

PHỤ LỤC C

Câu lệnh rẽ nhánh và lặp trong C++

a) Câu lệnh rẽ nhánh

- *Dạng khuyết*

Cú pháp:

```
if (điều kiện) <câu lệnh>;
```

trong đó:

– *Điều kiện* là biểu thức nhận giá trị logic được đặt trong cặp ngoặc tròn (và);

– *Câu lệnh* là câu lệnh đơn hoặc ghép.

Khi *điều kiện* nhận giá trị đúng thì *câu lệnh* được thực hiện.

Ví dụ. Đoạn chương trình sau đây kiểm tra nếu giá trị biến x bằng 100 thì đưa ra thông báo "x bằng 100":

```
if (x == 100)
    cout << " x bằng 100 " ;
```

- *Dạng đủ*

Cú pháp:

```
if (điều kiện) <câu lệnh 1>; else <câu lệnh 2>;
```

Trong câu lệnh này, khi *điều kiện* nhận giá trị đúng thì *câu lệnh 1* được thực hiện, ngược lại thì *câu lệnh 2* được thực hiện.

Ví dụ. Đoạn chương trình sau đây kiểm tra nếu giá trị biến x bằng 100 thì đưa ra màn hình thông báo "x bằng 100", ngược lại thì thông báo "x không bằng 100":

```
if (x == 100)
    cout << " x bằng 100 " ;
else cout << " x không bằng 100 " ;
```

b) Câu lệnh lặp

- *Câu lệnh lặp while*

Cú pháp:

```
while (điều kiện) <câu lệnh>;
```

Trong câu lệnh này, *điều kiện* là biểu thức nhận giá trị logic. Trong khi *điều kiện* còn nhận giá trị đúng thì còn thực hiện *câu lệnh*.

Ví dụ. Nhập từ bàn phím một số tự nhiên n ($n < 20$), đưa ra màn hình các số nguyên dương giảm dần từ n đến 1.

```

#include <iostream.h>
int main()
{
    int n;
    cout << "Nhap so nguyen duong n (n<20): ";
    cin >> n;
    while (n>0)
        cout << n-- << ", ";
    return 0;
}

```

Trong đó `cout << n--` hiển thị giá trị hiện thời của n sau đó giảm giá trị n một đơn vị. Vòng lặp được kết thúc khi *điều kiện* nhận giá trị *sai*. Trong ví dụ trên, giá trị của n được giảm nhờ toán tử "--" nên vòng lặp được kết thúc khi $n = 0$.

- **Câu lệnh lặp do**

Cú pháp:

`do <câu lệnh> while (điều kiện);`

Sau khi thực hiện *câu lệnh*, vòng lặp sẽ kiểm tra *điều kiện*, nếu *điều kiện* còn nhận giá trị đúng thì *câu lệnh* còn được thực hiện. *Điều kiện* được tính toán sau khi *câu lệnh* được thực hiện, vì vậy *câu lệnh* sẽ được thực hiện ít nhất một lần.

Ví dụ. Đưa ra màn hình số nguyên nào nhập vào từ bàn phím cho đến khi số được nhập là 0.

```

#include <iostream.h>
int main()
{
    long n;
    do {
        cout << "Nhap vao mot so nguyen(nhap so 0 de ket thuc): ";
        cin >> n;
        cout << "So da nhap la: " << n << "\n";
    }
    while (n != 0);
    return 0;
}

```

- **Câu lệnh lặp for**

Cú pháp:

`for (khởi tạo; điều kiện; thay đổi) <câu lệnh>;`

Vòng lặp *for* thực hiện lặp lại *câu lệnh* chúng nào *điều kiện* còn nhận giá trị *đúng* như trong vòng lặp *while*. Nhưng trong *for* còn chứa khả năng *khởi tạo* và khả năng làm *thay đổi* giá trị các biến điều khiển vòng lặp.

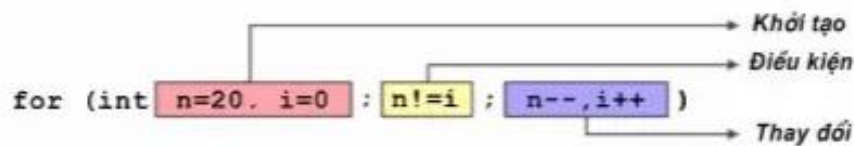
Hoạt động của vòng *for* như sau:

- Bước 1.* Đầu tiên *khởi tạo* được thực hiện thường để đặt một giá trị ban đầu cho các biến điều khiển, điều này chỉ được thực hiện một lần.
- Bước 2.* *Điều kiện* được kiểm tra, nếu nó nhận giá trị đúng thì *câu lệnh* được thực hiện, ngược lại thì tới bước 4.
- Bước 3.* *Thay đổi* được thực hiện để thay đổi giá trị biến điều khiển, quay về bước 2.
- Bước 4.* Kết thúc vòng lặp.

Ví dụ. Đưa ra màn hình các cặp số tự nhiên khác nhau có tổng bằng 20.

```
#include <iostream.h>
int main()
{
    for (int n=20, i=0; n!=i; n--, i++)
        cout << n << ", " << i << '\n';
    return 0;
}
```

Giải thích:



Ban đầu có cặp số ($n = 20; i = 0$) được đưa ra màn hình. Để tạo các cặp số khác, cho n giảm dần từng đơn vị, đồng thời i tăng dần từng đơn vị (để tổng của chúng vẫn bằng 20). Vòng lặp kết thúc khi gặp cặp số ($n = 10; i = 10$).