KỸ THUẬT LẬP TRÌNH C

Chương 4: Các câu lệnh lặp

1. Câu lệnh for
2. Câu lệnh while
3. Câu lệnh do …while
4. Câu lệnh break
5. Câu lệnh continue

1. Câu lệnh for

Cú pháp:

for (<BT1>**;** <BT2>**;** <BT3>**)**

<Lệnh>**;**

<Lệnh>: Lệnh đơn/phức.

Ý nghĩa:

Bước 1: Tính giá trị của biểu thức <BT1>.

Bước 2: Tính giá trị của biểu thức <BT2>.

Bước 3: Nếu giá trị của <BT2> là đúng thì máy sẽ thực hiện <Lệnh> và nhảy đến bước 4, ngược lại thì máy sẽ ra khỏi câu lệnh for.

Bước 4: Tính giá trị biểu thức <BT3> và quay về bước 2.

Lưu ý:

- Trong for có thể sẽ không có cả <BT1>, <BT2> và <BT3>. Khi không có <BT2> thì nó được xem là đúng.
- <BT1>, <BT2>, <BT3> có thể bao gồm nhiều biểu thức con phân cách nhau bởi dấu phẩy. Khi <BT2> gồm nhiều biểu thức thì tính đúng sai của nó được xem là tính đúng sai của biểu thức cuối cùng.

Ví dụ: Nhập vào n nguyên dương, hãy tính và in ra tổng: s = 1 + 2 + … + n

```
void main() {
    int i, n, s ;
    <Nhập n>
    if(n <= 0) <Dữ liệu nhập không hợp lệ>
    else {
            s = 0;
            for(i = 1; i <= n; i++)
                    s += i;
            <In s> }}
```

Chú ý: Đoạn mã trong phần else ở trên cũng có thể viết theo cách khác như sau:

```
for(s = 0, i = 1; i <= n; i++)
            s += i;
```

2. Câu lệnh while

Cú pháp:

while (<BT>)

<Lệnh>;

<Lệnh>: Lệnh đơn/phức.

Ý nghĩa:

Bước 1: Tính giá trị của biểu thức <BT>.

Bước 2: Nếu giá trị của <BT> là đúng thì máy sẽ thực hiện <Lệnh> và quay lại bước 1, ngược lại thì máy sẽ ra khỏi câu lệnh while.

Lưu ý:

<Lệnh> trong câu lệnh while có thể không thực hiện lần nào nếu như <BT> ngay từ đầu đã là sai.

Ví dụ: Làm lại ví dụ trên bằng lệnh while

```
void main() {
        <Nhập n>
        if(n <= 0)
                < Dữ liệu nhập không hợp lệ >
        else {
                s = 0; i = 1;
                while(i <= n) {
                        s += i;
                        i++;
                }
                <In s>
        }
}
```

3. Câu lệnh do … while

Cú pháp:

do <Lệnh>;

while (<BT>);

<Lệnh>: Lệnh đơn/phức.

Ý nghĩa:

Bước 1: Thực hiện <Lệnh> .

Bước 2: Tính giá trị của biểu thức <BT>. Nếu giá trị của <BT> là đúng thì máy sẽ quay lại bước 1, ngược lại thì máy sẽ ra khỏi câu lệnh do while.

Lưu ý:

<Lệnh> trong câu lệnh do… while sẽ được thực hiện ít nhất 1 lần do biểu thức <BT> được kiểm tra ở cuối.

Ví dụ: Làm lại ví dụ trên nhưng cho phép người sử dụng chạy CT nhiều lần, mỗi lần kết thúc tính toán thì nhắc người sử dụng "Tiếp tục hay ngừng CT?".

```
void main() {
    char traloi;
    do {
        <Nhap n>
        if(n <= 0)
            < DL nhập không hợp lệ >
```

```
            else {
                    s = 0; i =1;
                    while(i <= n)  {
                            s += i;
                            i++;
                    }
                    <In s>
            }
            printf("Tiep tuc hay ngung CT? (Y/N)")
            fflush(stdin);
            scanf("%c", &traloi);
    } while(traloi == 'Y' || traloi == 'y');
}
```

4. Câu lệnh break

Cú pháp:

break;

Ý nghĩa:

Khi gặp câu lệnh này trong các câu lệnh lặp máy sẽ ra khỏi các câu lệnh lặp.

Ví dụ: Làm lại ví dụ trên bằng câu lệnh while và break

```
void main() {
        <Nhap n>
        if(n <= 0)
                <Dữ liệu nhập không hợp lệ>
        else {
                s = 0; i = 1;
```

```
            while(1) { /*Biểu thức trong ngoặc luôn luôn
                                đúng*/
                    s += i;
                    i++;
                    if(i > n) break;
            }
            <In s>
    }
}
```

Lưu ý: Câu lệnh while(1) tương đương câu lệnh for(; ;)

5. Câu lệnh continue

Cú pháp:

continue;

Ý nghĩa:

Khi gặp câu lệnh này trong các câu lệnh lặp máy sẽ bỏ qua phần còn lại trong vòng lặp và tiếp tục thực hiện vòng lặp tiếp theo.

Ví dụ:

```
void main() {
        int n;
        for(; ;) {
                <Nhập n>
                if(n % 2 == 0) continue;
                else if(n % 3 == 0) break;
                printf("%d khong chia het cho 2 va 3\n", n);
        }
        printf("Da tim được mot so chia het cho  3, do la
                %d\n", n);
}
```

Hết