu ngoặc nhọn { }

```
{  
    lenh_1;  
    lenh_2;  
    ...  
    lenh_n;  
}
```
- C cho phép khai báo biến trong lệnh khối, nhưng phần khai báo phải nằm trước các câu lệnh khác, **biến chỉ tồn tại trong phạm vi của khối.**

4.1. Cấu trúc lệnh khối (2)

- Lệnh khối lồng nhau:
 - Trong một lệnh khối có thể chứa lệnh khối khác
 - Sự lồng nhau là không hạn chế

```
{  
    lệnh;  
    {  
        lệnh;  
        ...  
    }  
    ...  
}
```

4.1. Cấu trúc lệnh khối (3)

- Các lệnh ở khối cha không truy nhập được các biến ở trong khối con
- Các lệnh ở khối con có thể truy nhập được các biến của khối cha

```
{  
    int a = 10;  
    {  
        int b = 20;  
        a += 5; // hợp lệ  
    }  
    b += 10; // không hợp lệ  
}
```

Ví dụ

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

//Nội dung của hàm main() cũng là một khối lệnh
void main()
{
    // khai báo biến
    int c;
    c = 10;
    printf("Giá trị của c = %d đây là c ngoài", c);
```

Ví dụ

```
// bat dau mot khoi lenh khac
{
    int c;
    c = 10;
    printf("\n Gia tri cua c = %d day la c trong", c);
    printf("\n Tang gia tri cua c them 10 don vi");
    c = c + 10;
    printf("\n Gia tri cua c = %d day la c trong", c);
}
printf("\n Gia tri cua c = %d day la c ngoai", c);
getch();
} // ket thuc khoi lenh cua ham main()
```

Ví dụ

- Kết quả:

Gia tri cua $c = 10$ day la c ngoai

Gia tri cua $c = 10$ day la c trong

Tang gia tri cua c them 10 don vi

Gia tri cua $c = 20$ day la c trong

Gia tri cua $c = 10$ day la c ngoai

Nội dung

- 4.1. Cấu trúc lệnh khối



- 4.2. Cấu trúc rẽ nhánh

- 4.2.1. Cấu trúc if, if ... else
- 4.2.2. Cấu trúc lựa chọn switch

- 4.3. Cấu trúc lặp

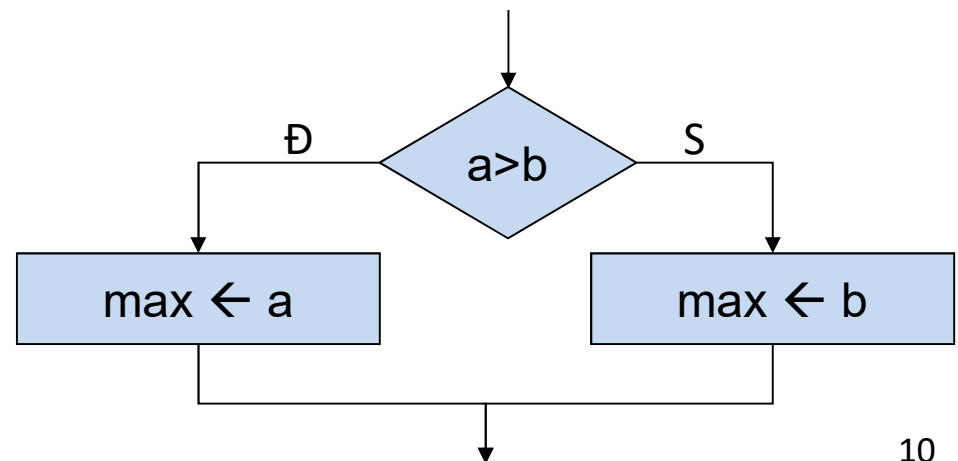
- 4.3.1. Vòng lặp for
- 4.3.2. Vòng lặp while

- 4.4. Các lệnh thay đổi cấu trúc lập trình

- 4.4.1. continue
- 4.4.2. break

4.2.1. Cấu trúc if, if ... else

- Cú pháp cấu trúc **if**
if (bieu_thuc_dieu_kien)
 lenh;
- Cú pháp cấu trúc **if ... else**
if (bieu_thuc_dieu_kien)
 lenh_1;
else
 lenh_2;



Kết hợp lệnh khối

```
if (bieu_thuc_dieu_kien)
{
    lenh_11;
    lenh_12;
}
```

```
if (bieu_thuc_dieu_kien)
{
    lenh_21;
    lenh_22;
}
else
{
    lenh_31;
    lenh_32;
}
```

4.2.1. Cấu trúc if, if ... else (2)

- Ví dụ: Tìm số lớn nhất trong 2 số thực a và b

```
#include <conio.h>
#include <stdio.h>
void main()
{
    // khai bao bien
    float a, b;
    float max;
    printf(" Nhap gia tri a va b: ");
    scanf("%f %f", &a, &b);
```

4.2.1. Cấu trúc if, if ... else (tiếp)

```
if (a < b)
    max = b;
else
    max = a;
printf("\n So lon nhat trong 2 so
    %.0f va %.0f la %.0f ", a, b, max);
getch();
} //ket thuc ham main()
```

- Kết quả:

Nhap gia tri a va b: 23 247

So lon nhat trong hai so 23 va 247 la 247

Ví dụ

- Nhập 3 số thực a, b, c từ bàn phím. Tìm và in ra số lớn nhất và nhỏ nhất.

4.2.2. Cấu trúc lựa chọn switch

- Cú pháp cấu trúc **switch**

```
switch (bieu_thuc)
{
    case gia_tri_1: lenh_1; [break];
    case gia_tri_2: lenh_2; [break];
    ...
    case gia_tri_n: lenh_n; [break];
    [default: lenh_n+1; [break];]
}
```

4.2.2. Cấu trúc lựa chọn switch (2)

- Giá trị của biểu thức kiểm tra (**bieu_thuc**) phải là số nguyên:
 - Phải có kiểu dữ liệu là **char**, **int**, **long**.
- Tương ứng các giá trị sau **case** (gia_tri_1, gia_tri_2,...) cũng phải là số nguyên.

```
int x;  
switch (x)
```



```
char* str;  
switch (str)
```



```
float r;  
switch (r)
```



```
case 'a':
```



```
case 47:
```



```
case 0x34:
```



```
case "Chau":
```



4.2.2. Cấu trúc lựa chọn switch (3)

- Ví dụ: Nhập vào số nguyên không âm, đưa ra ngày trong tuần tương ứng (theo số dư khi chia cho 7).

```
#include <conio.h>
#include <stdio.h>
void main()
{
    int a;
    printf("Nhap mot gia tri so nguyen\nkhong am: ");
    scanf("%d", &a);
```

4.2.2. Cấu trúc lựa chọn switch (4)

```
switch (a % 7)
{
    case 0: printf(" Chu nhat"); break;
    case 1: printf(" Thu Hai"); break;
    case 2: printf(" Thu Ba"); break;
    case 3: printf(" Thu Tu"); break;
    case 4: printf(" Thu Nam"); break;
    case 5: printf(" Thu Sau"); break;
    case 6: printf(" Thu Bay"); break;
}
getch();
}
```

4.2.2. Cấu trúc lựa chọn switch (5)

- Bài tập:
 - 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.

4.2.2. Cấu trúc lựa chọn switch (6)

```
#include <conio.h>
#include <stdio.h>
void main ()
{
    int thang; clrscr();
    printf("\n 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("\n Thang %d co 31 ngay ",thang);
            break;
```

4.2.2. Cấu trúc lựa chọn switch (7)

```
case 4:
case 6:
case 9:
case 11:
    printf("\n Tháng %d có 30 ngày ",thang);
    break;
case 2:
    printf ("\n Tháng 2 có 28 hoặc 29 ngày");
    break;
default :
    printf("\n Không có tháng %d", thang);
    break;
}
getch();
}
```

Bài tập

Bài 1: Viết chương trình nhập 2 hệ số bất kỳ a và b từ bàn phím và giải phương trình bậc nhất $ax + b = 0$

Bài 2: Viết chương trình nhập 3 hệ số bất kỳ a, b và c từ bàn phím và giải phương trình bậc hai $ax^2 + bx + c = 0$

Bài 3: Nhập vào 3 số a, b, c bất kỳ. Kiểm tra xem a, b, c có phải là 3 cạnh của tam giác hay không? Nếu đúng thì là tam giác gì?

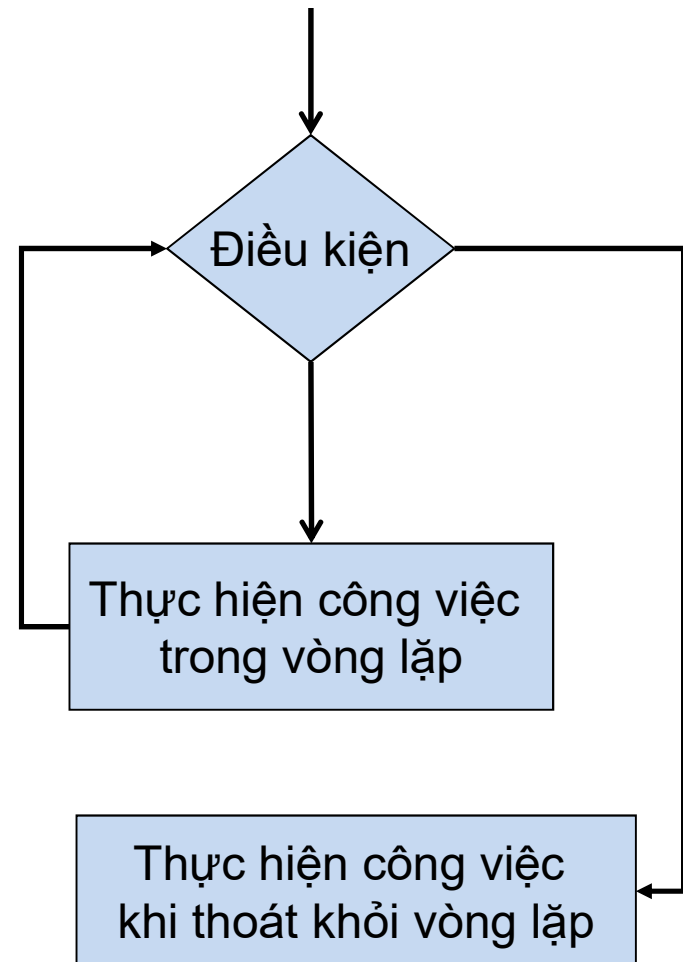
Bài 4: Nhập vào một năm bất kỳ, kiểm tra xem năm đó có phải là năm nhuận không?

Nội dung

- 4.1. Cấu trúc lệnh khối
- 4.2. Cấu trúc rẽ nhánh
 - 4.2.1. Cấu trúc if, if ... else
 - 4.2.2. Cấu trúc lựa chọn switch
- ➔ • 4.3. Cấu trúc lặp
 - 4.3.1. Vòng lặp for
 - 4.3.2. Vòng lặp while
- 4.4. Các lệnh thay đổi cấu trúc lập trình
 - 4.4.1. continue
 - 4.4.2. break

4.3. Cấu trúc lặp

- Một tác động/nhiệm vụ có thể được thực hiện lặp nhiều lần.
- Số lần lặp có thể biết trước hoặc không biết trước. Tuy nhiên số lần lặp phải hữu hạn.
- Kết hợp giữa điều kiện và công việc cần thực hiện



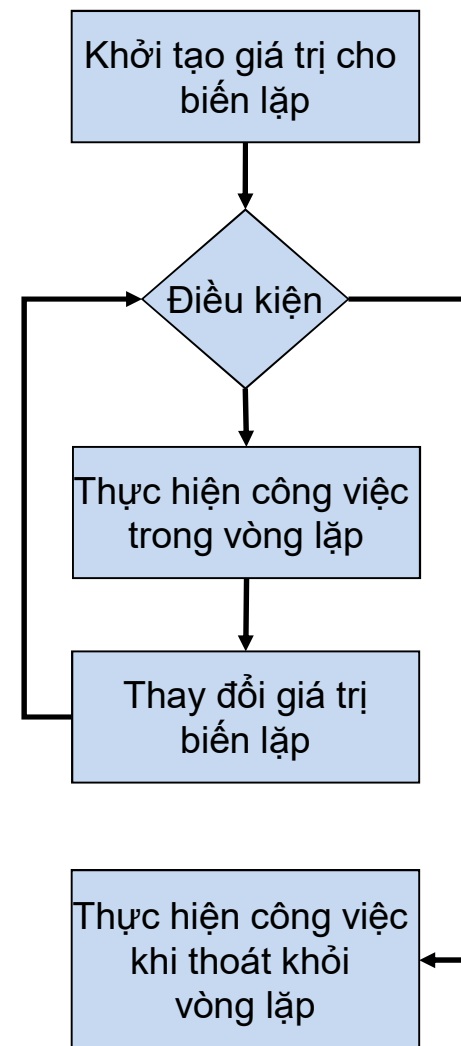
4.3.1. Vòng lặp for

- **Mục đích:** Dùng để thực hiện lặp đi lặp lại một công việc nào đó với số lần lặp xác định.
- **Cú pháp:**

```
for(bieu_thuc_1;bieu_thuc_2;bieu_thuc_3)
{
    day_cac_lenh;
}
```

Trong đó:

- **bieu_thuc_1:** Khởi tạo giá trị ban đầu cho vòng lặp
- **bieu_thuc_2:** Điều kiện tiếp tục vòng lặp
- **bieu_thuc_3:** Thực hiện bước tăng của vòng lặp
- Chú ý các biểu thức 1, 2, 3 có thể có hoặc không



4.3.1. Vòng lặp for (2)

- Ví dụ: Đưa ra màn hình các số nguyên lẻ nhỏ hơn 100

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

4.3.1. Vòng lặp for (3)

- Ví dụ: Đưa ra màn hình các số nguyên lẻ nhỏ hơn 100

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

4.3.1. Vòng lặp for (4)

- Kết quả thực hiện

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

- Nhập vào số nguyên N, tính tổng và tích của các số nguyên từ 1 đến N.

```
#include <stdio.h>
void main()
{
    int i, N, S, P;
    printf("Nhap N = "); scanf("%d", &N);
    S = 0; P = 1;
    for(i = 1; i <= N; i++)
    {
        S += i;
        P *= i;
    }
    printf("S = %d\n", S);
    printf("P = %d\n", P);
}
```

Bài tập

- Lập trình in ra màn hình kết quả như sau (chiều cao n được nhập từ bàn phím)

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
*****
```

Hình 1

```
      *
     **
    ***
   ****
  *****
 *****
*****
*****
*****
*****
*****
*****
```

Hình 2

```
          *
         ***
        *****
       ********
      **********
     **********
    **********
   **********
  **********
 **********
 **********
 **********
 **********
 **********
```

Hình 3

4.3.2. Vòng lặp while

- Mục đích:
 - Dùng để thực hiện lặp đi lặp lại một công việc nào đó với số lần lặp **không** xác định trước.

- Cú pháp: có 2 dạng:

```
while (biểu_thức)
```

```
{
```

```
    lệnh;
```

```
}
```

hoặc

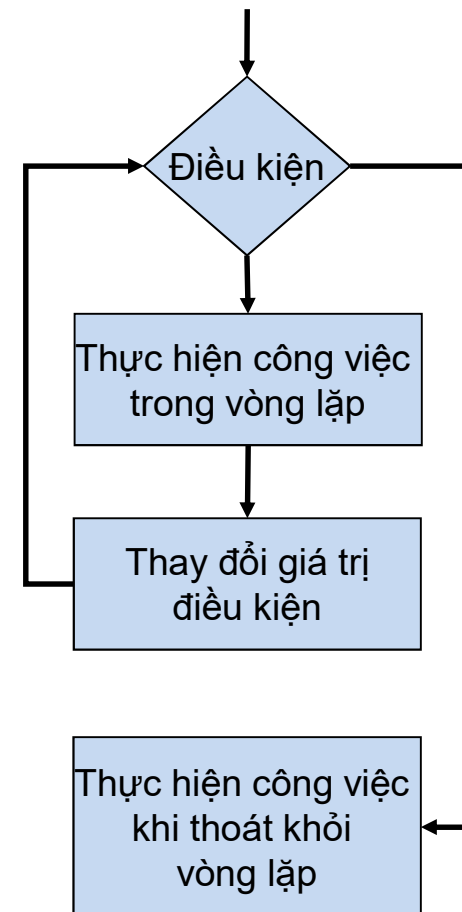
```
do
```

```
{
```

```
    lệnh;
```

```
}
```

```
while (biểu_thức) ;
```



4.3.2. Vòng lặp while (2)

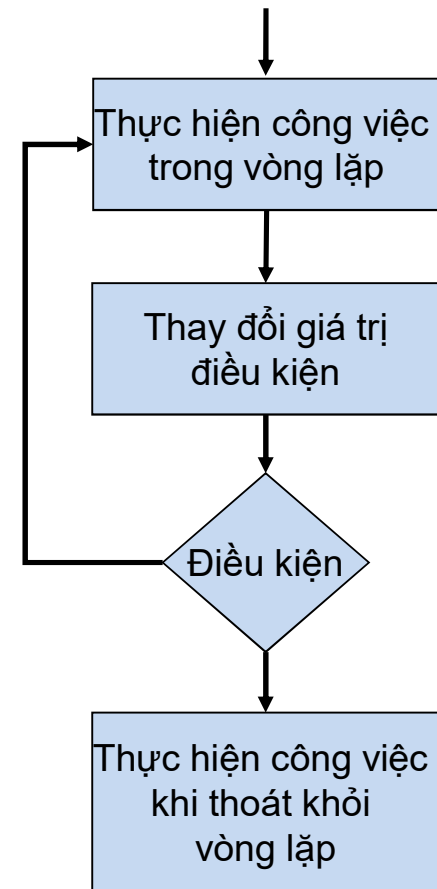
- **while** và **do{...} while**:

- **while**:

- Kiểm tra điều kiện vòng lặp (tức là giá trị của biểu thức) trước rồi mới thực hiện lệnh.
 - Các **lệnh** trong vòng lặp có thể không được thực hiện lần nào.

- **do{...} while**:

- Thực hiện **lệnh** trước rồi mới kiểm tra **dieu_kien** của vòng lặp.
 - Các **lệnh** trong vòng lặp được thực hiện ít nhất 1 lần dù **bieu_thuc** có giá trị như thế nào.



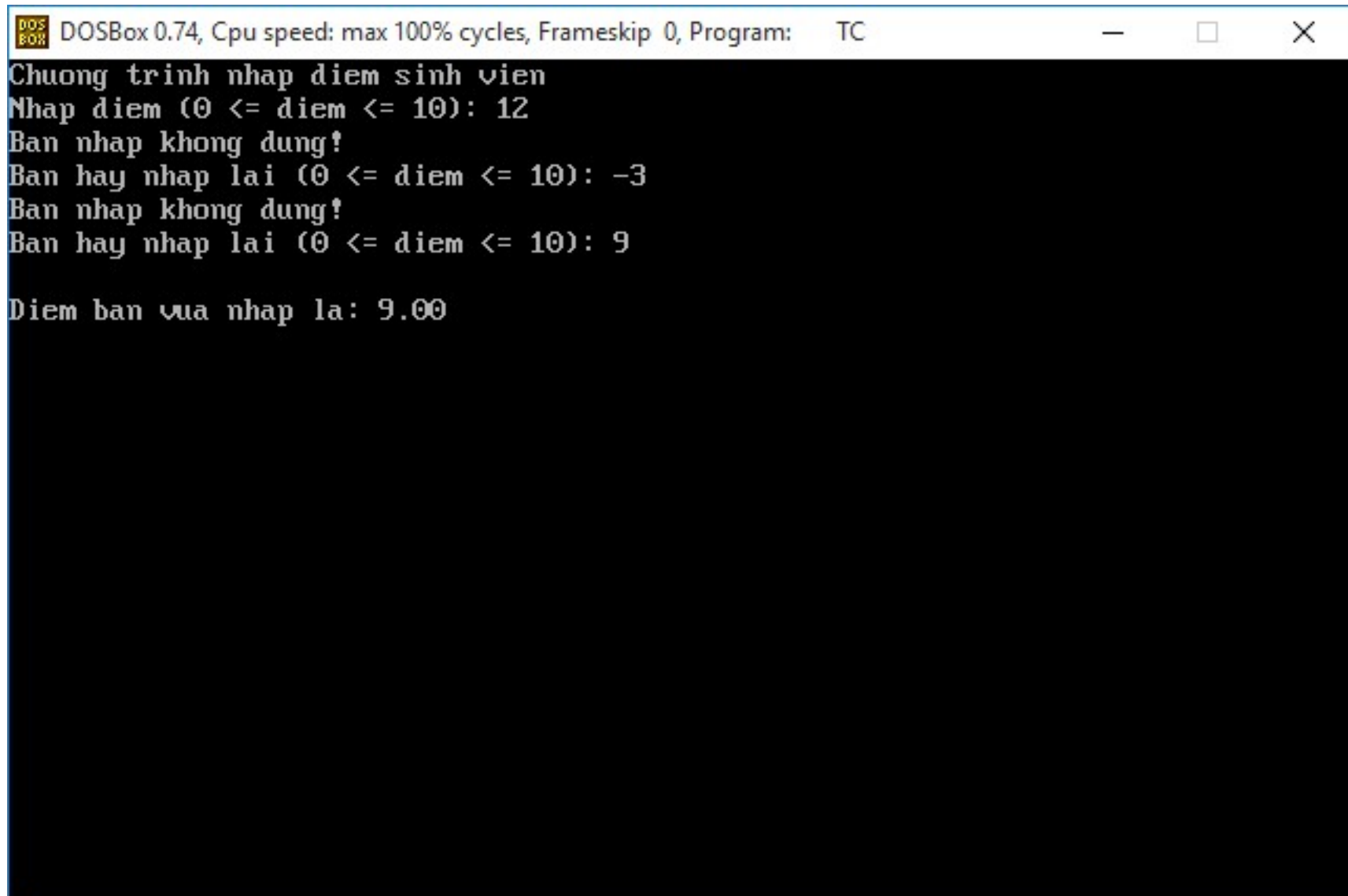
4.3.2. Vòng lặp while (3)

- Ví dụ: nhập vào điểm của một sinh viên, nếu điểm đó không nằm trong đoạn $[0, 10]$ thì thông báo cho người dùng nhập lại.
- Cách làm:
 - Nếu dùng lệnh **if** → chỉ kiểm tra được 1 lần
 - Không dùng **for** được vì chưa biết trước số lần lặp.
 - Sử dụng vòng lặp **while**

4.3.2. Vòng lặp while (4)

```
#include <stdio.h>
void 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", diem);
}
```

4.3.2. Vòng lặp while (5)



```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Chương trình nhập điểm sinh viên
Nhập điểm (0 <= điểm <= 10): 12
Bạn nhập không đúng!
Bạn hãy nhập lại (0 <= điểm <= 10): -3
Bạn nhập không đúng!
Bạn hãy nhập lại (0 <= điểm <= 10): 9
Điểm bạn vừa nhập là: 9.00
```

4.3.2. Vòng lặp while (6)

```
#include <stdio.h>
int main()
{
    float diem;
    printf("Chương trình nhập diem sinh vien\n");
    do
    {
        printf("Nhập diem (0<=diem<=10):");
        scanf("%f", &diem);
        if (diem < 0 || diem > 10)
            printf("\nBan nhập không dung!\n");
    } while (diem < 0 || diem > 10);

    printf("\nDiem ban vua nhập la: %.2f", diem);
    return 0;
}
```

Sửa bài xác định ngày trong tháng

```
#include <stdio.h>
int main()
{
    int thang;
    do {
        printf("Nhap vao thang trong nam ");
        scanf("%d", &thang);
    } while(thang < 1 || thang > 12);

    switch(thang)
    {
        case 2:
            printf ("Thang 2 co 28 hoac 29 ngay");
            break;
```

Sửa bài xác định ngày trong tháng (2)

```
case 4:
case 6:
case 9:
case 11:
    printf("Thang %d co 30 ngay", thang);
    break;
default :
    printf("Thang %d co 31 ngay", thang);
    break;
}

return 0;
}
```

So sánh cấu trúc lặp **for** và **while**

Cấu trúc **for**

- Số lần lặp xác định
- Cần xác định **biến chạy**: giá trị ban đầu, giá trị kết thúc, thay đổi giá trị

```
int i;  
for (i = 0; i < 10; i++)  
{  
    lệnh;  
}
```

Cấu trúc **while**

- Số lần lặp không xác định
- Cần xác **định điều kiện kết thúc vòng lặp**

```
int i = 0;  
while (i < 10) {  
    lệnh;  
    i++;  
}
```

Ví dụ

Nhập vào số nguyên dương N, in ra màn hình tổng các chữ số của N.

```
#include <stdio.h>
void main()
{
    int N, S;
    printf("Nhap N = "); scanf("%d", &N);
    S = 0;
    while (N > 0)
    {
        S += N % 10;
        N /= 10;
    }
    printf("S = %d\n", S);
}
```


Ví dụ

Nhập vào số nguyên dương N, in ra màn hình biểu diễn nhị phân của N.

```
#include <stdio.h>
void main()
{
    int N, S, i;
    printf("Nhap N = "); scanf("%d", &N);
    S = 0; i = 1;
    while (N > 0)
    {
        S += (N % 2) * i;
        N /= 2;
        i *= 10;
    }
    printf("S = %d\n", S);
}
```

Nội dung

- 4.1. Cấu trúc lệnh khối
- 4.2. Cấu trúc rẽ nhánh
 - 4.2.1. Cấu trúc if, if ... else
 - 4.2.2. Cấu trúc lựa chọn switch
- 4.3. Cấu trúc lặp
 - 4.3.1. Vòng lặp for
 - 4.3.2. Vòng lặp while



- 4.4. Các lệnh thay đổi cấu trúc lập trình
 - 4.4.1. continue
 - 4.4.2. break

4.4. Các lệnh thay đổi cấu trúc lập trình

- Đối với các lệnh lặp:
 - **while**, **do{...} while**, hoặc **for**
- Thay đổi việc thực hiện lệnh trong vòng lặp
→ C cung cấp 2 lệnh:
 - **continue**;
 - **break**;

4.4. Các lệnh thay đổi cấu trúc lập trình

- **continue**

- Bỏ qua việc thực hiện các câu lệnh nằm sau lệnh **continue** trong thân vòng lặp.
- Chuyển sang thực hiện một lần lặp mới
- Thường kết hợp với cấu trúc điều kiện **if**

```
int i;  
for (i = 0; i < 10; i++) {  
    lệnh_1;  
    if (điều_kiện)  
        continue;  
    lệnh_2;  
}
```

```
int i = 0;  
while (i < 10) {  
    lệnh_1;  
    if (điều_kiện)  
        continue;  
    lệnh_2;  
    i++;  
}
```

4.4. Các lệnh thay đổi cấu trúc lập trình

- **break**

- Thoát khỏi vòng lặp ngay cả khi biểu thức điều kiện của vòng lặp vẫn còn được thỏa mãn.
- Thường kết hợp với cấu trúc điều kiện **if**

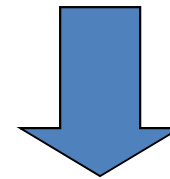
```
int i;  
for (i = 0; i < 10; i++) {  
    lệnh_1;  
    if (điều_kiện)  
        break;  
    lệnh_2;  
}
```

```
int i = 0;  
while (i < 10) {  
    lệnh_1;  
    if (điều_kiện)  
        break;  
    lệnh_2;  
    i++;  
}
```

4.4. Các lệnh thay đổi cấu trúc lập trình

- Ví dụ:

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



```
DOS
BOX DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0,
1 2 3 4 6 7
```

4.4. Các lệnh thay đổi cấu trúc lập trình

- Ví dụ: nhập vào 1 số nguyên. Kết luận số đó là số nguyên tố hay là hợp số?

Ví dụ

```
#include <stdio.h>
int main()
{
    int n, i, flag = 1;
    printf("\nNhập giá trị n : ");
    scanf("%d", &n);
    if (n < 2)
        printf("\nSố %d không là số nguyên tố và không là hợp số.", n);
    else
    {
        for (i = 2; i < n; i++)
            if (n % i == 0)
            {
                flag = 0;
                break;
            }
        if (flag)
            printf("\nSố %d là số nguyên tố.", n);
        else
            printf("\nSố %d là hợp số.", n);
    }
    return 0;
}
```


Ví dụ

```
#include <stdio.h>
#include <math.h>
int main()
{
    int n, i, flag = 1;
    printf("\nNhap gia tri n : ");
    scanf("%d", &n);
    if (n < 2)
        printf("\nSo %d khong la so nguyen to va khong la hop so.", n);
    else
    {
        for (i = 2; i <= (int)sqrt(n); i++)
            if (n % i == 0)
            {
                flag = 0;
                break;
            }
        if (flag)
            printf("\nSo %d la so nguyen to.", n);
        else
            printf("\nSo %d la hop so.", n);
    }
    return 0;
}
```

Bài tập

Bài 1. Lập trình nhập vào số nguyên dương N . In các số nguyên tố nhỏ hơn N .

Bài 2. Lập trình nhập vào 2 số nguyên dương a, b . In ra ước số chung lớn nhất và bội số chung nhỏ nhất của a và b .

Bài 3. Lập trình nhập vào số nguyên dương N . In ra dãy số Fibonacci nhỏ hơn N .

Bài 4. Lập trình nhập vào số nguyên dương N . Kiểm tra xem N có phải là số hoàn hảo hay không?

Bài tập

Bài 5. Viết chương trình yêu cầu người sử dụng nhập vào các số nguyên từ bàn phím cho đến khi nhập số 0 thì kết thúc. Tính tổng các số chẵn và tổng các số lẻ đã được nhập.

Bài 6. Tính xấp xỉ số PI theo công thức

$$\frac{\pi}{4} = 1 + \sum_{i=1}^{\infty} \frac{(-1)^i}{2i+1}$$

Với sai số epsilon nhập từ bàn phím (số hạng cuối cùng có giá trị tuyệt đối không lớn hơn epsilon)