Câu lệnh lặp while, do...while

Mục tiêu

- Ý nghĩa, cách hoạt động của vòng lặp.
- Cú pháp, ý nghĩa, cách sử dụng lệnh while, do…while.
- Ý nghĩa và cách sử dụng lệnh break, continue.
- Một số bài toán sử dụng lệnh while, do…while thông qua các ví dụ.
- So sánh, đánh giá một số bài toán sử dụng lệnh for, while hoặc do…while.
- Cấu trúc vòng lặp lồng nhau.

Nội dung

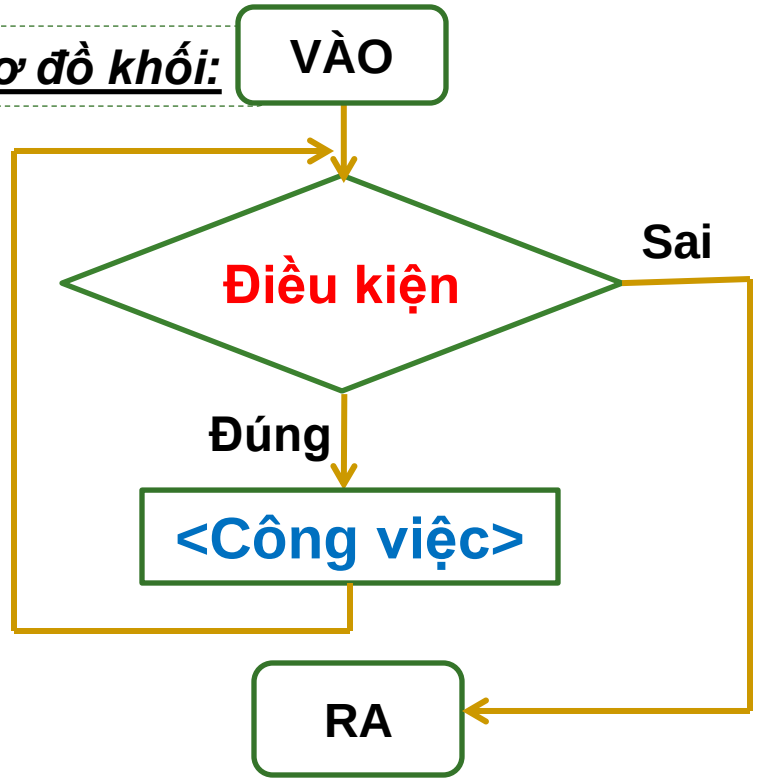
Câu lệnh while

Câu lệnh do .. while

Một số kinh nghiệm lập trình

Câu lệnh while

Sơ đồ khối:



Hoạt động:

<Biểu thức> được kiểm tra:
- Nếu Sai:
  + Kết thúc vòng lặp while
  + <Công việc> không được thực hiện một lần nào.
- Nếu Đúng:
  + Thực hiện <Công việc>;
  + Lặp lại kiểm tra <Biểu thức>

Cú pháp:

while (<Biểu thức>)

<Công việc>

<Công việc>:
- Câu lệnh đơn
- Câu lệnh phức đặt trong cặp { và }

Câu lệnh while

- Biểu thức: có thể là một biểu thức hoặc nhiều biểu thức con. Nếu là nhiều biểu thức con thì cách nhau bởi dấu phẩy (,) và tính đúng sai của biểu thức được quyết định bởi biểu thức con cuối cùng.
- Trong thân while (Công việc) có thể chứa một hoặc nhiều cấu trúc điều khiển khác.
- Trong thân while có thể sử dụng lệnh continue để chuyển đến đầu vòng lặp (bỏ qua các câu lệnh còn lại trong thân).
- Muốn thoát khỏi vòng lặp while tùy ý có thể dùng các lệnh break, goto, return như lệnh for.

Câu lệnh while vs for

```
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\n", 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 sai.

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

Câu lệnh while – 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

- **VD1: Viết ra 3 dòng MSG**

```
a. int i = 0;
   while (i++ < 3)
      printf("%s", MSG);
b. while (printf("%s", MSG), ++i < 3);
```

- **VD2: Tính tổng từ 1..n.**

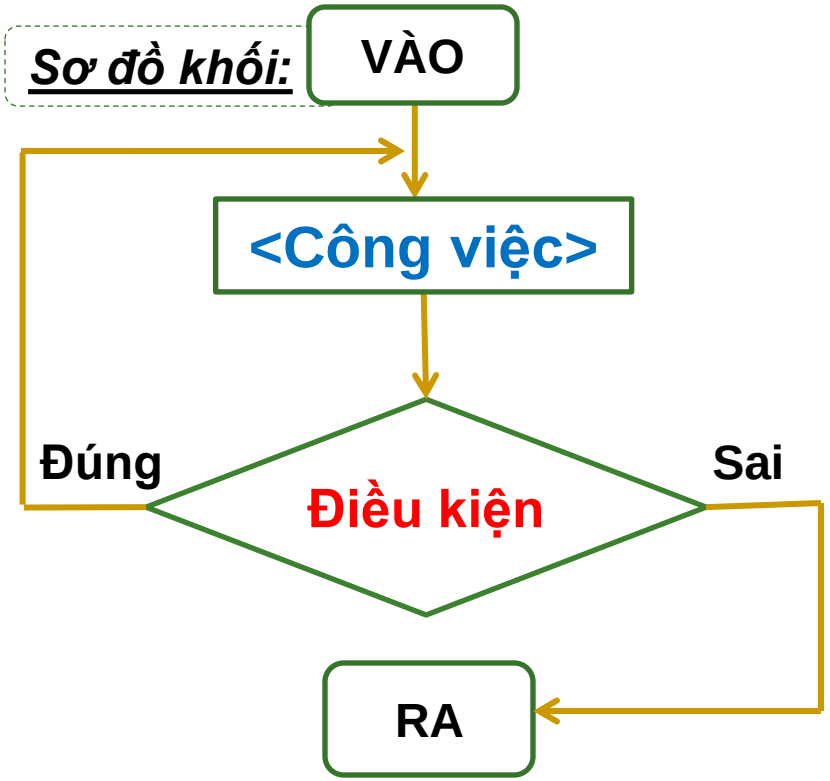
```
a. while (i++ < in)
      is = is + i;
b. while (is = is+i, i++ < in);
```

- **VD3:** `for(; ;){<Công việc>}`

  ⇔ `while(1){<Công việc>}`

Câu lênh do .. while

Sơ đồ khối:



Hoạt động:

- <Công việc> được thực hiện
-<Biểu thức> được kiểm tra:
  + Nếu Đúng:
    Lặp lại thực hiện <Công việc>
  + Nếu Sai:
    Kết thúc vòng lặp

Cú pháp:

```
do
    <Công việc>;
while (<Biểu thức>);
```

<Công việc>:
- Câu lệnh đơn
- Câu lệnh phức đặt trong cặp { và }

Câu lệnh do .. while

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

```
int i = 0;
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.
- Câu lệnh do… while có thể bị lặp vô hạn

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

Câu lệnh do .. while – Một số lưu ý

- VD: Nhập password = 123456

a.
```
do {
        printf("Nhap vao password: ");
        scanf("%d", &in);
    } while (in != 123456);
```

b.
```
do{}while(printf("Nhap vao password: "),
  scanf("%d", &in), in != 12345);
```

for, while, do ..while

- Đều có khả năng lặp lại nhiều hành động
- Vòng lặp for thường sử dụng khi biết được số lần lặp xác định.
- Vòng lặp thường while, do…while sử dụng khi không biết rõ số lần lặp.
- Khi gọi vòng lặp while, nếu biểu thức sai vòng lặp while sẽ không được thực hiện lần nào nhưng vòng lặp do…while thực hiện được 1 lần ➔ Số lần thực hiện ít nhất của while là 0 và của do…while là 1
- Các lệnh lặp for, while, do…while có thể lồng vào chính nó, hoặc lồng vào lẫn nhau. Nếu không cần thiết không nên lồng vào nhiều cấp dễ gây nhầm lẫn khi lập trình cũng như kiểm soát chương trình.

Bài tập

- 1. Nhập vào số là biển số xe (gồm 4 chữ số) của bạn. Cho biết số xe của bạn được mấy nút?
- 2. Viết chương trình in ra các số lẻ nhỏ hơn hoặc bằng n (Với n là số nguyên dương được nhập). Sao cho 15 số lẻ được in trên một dòng.
- 3. Viết chương trình in ra tất cả các ước của một số n (Với n được nhập từ bàn phím).
- 4. Nhập vào 1 số n, kiểm tra n có phải là số nguyên tố không?

- 5. Cho số nguyên dương n. Hãy tìm chữ số đầu tiên của n.
- 6. Tìm chữ số nhỏ nhất/ lớn nhất của của số nguyên dương n.

Hướng dẫn:5

```
do
        {
                printf("\nNhap n: ");
                scanf("%d", &n);
        }while(n < 0 && printf("\nLoi: (n >= 0)"));
        tam = n;
        do
        {
                i = tam % 10;
                tam /= 10;
        }while(tam != 0);
        printf("\nChu so dau tien la %d", i);
```

Hướng dẫn: 6

```
do   {
            printf("\nNhap n: ");
            scanf("%d", &n);
            }while(n < 0 && printf("\nLoi: n >= 0 !"));
            min = n % 10;  // khoi tao min
            if(n == 0)
                        min = 0;
            do
            {
                        i = n % 10;
              if(i < min)
              {
                           min = i;
              }
            }    while(n=n/10);
            printf("\nChu so nho nhat la %d", min);
```

Hỏi & Đáp

THANKS YOU