

Bài 5. Giới thiệu một số thư viện và câu lệnh điều khiển

Nội dung

- Giới thiệu các hàm chuẩn trong thư viện `stdio.h`, `conio.h`, `math.h`, `string.h`
- Trình bày các lệnh điều khiển `if`, `for`, `switch`, `while`, `goto`, `continue`, `break`.
- Ví dụ .

1. Một số thư viện – stdio.h

Hàm chuẩn	Ý nghĩa
<code>printf()</code>	In ra màn hình
<code>scanf()</code>	Nhập giá trị cho biến từ bàn phím
<code>puts(s)</code>	In ra màn hình chuỗi kí tự s , dấu hiệu kết thúc là kí tự mã '\0'
<code>gets(s)</code>	Nhập chuỗi kí tự từ bàn phím, dấu hiệu kết thúc là dấu Enter .
<code>getchar()</code>	Nhận một kí tự từ dòng vào chuẩn stdin bàn phím trả về giá trị tiếp theo của stdin

1. Một số thư viện – conio.h

Tên hàm	Ý nghĩa
<code>getch()</code>	nhập một kí tự từ bàn phím nhưng không hiển thị ra màn hình
<code>putch(c)</code>	đưa kí tự c ra màn hình
<code>gotoxy(x, y)</code>	chuyển con trỏ đến vị trí (x,y) trên màn hình văn bản
<code>clrscr()</code>	Xóa màn hình, đưa con trỏ tới đầu màn hình
<code>textcolor(color)</code>	chọn màu hiển thị kí tự là màu color
<code>textbackground(color)</code>	chọn màu nền văn bản là màu color

1. Một số thư viện – math.h

Tên hàm	Ý nghĩa
<code>fabs(x)</code> , <code>abs(x)</code>	$ x $
<code>sin(x)</code> , <code>cos(x)</code> , <code>tan(x)</code>	$\sin x$, $\cos x$, $\tan x$
<code>log(x)</code> , <code>log10(x)</code>	$\ln x$, $\log_{10}x$
<code>pow(y, x)</code> , <code>exp(x)</code>	y^x , e^x
<code>ceil(x)</code>	số nguyên bé nhất lớn hơn x
<code>sqrt(x)</code>	căn bậc 2 của x

1. Một số thư viện – string.h

Tên hàm	Ý nghĩa
<code>strlen(s)</code>	cho độ dài thực của chuỗi kí tự s
<code>strcpy(dest, source)</code>	copy hai chuỗi kí tự
<code>strlwr(s)</code> , <code>strupr(s)</code>	chuyển thành chuỗi kí tự gồm chữ thường, chữ hoa.
<code>strcat(dest, source)</code>	dest = dest + source
<code>strcmp(s1, s2)</code>	So sánh hai chuỗi kí tự theo thứ tự từ điển

1. Một số thư viện – stdlib.h

Tên hàm	Ý nghĩa
<code>int rand()</code>	trả về một số nguyên ngẫu nhiên trong phạm vi từ 0 đến RAND_MAX
<code>int random(int N)</code>	Trả về giá trị nguyên ngẫu nhiên trong phạm vi từ 0 đến (N- 1)

2. Câu lệnh điều khiển

- if, if else
- switch, case
- goto
- for
- while
- continue
- break

Câu lệnh điều khiển if

Có 2 cách viết câu lệnh **if**:

Cách 1:

if (**test**) P **else** Q

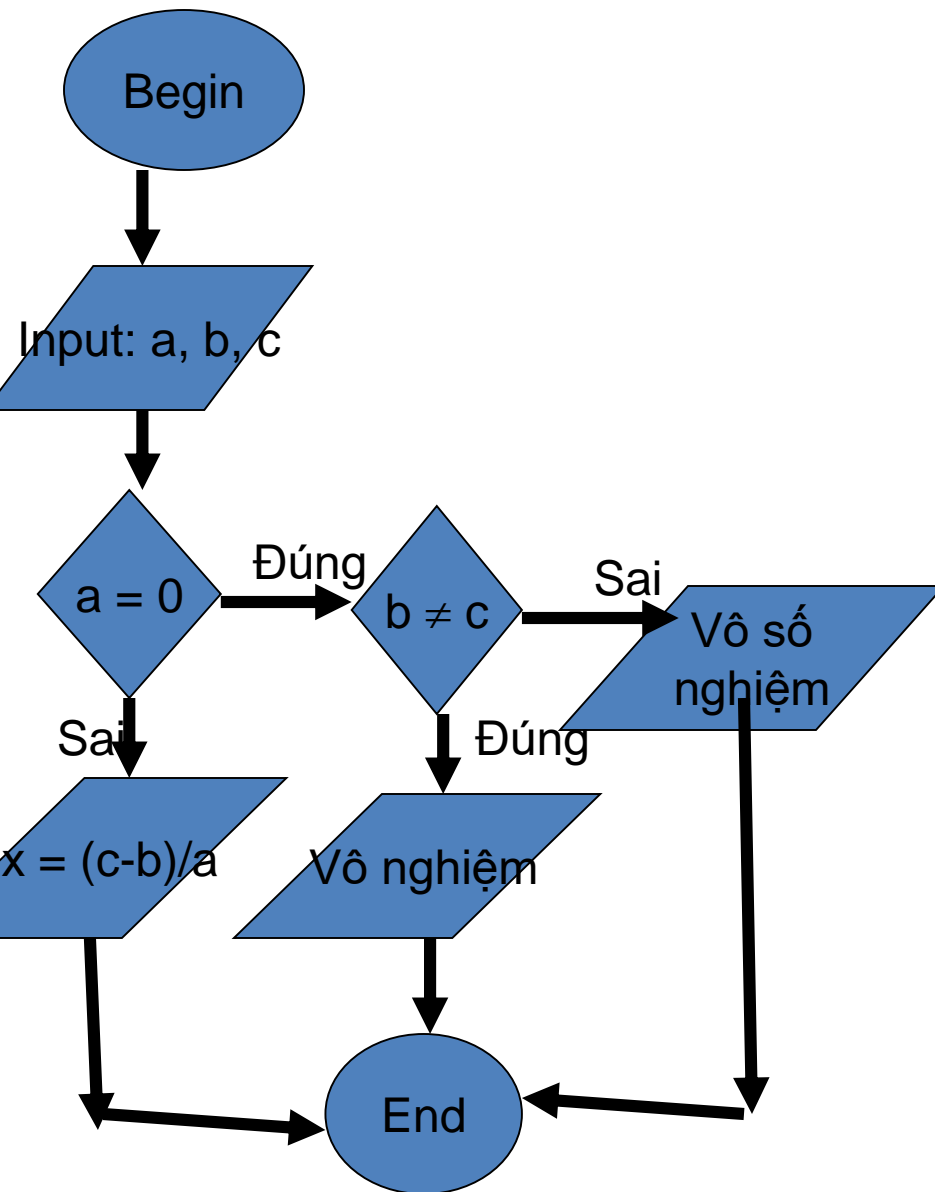
Cách 2:

if (**test**) P

Lưu ý:

- P và Q có thể là câu lệnh hoặc khối lệnh
- Câu lệnh kết thúc bằng dấu chấm phẩy ;
- Khối lệnh được bao bởi cặp dấu { }

Ví dụ: giải phương trình $ax+b=c$



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

```
main()
{
```

```
    float a,b,c;
    printf("Nhap a, b, c=");
    scanf("%f%f%f",&a,&b,&c);
    if (a==0)
        if (b!=c) printf("vo nghiem");
        else printf("vo so nghiem");
    else printf("nghiem = %3.2f", (c-b)/a);
    getch();
}
```

2. Câu lệnh điều khiển **break**

- Cách viết:
break ;
- Câu lệnh **break** dùng để thoát ra khỏi câu lệnh **while**, **do**, **for**, hoặc **switch** trong cùng chứa nó.

Câu lệnh điều khiển **goto**

➤ Cách viết:

Goto nhãn ;

➤ Khi gặp toán tử này, máy sẽ nhảy tới câu lệnh có nhãn viết ngay sau từ khóa **goto**

➤ Lưu ý:

- Nhãn được tạo bởi tên và dấu hai chấm (:)
- Câu lệnh **goto** và **nhãn** phải nằm trong cùng một hàm
- Không dùng goto để nhảy từ ngoài vào trong một khối lệnh, nhưng được nhảy từ trong khối lệnh ra ngoài.

Ví dụ:

```
{  
...  
tt: s += a[i];  
...  
goto tt;  
...  
}
```

2. Câu lệnh điều khiển **goto**

Ví dụ: Viết một trò chơi đoán số với cách chơi như sau:

Máy tính sẽ sinh ra một số nguyên ngẫu nhiên trong đoạn [0, 99]. Người chơi sẽ được đoán 6 lần, với những lần đoán sai máy tính sẽ đưa ra gợi ý. Trong 6 lần đoán, người chơi đoán trúng thì sẽ thắng cuộc, nếu không người chơi là thua.

```
1. #include<stdlib.h>
2. #include<stdio.h>
3. #include<conio.h>
4. int main() {
5.     int may, nguoi, landoan, winner;
6.     clrscr();
7.     srand(1000);
8.     may = random(100);
9.     printf("\nHay doan so
           nguyen trong [0, 99]:\n");
10.    landoan = 0;
11.    winner = 0; /*0: thua, 1: thang*/
12.    Lap:
13.        landoan ++;
14.        printf("\nLan doan thu %d:", landoan);
15.        scanf("%d",&nguoi);
16.        if(may>nguoi) printf("So qua nho");
17.        else if(may < nguoi) printf("So qua lon");
18.        else { winner = 1; goto KT; }
19.        if(landoan<6) goto Lap;
20.    KT:
21.        if(winner) printf("\nChuc mung ban,
                           ban la nguoi thang cuoc");
22.        else {
23.            printf("\nChuc may man lan sau, ban thua cuoc");
24.            printf("\nGia tri can doan la: %d", may);
25.        }
26.        getch();
27.        return 0;
28.    }
29. }
```

Câu lệnh điều khiển **switch**

```
switch (test)
{
    case T1:    các câu lệnh; break;
    case T2:    các câu lệnh; break;
    ...
    case Tn:    các câu lệnh; break;
    [ default: các câu lệnh; ]
}
```

- **test** là biểu thức nguyên
- **T1, T2, ..., Tn** là các giá trị nguyên không trùng nhau.
- Nếu giá trị **test** bằng giá trị **Ti** ($i = 1, 2, \dots, n$) trong **case** nào thì các câu lệnh ứng với **case** đó sẽ được thực hiện.
- Nếu **test** không bằng giá trị nào thì các câu lệnh sau **default**: được thực hiện
- Ra khỏi toán tử **switch**: khi gặp câu lệnh **break** hoặc gặp dấu ngoặc nhọn đóng cuối cùng của thân **switch**.

Câu lệnh điều khiển **switch**

Ví dụ:

- Viết chương trình để nhập tháng của năm bất kỳ, in ra màn hình số ngày có trong tháng của năm đó.
- Phân tích bài toán
 - Số ngày trong từng tháng của năm có qui luật:
 - Các tháng có 31 ngày: 1, 3, 5, 7, 8, 10, 12
 - Các tháng có 30 ngày: 4, 6, 9, 11
 - Tháng 2 có 29 ngày với những năm chia 4 và 28 ngày với những năm còn lại

Câu lệnh điều khiển **switch**

Ví dụ

```
1.  #include<stdio.h>
2.  #include<conio.h>
3.  main() {
4.      int month, year, songay;
5.      printf("\n nhap thang: "); scanf("%d", &month);
6.      printf("\n nhap nam: ");   scanf("%d", &year);
7.      switch(month){
8.          /*tinh cac thang co 31 ngay*/
9.              case 1:
10.             case 3:
11.             case 5:
12.             case 7:
13.             case 8:
14.             case 10:
15.             case 12: songay = 31; break;
16.             /*tinh cac thang co 30 ngay*/
17.             case 4:
18.             case 6:
19.             case 9:
20.             case 11: songay = 30; break;
21.             /*Xet cho thang 2*/
22.             case 2:
23.                 switch(year % 4)
24.                 {
25.                     case 0: songay = 29; break;
26.                     default: songay = 28; break;
27.                 }
28.                 break;
29.             default: printf ("\n du lieu nhap khong
30.             hop le"); goto tt;
31.             }
31.             printf("\n so ngay trong thang %d nam %d
32.                     la %d\n", month, year, songay);
33.             tt:
32.             getch();
33.             }
```


Câu lệnh điều khiển **for**

Dùng để lập các câu lệnh

Cách viết câu lệnh:

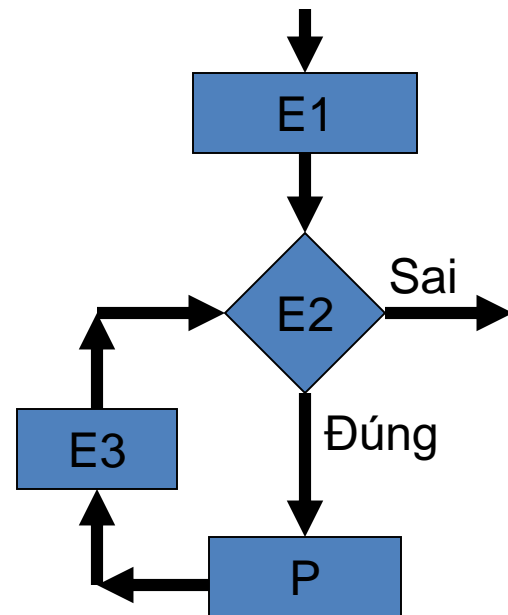
for (E1; E2; E3) P

Ý nghĩa

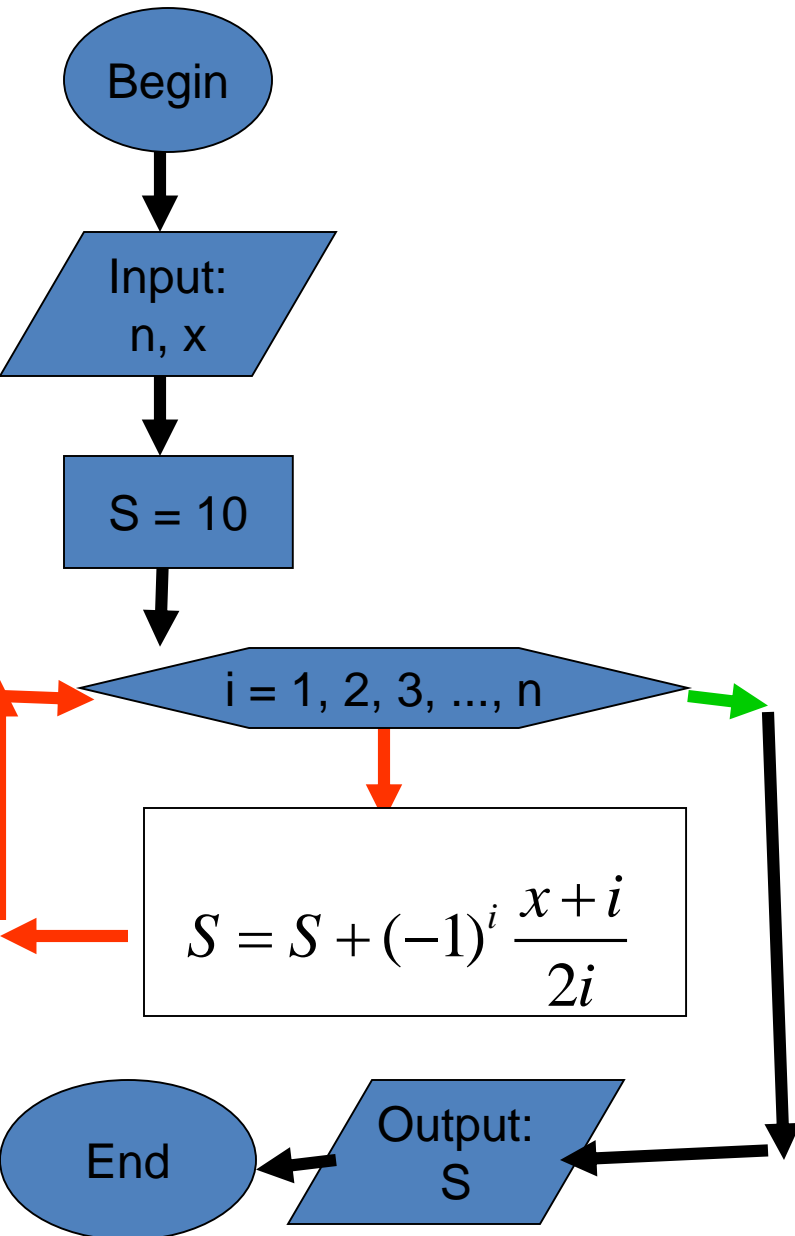
- **P** là câu lệnh hoặc khối lệnh, **P** được thực hiện lặp cho đến khi giá trị của E2 là 0 (điều kiện sai).
- TRƯỚC lần lặp đầu tiên, E1 được tính. E1 thường được dùng để khởi tạo các biến cho vòng lặp.
- SAU mỗi lần lặp, E3 được tính. E3 thường được dùng để tăng giá trị cho biến đếm số lần lặp.

Lưu ý:

- Cả ba biểu thức đều là lựa chọn. Nếu E2 không có, nó được cho là luôn đúng.
- Có thể lồng nhau
- Thoát khỏi vòng lặp **for** bởi câu lệnh **goto**, **break** và **continue**



Ví dụ vòng lặp **for**



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

main()
{
    float S, x;
    int i,n;
    S=10;
    printf("Nhap n= ");
    scanf("%d",&n);
    printf("\nNhap x= ");
    scanf("%f",&x);
    for (i=1; i<=n; i++)
        S+=pow(-1,i)*(x+i)/(2*i);
    printf("S = %3.2f",S);
    getch();
}
```

2. Câu lệnh điều khiển **while**

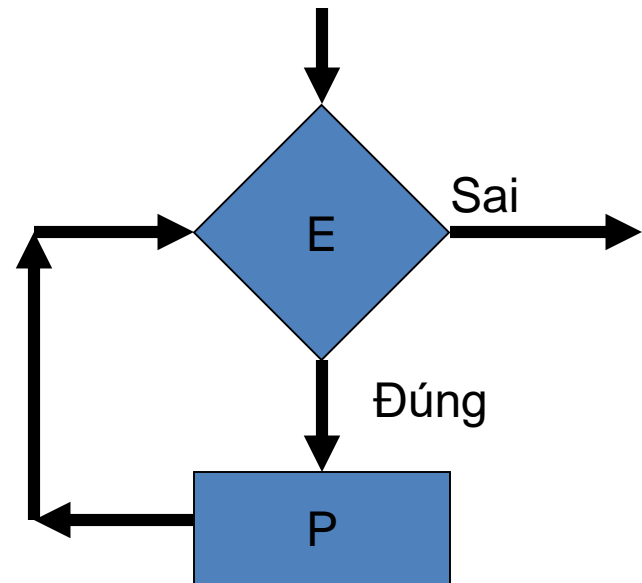
Dùng để lặp

Cách viết

while (E) P

Ý nghĩa

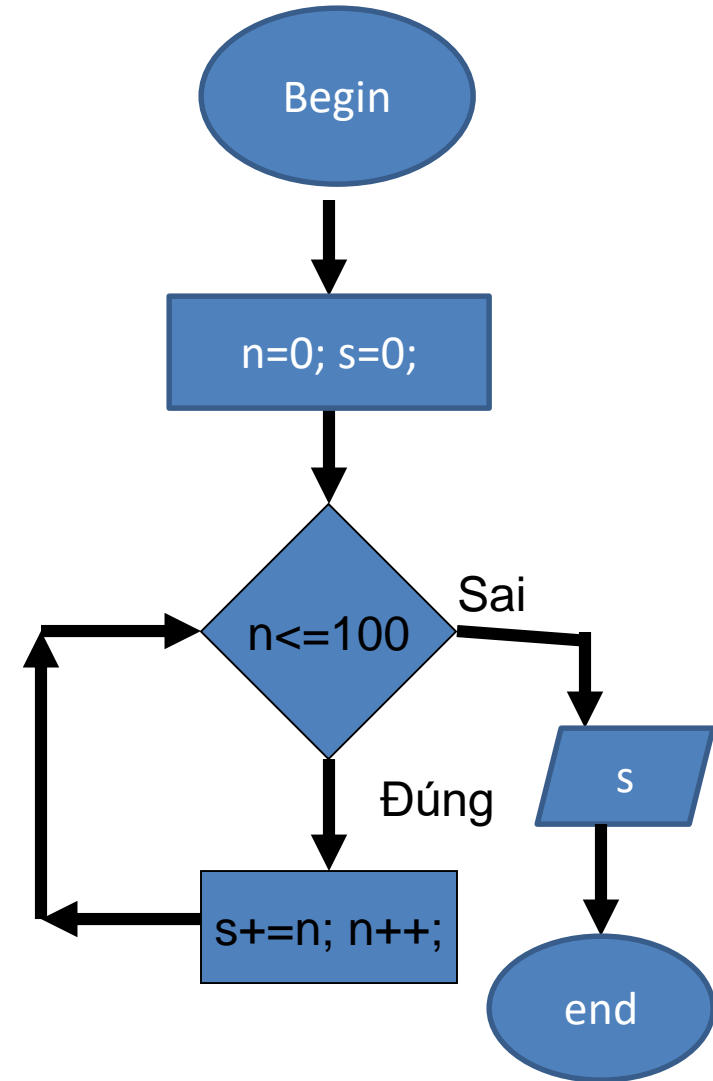
- P là câu lệnh hoặc khối lệnh, P được thực hiện lặp lại khi giá trị của E vẫn còn đúng.
- Việc kiểm tra E được thực hiện TRƯỚC mỗi lần lặp của P.



Ví dụ vòng lặp **while**

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

main()
{
    int n, s;
    s=0;n=0;
    while (n<=100)
    {
        s+=n;
        n++;
    }
    printf("Tong cac so tu nhien <= 100 la %d ",s);
    getch();
}
```



Ví dụ vòng lặp **while** (tiếp)

- Tìm USCLN của 2 số nguyên dương a,b

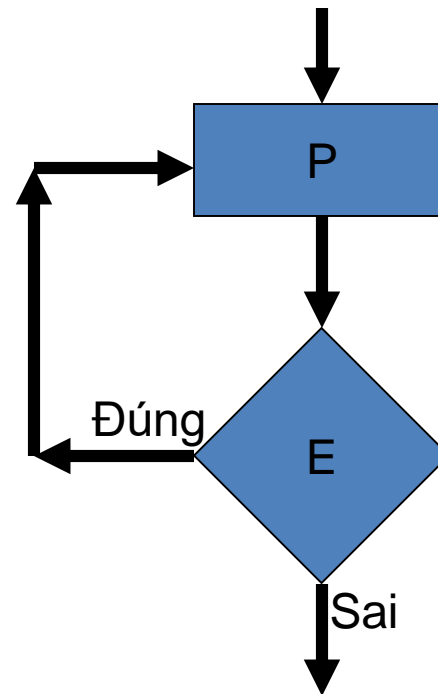
2. Câu lệnh điều khiển **do ... while**

Cách viết câu lệnh

do P **while** (E);

Ý nghĩa

- P là câu lệnh hoặc khối lệnh, P được thực hiện lặp lại khi giá trị của E vẫn còn đúng.
- Việc kiểm tra E được thực hiện SAU mỗi lần lặp P.



2. Câu lệnh điều khiển **continue**

Cách viết:

continue ;

Ý nghĩa:

- Chuyển điều khiển bỏ qua phần còn lại của khối lệnh của **while**, **do**, hoặc **for**, từ điểm này điều kiện lặp tiếp tục được tính lại.

Ví dụ:

```
for (i = 0; i < 20; i++) {  
    if (array[i] == 0)  
        continue;  
    array[i] = 1/array[i];  
}
```