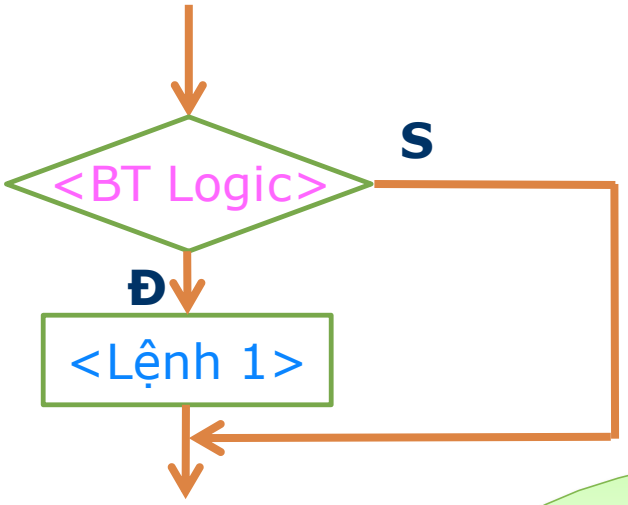
CÁC CẤU TRÚC ĐIỀU KHIỂN

Khoa: Công nghệ thông tin 1

Nội dung

1. Câu lệnh điều kiện if
2. Câu lệnh rẽ nhánh switch
3. Một số kinh nghiệm lập trình
4. Một số ví dụ minh họa

Câu lệnh if (thiếu)




```
if (<BT Logic>)
    <Lệnh 1>;
```

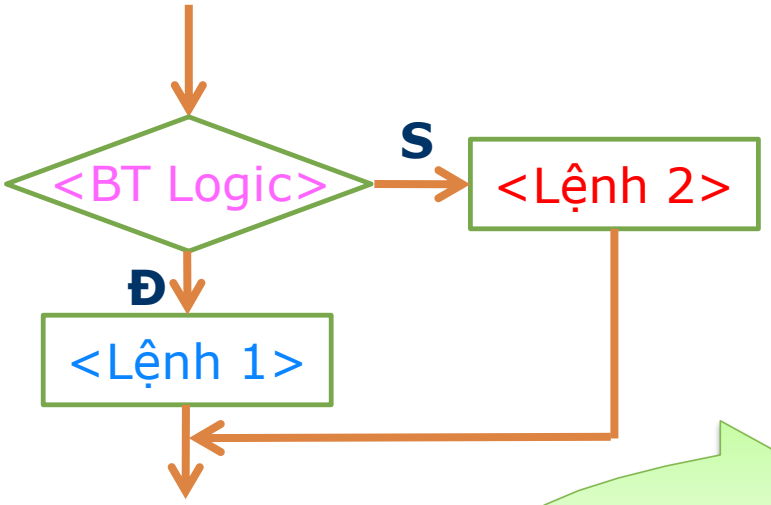
Trong ( ), cho kết quả (sai = 0, đúng ≠ 0)

Câu lệnh đơn hoặc Câu lệnh phức (kẹp giữa { và })

Câu lệnh if (thiếu)

```
void main()
{
        if (a == 0)
                printf("a bang 0");

        if (a == 0)
        {
                printf("a bang 0");
                a = 2912;
        }
}
```

Câu lệnh if (đủ)




```
if (<BT Logic>)
        <Lệnh 1>;
else
        <Lệnh 2>;
```

Trong ( ), cho kết quả (sai = 0, đúng ≠ 0)

Câu lệnh đơn hoặc Câu lệnh phức (kẹp giữa { và })

Câu lệnh if (đủ)

```
void main()
{
        if (a == 0)
                printf("a bang 0");
        else
                printf("a khac 0");

        if (a == 0)
        {
                printf("a bang 0");
                a = 2912;
        }
        else
                printf("a khac 0");
}
```

Câu lệnh if - Một số lưu ý

- Câu lệnh if và câu lệnh if... else là một câu lệnh đơn.

```
{
        if (a == 0)
                printf("a bang 0");
}
{
        if (a == 0)
        {
                printf("a bang 0");
                a = 2912;
        }
        else
                printf("a khac 0");
}
```

Câu lệnh if - Một số lưu ý

- Câu lệnh if có thể lồng vào nhau và else sẽ tương ứng với if gần nó nhất.

```
if (a != 0)
        if (b > 0)
                printf("a != 0 va b > 0");
else
        printf("a != 0 va b <= 0");

if (a !=0)
{
        if (b > 0)
                printf("a != 0 va b > 0");
        else
                printf("a != 0 va b <= 0");
}
```

Câu lệnh if - Một số lưu ý

- Nên dùng else để loại trừ trường hợp.

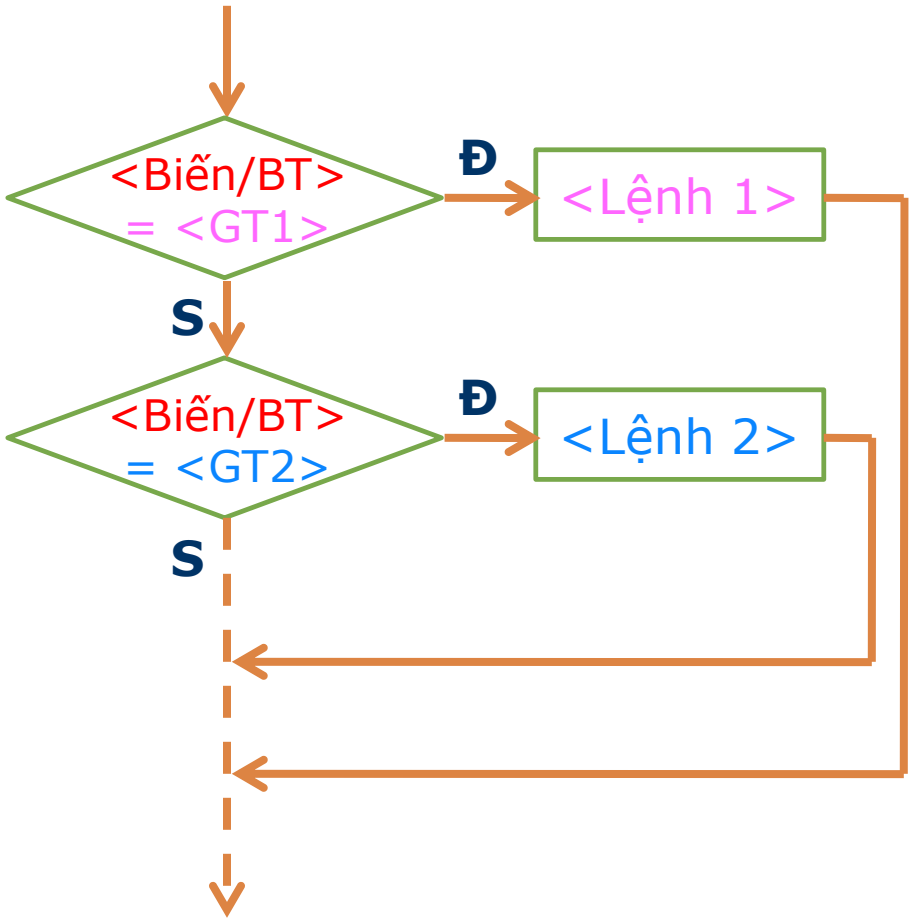
```
if (delta < 0)
      printf("PT vo nghiem");
if (delta == 0)
      printf("PT co nghiem kep");
if (delta > 0)
      printf("PT co 2 nghiem");

if (delta < 0)
      printf("PT vo nghiem");
else  // delta >= 0
      if (delta == 0)
            printf("PT co nghiem kep");
      else
            printf("PT co 2 nghiem");
```

Câu lệnh if - Một số lưu ý

- Không được thêm ; sau điều kiện của if.

```
void main()
{
        int a = 0;
        if (a != 0)
                printf("a khac 0.");

        if (a != 0);
                printf("a khac 0.");

        if (a != 0)
        {
        };
        printf("a khac 0.");
}
```

Câu lệnh switch (thiếu)



switch (<Biến/BT>**)**

{

case <GT1>:<L1>**;break;**

case <GT2>:<L2>**;break;**

…

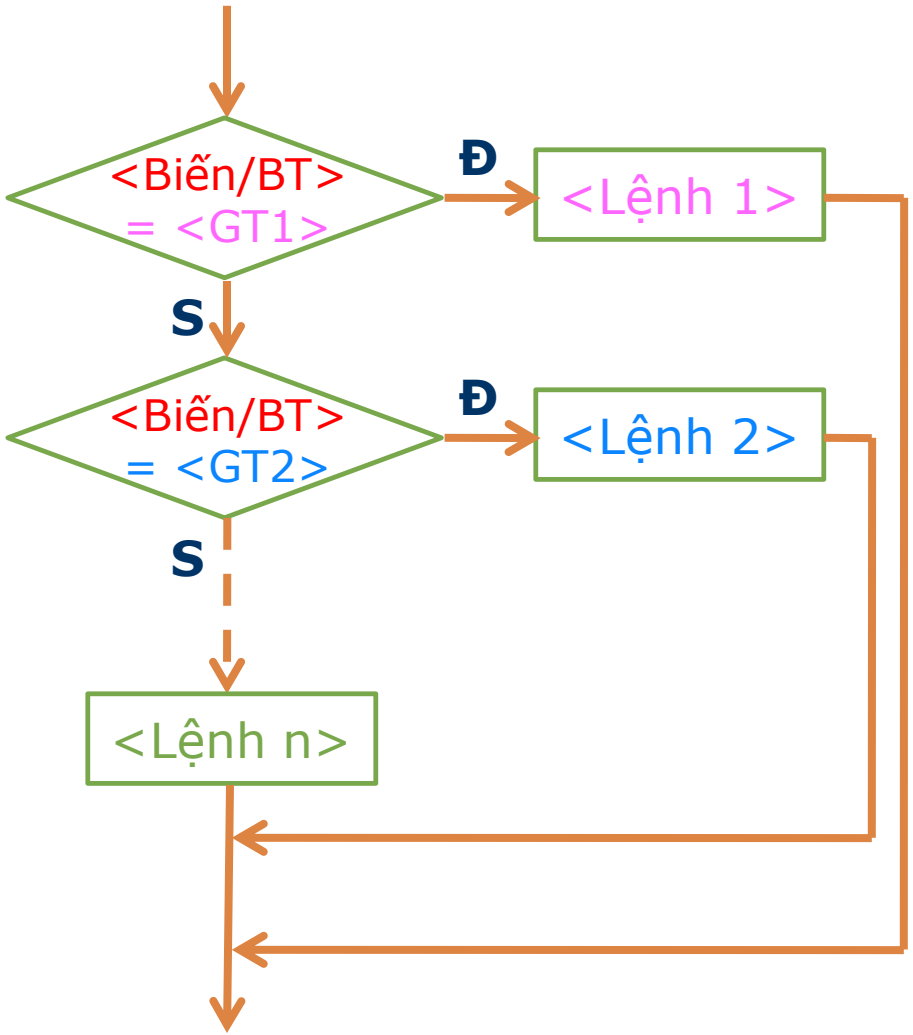
}

- <Biến/BT> là biến/biểu thức cho giá trị rời rạc.
- <Lệnh> : đơn hoặc khối lệnh {}.

Câu lệnh switch (thiếu)

```
void main()
{
        int a;
        printf("Nhap a: ");
        scanf("%d", &a);

        switch (a)
        {
                case 1 : printf("Mot"); break;
                case 2 : printf("Hai"); break;
                case 3 : printf("Ba"); break;
        }
}
```

Câu lệnh switch (đủ)




```
switch (<Biến/BT>)
{
   <GT1>:<Lệnh 1>;break;
   <GT2>:<Lệnh 2>;break;
   …
   default:
         <Lệnh n>;
}
```

Câu lệnh switch (đủ)

```
void main()
{
        int a;
        printf("Nhap a: ");
        scanf("%d", &a);

        switch (a)
        {
                case 1 : printf("Mot"); break;
                case 2 : printf("Hai"); break;
                case 3 : printf("Ba"); break;
                default : printf("Ko biet doc");
        }
}
```

Câu lệnh switch - Một số lưu ý

- Câu lệnh switch là một câu lệnh đơn và có thể lồng nhau.

```
{
        switch (a)
        {
                case 1 : printf("Mot"); break;
                case 2 : switch (b)
                         {
                            case 1 : printf("A"); break;
                            case 2 : printf("B"); break;
                         } break;
                case 3 : printf("Ba"); break;
                default : printf("Khong biet doc");
        }
}
```

Câu lệnh switch - Một số lưu ý

- Các giá trị trong mỗi trường hợp phải khác nhau.

```
switch (a)
{
        case 1 : printf("Mot"); break;
        case 1 : printf("MOT"); break;
        case 2 : printf("Hai"); break;
        case 3 : printf("Ba"); break;
        case 1 : printf("1"); break;
        case 1 : printf("mot"); break;
        default : printf("Khong biet doc");
}
```

Câu lệnh switch - Một số lưu ý

- switch sẽ nhảy đến case tương ứng và thực hiện đến khi nào gặp break hoặc cuối switch sẽ kết thúc.

```
switch (a)
{
        case 1 : printf("Mot"); break;
        case 2 : printf("Hai"); break;
        case 3 : printf("Ba"); break;
}
```

Câu lệnh switch - Một số lưu ý

- switch nhảy đến case tương ứng và thực hiện đến khi nào gặp break hoặc cuối switch sẽ kết thúc.

```
switch (a)
{
        case 1 : printf("Mot"); break;
        case 2 : printf("Hai"); break;
        case 3 : printf("Ba"); break;
}
switch (a)
{
        case 1 : printf("Mot"); break;
        case 2 : printf("Hai"); break;
        case 3 : printf("Ba"); break;
}
```

Câu lệnh switch - Một số lưu ý

- Tận dụng tính chất khi bỏ break;

```
switch (a)
{
        case 1 : printf("So le"); break;
        case 2 : printf("So chan"); break;
        case 3 : printf("So le"); break;
        case 4 : printf("So chan"); break;
}
switch (a)
{
        case 1 :
        case 3 : printf("So le"); break;
        case 2 :
        case 4 : printf("So chan"); break;
}
```

Kinh nghiệm lập trình

Câu lệnh if

```
if (a == 1)
   printf("Mot");
if (a == 2)
   printf("Hai");
if (a == 3)
   printf("Ba");
if (a == 4)
   printf("Bon");
if (a == 5)
   printf("Nam");
```

Câu lệnh switch

```
switch (a)
{
   case 1:  printf("Mot");
            break;
   case 2:  printf("Hai");
            break;
   case 3:  printf("Ba");
            break;
   case 4:  printf("Bon");
            break;
   case 5:  printf("Nam");
}
```

Kinh nghiệm lập trình

❖ Câu lệnh switch

```
switch (a)
{
case 3.14:
case <10:
case 1: printf("OK");
        break;
case 2:
case 3: printf("OK");
        break;
}
```

❖ Câu lệnh if

```
if (a == 3.14)
    printf("OK");
if (a < 10)
    printf("OK");
if (a == 1)
    printf("OK");
if (a == 2 || a == 3)
    printf("OK");
```

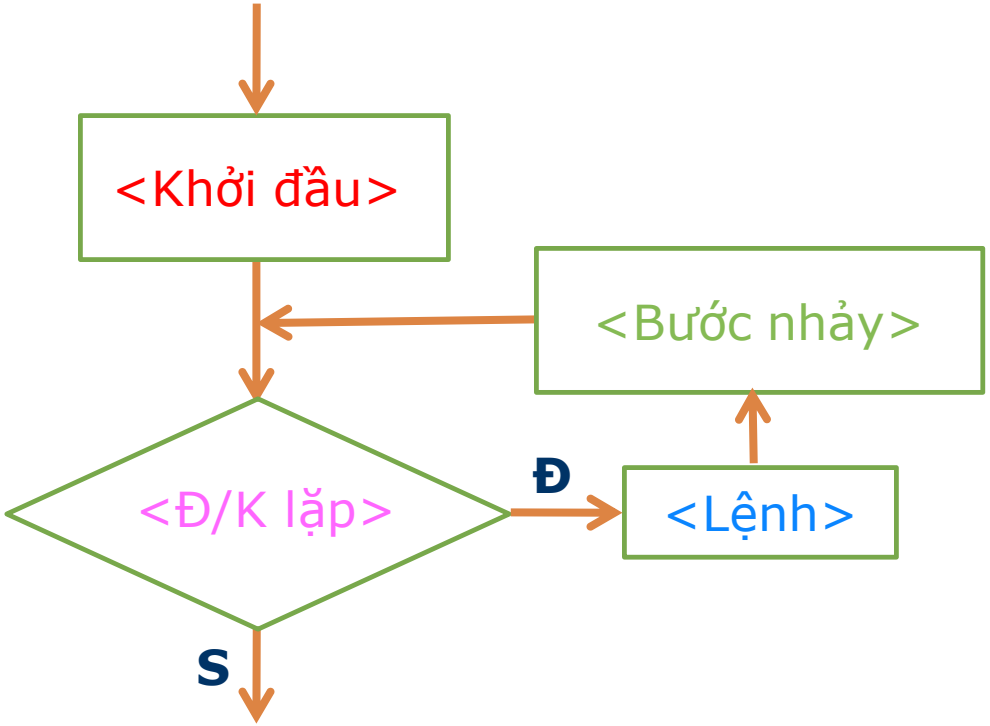
CÁC CẤU TRÚC LẶP

1. Câu lệnh for
2. Câu lệnh while
3. Câu lệnh do… while
4. Một số kinh nghiệm lập trình

Đặt vấn đề

- Ví dụ
  - Viết chương trình xuất các số từ 1 đến 10
  - => Dùng 10 câu lệnh printf
  - Viết chương trình xuất các số từ 1 đến 1000
  - => Dùng 1000 câu lệnh printf!!!
- Giải pháp
  - Sử dụng cấu trúc lặp lại một hành động trong khi còn thỏa một điều kiện nào đó.
  - 3 lệnh lặp: FOR, WHILE, DO… WHILE

Câu lệnh for



for (<Khởi đầu>; <Đ/K lặp>; <Bước nhảy>)

<Lệnh>;

<Khởi đầu>, <Đ/K lặp>, <Bước nhảy>: là biểu thức C bất kỳ có chức năng riêng

<Lệnh>: đơn hoặc khối lệnh.

Câu lệnh for

```
void main()
{
	int i;
	for (i = 0; i < 10; i++)
		printf("%d\n", i);

	for (int j = 0; j < 10; j = j + 1)
		printf("%d\n", j);

	for (int k = 0; k < 10; k += 2)
	{
		printf("%d", k);
		printf("\n");
	}
}
```

Câu lệnh for - Một số lưu ý

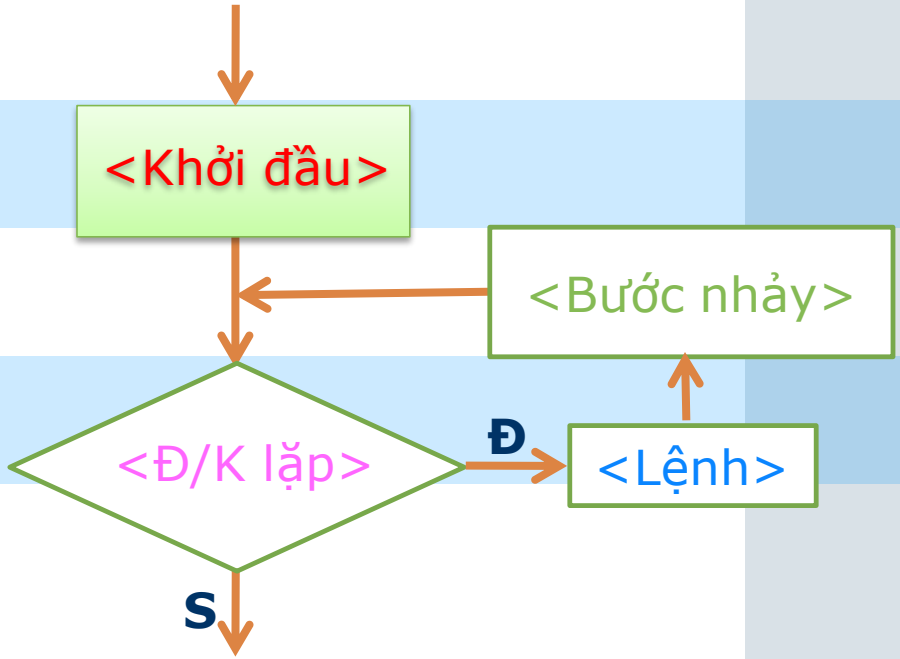
- Câu lệnh FOR là một câu lệnh đơn và có thể lồng nhau.

```
if (n < 10 && m < 20)
{
	for (int i = 0; i < n; i++)
	{
		for (int j = 0; j < m; j++)
		{
			printf("%d", i + j);
			printf("\n");
		}
	}
}
```

Câu lệnh for - Một số lưu ý

- Trong câu lệnh for, có thể sẽ không có phần <Khởi đầu>

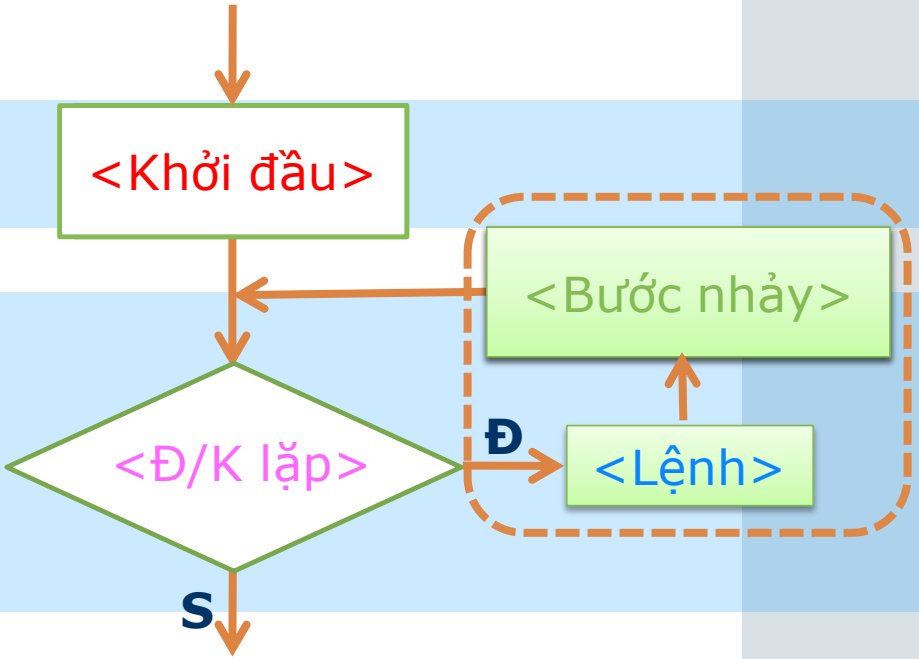
```
int i;
for (i = 0; i < 10; i++)
      printf("%d\n", i);

int i = 0;
for (; i < 10; i++)
      printf("%d\n", i);
```



Câu lệnh for - Một số lưu ý

- Trong câu lệnh for, có thể sẽ không có phần <Bước nhảy>

```
int i;
for (i = 0; i < 10; i++)
      printf("%d\n", i);

for (i = 0; i < 10; )
{
      printf("%d\n", i);
      i++;
}
```



Câu lệnh for - Một số lưu ý

- Trong câu lệnh for, có thể sẽ không có phần <Đ/K lặp>

```
int i;
for (i = 0; i < 10; i++)
      printf("%d\n", i);

for (i = 0; ; i++)
      printf("%d\n", i);

for (i = 0; ; i++)
{
      if (i >= 10)
            break;
      printf("%d\n", i);
}
```

Câu lệnh for - Một số lưu ý

- Lệnh break làm kết thúc câu lệnh.
- Lệnh continue bỏ qua lần lặp hiện tại.

```
for (i = 0; i < 10; i++)
{
        if (i % 2 == 0)
                break;
        printf("%d\n", i);
}

for (i = 0; i < 10; i++)
{
        if (i % 2 == 0)
                continue;
        printf("%d\n", i);
}
```

Câu lệnh for - Một số lưu ý

- Không được thêm ; ngay sau lệnh lệnh for.

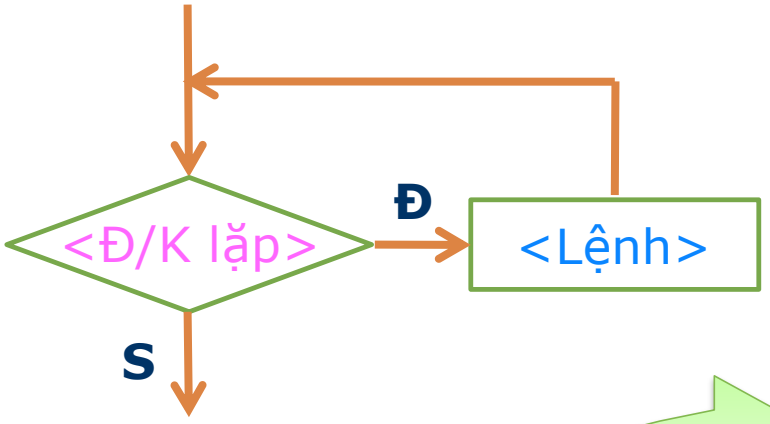
```
for (i = 0; i < 10; i++);
{
      printf("%d", i);
      printf("\n");
}

for (i = 0; i < 10; i++)
{
};
{
      printf("%d", i);
      printf("\n");
}
```

Câu lệnh for - Một số lưu ý

- Các thành phần <Khởi đầu>, <Đ/K lặp>, <Bước nhảy> cách nhau bằng dấu ;
- Nếu có nhiều thành phần trong mỗi phần thì được cách nhau bằng dấu ,

```
for (int i = 1, j = 2; i + j < 10; i++, j += 2)
      printf("%d\n", i + j);
```

Câu lệnh while



while (<Đ/K lặp>**)**

<Lệnh>;

Biểu thức C bất kỳ, thường là biểu thức quan hệ cho kết quả 0 (sai) và != 0 (đúng)

Câu lệnh đơn hoặc Câu lệnh phức (kẹp giữa { và })

Câu lệnh while

```
int i = 0;
while (i < 10)
{
        printf("%d\n", i);
        i++;
}

for (int i = 0; i < 10; i++)
        printf("%d\n", i);

int i = 0;
for (; i < 10; )
{
        printf("%d\n", i);
        i++;
}
```

Câu lệnh while - Một số lưu ý

- Câu lệnh while là một câu lệnh đơn và có thể lồng nhau.

```
if (n < 10 && m < 20)
{
        while (n >= 1)
        {
                while (m >= 1)
                {
                        printf("%d", m);
                        m--;
                }
                n--;
        }
}
```

Câu lệnh while - Một số lưu ý

- Câu lệnh while có thể không thực hiện lần nào do điều kiện lặp ngay từ lần đầu đã không thỏa.

```
void main()
{
        int n = 1;
        while (n > 10)
        {
                printf("%d\n", n);
                n--;
        }
        …
}
```

Câu lệnh for - Một số lưu ý

- Không được thêm ; ngay sau lệnh lệnh while.

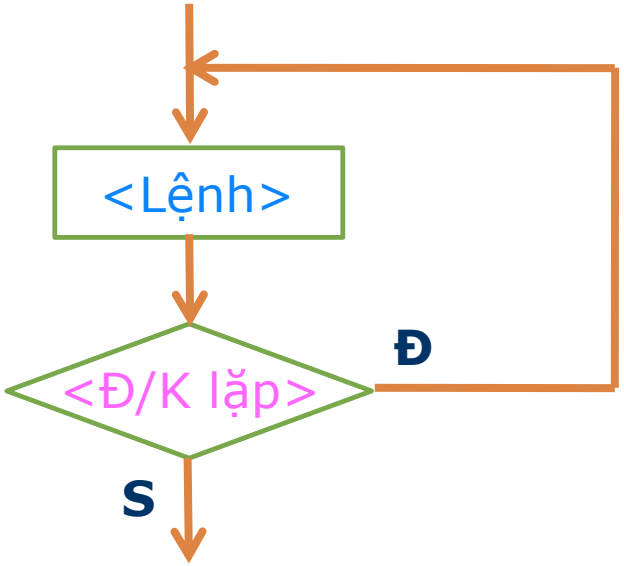
```
int n = 0;
while (n < 10);
{
        printf("%d\n", n);
        n++;
}

while (n < 10)
{
};
{
        printf("%d\n", n);
        n++;
}
```

Câu lệnh WHILE - Một số lưu ý

- Câu lệnh while có thể bị lặp vô tận (loop)

```
void main()
{
        int n = 1;
        while (n < 10)
        {
                printf("%d", n);
                n--;
        }

        n = 1;
        while (n < 10)
                printf("%d", n);
}
```

Câu lệnh do… while




```
do
        <Lệnh>;
while (<Đ/K lặp>);
```

Câu lệnh đơn hoặc Câu lệnh phức (kẹp giữa { và })

Biểu thức C bất kỳ, thường là biểu thức quan hệ cho kết quả 0 (sai) và != 0 (đúng)

Câu lệnh do… while

```
int i = 0;
do
{
        printf("%d\n", i);
        i++;
}
while (i < 10);
```

```
int i = 0;
printf("%d\n", i);
i++;
for (; i < 10; )
{
        printf("%d\n", i);
        i++;
}
```

Câu lệnh do… while - Một số lưu ý

- Câu lệnh do… while là một câu lệnh đơn và có thể lồng nhau.

```
int a = 1, b;
do
{
        b = 1;
        do
        {
                printf("%d\n", a + b);
                b = b + 2;
        }
        while (b < 20);
        a++;
}
while (a < 20);
```

Câu lệnh do while- Một số lưu ý

- Câu lệnh do… while sẽ được thực hiện ít nhất 1 lần do điều kiện lặp được kiểm tra ở cuối.

```
void main()
{
        int n;
        do
        {
                printf("Nhap n: ");
                scanf("%d", &n);
        }
        while (n < 1 || n > 100);
}
```

Câu lệnh WHILE - Một số lưu ý

- Câu lệnh do… while có thể bị lặp vô tận (loop)

```
…
        int n = 1;
        do
        {
                printf("%d", n);
                n--;
        }
        while (n < 10);

        n = 1;
        do
                printf("%d", n);
        while (n < 10);
…
```

FOR, WHILE & DO… WHILE

- Đều có khả năng lặp lại nhiều hành động.

```
int n = 10;
for (int i = 1; i <= n; i++)
      printf("%d\n", i);

int i = 1;
while (i <= n)
{
      printf("%d\n", i); i++;
}

int i = 1;
do {
      printf("%d\n", i); i++;
} while (i > n);
```

FOR, WHILE & DO… WHILE

- Số lần lặp xác định ngay trong câu lệnh for

```
int n = 10;
for (int i = 1; i <= n; i++)
        …;
```

```
int i = 1;
while (i <= n)
{
        …;
}
```

```
int i = 1;
do {
        …;
} while (i > n);
```

WHILE & DO… WHILE

- while có thể không thực hiện lần nào.
- do… while sẽ được thực hiện ít nhất 1 lần.

```
int n = 100;
while (n < 10)
{
        …;
}
…
do
{
        printf("Nhap n: ");
        scanf("%d", &n);
}
while (n > 10);
```

CÁC CÂU LỆNH ĐẶC BIỆT

- Lệnh break làm kết thúc câu lệnh.
- Lệnh continue bỏ qua lần lặp hiện tại.

Ví dụ:

```
while (x != y)
        {     …
            if (x==a) continue;
            b+=6;
            …
            if (y==b) break;
            …
        }
```