

TIN ĐẠI CƯƠNG

BÀI 3: VÒNG LẶP

Nhắc lại nội dung bài trước

- Cấu trúc của một chương trình đơn giản
- Các khái niệm cơ sở
 1. Biến (variable) và định danh (identifier)
 2. Biểu thức (expression)
 3. Phép gán
 4. Vài kiểu dữ liệu cơ bản: nguyên, thực, logic
- Nhập và xuất dữ liệu
- Phân rã bài toán (vấn đề)
- Hàm (function)

Nội dung chính

1. Đặt vấn đề
2. Cách giải quyết bằng vòng lặp
3. Ví dụ về vòng lặp dùng biến đếm
4. Ví dụ về vòng lặp dùng điều kiện
5. Biểu thức logic
6. Cú pháp các cấu trúc lặp
 1. Lặp **for**
 2. Lặp **while**
 3. Lặp **do-while**
7. Bài tập

Phần 1

Đặt vấn đề

Đặt vấn đề

- Ba cấu trúc điều khiển cơ bản trong máy tính
 - Tuần tự ← Đã học trong bài trước
 - Lặp ← Chương 3 (bài này)
 - Lựa chọn ← Chương 4 (bài sau)
- Nhiều hành vi, thuật toán trong cuộc sống về bản chất đã có tính lặp
 - Đếm số học sinh trong lớp
 - Tập luyện thể thao
 - Tính tổng dãy số
 - Các phương pháp tính xấp xỉ
 - Các phương pháp thử sai

Đặt vấn đề

- Một số bài toán giản đơn có thể giải quyết bằng phương pháp tuần tự, tuy nhiên có nhiều bất cập nếu chỉ dùng tuần tự
 - Chương trình dài, nhàm chán, dễ nhầm lẫn
 - Không thể tổng quát hóa
- **Ví dụ:** nhập và tính xem điểm trung bình của lớp K57-N13 môn Tin Đại Cương là bao nhiêu?
 - Khai báo 125 biến để lưu điểm của 125 sinh viên?
 - Viết 125 lệnh nhập dữ liệu?
 - Viết 125 lệnh cộng giá trị các biến với nhau?

Phần 2

Cách giải quyết bằng vòng lặp

Cách giải quyết bằng vòng lặp

- Hai kiểu lặp thông dụng trong cuộc sống
 - Lặp sử dụng điều kiện dừng
 - “Ăn cho đến khi no”
 - “Học cho đến khi thuộc”
 - Nhiều hành vi cuộc sống là lặp
 - Lặp sử dụng biến đếm
 - “Đếm số người trong một bàn tiệc”
 - “Chọn 10 bạn học giỏi nhất lớp”
 - Cũng một dạng điều kiện dừng đặc biệt
- Ngôn ngữ C/C++ có các lệnh cho phép dễ dàng thực hiện các kiểu lặp như trên

Ví dụ: tính tổng $1+2+3+4+5+\dots$

tong	tong = tong + dem	dem
0	tong = 0 + 1	1
1	tong = 1 + 2	2
3	tong = 3 + 3	3
6	tong = 6 + 4	4
10	tong = 10 + 5	5
15	tong = 15 + 6	6
...

Phần 3

Ví dụ về vòng lặp dùng biến đếm

Ví dụ 1

Yêu cầu: in ra màn hình các số từ 1 đến 100 mỗi số trên 1 dòng.

Cách làm: dùng số i làm biến đếm, cho i chạy từ 1 đến 100, mỗi lần chạy thì in i ra màn hình.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      for (int i = 1; i <= 100; i = i+1)
6          cout << i << endl;
7  }
```

Ví dụ 2

Yêu cầu (mở rộng của bài trước): in ra các số từ 1 đến n mỗi số trên 1 dòng.

Cách làm: nhập **n**, dùng **i** làm biến đếm, **i** chạy từ 1 đến **n**, mỗi lần chạy thì in **i** ra màn hình.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n;
6
7      cout << "N = "; cin >> n;
8
9      for (int i = 1; i <= n; i = i+1)
10         cout << i << endl;
11 }
```

Ví dụ 3

Yêu cầu: tính tổng các số từ 1 đến n

Cách làm: nhập n , cho biến i chạy từ 1 đến n , mỗi lần chạy cộng dồn i vào biến $tong$.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n, tong = 0;
6
7      cout << "N = "; cin >> n;
8
9      for (int i = 1; i <= n; i = i+1)
10         tong = tong + i;
11
12         cout << "Tong = " << tong << endl;
13     }
14 }
```

Ví dụ 4

Yêu cầu: nhập n và tính $n!$

Cách làm: nhập n cho biến i chạy từ 1 đến n , mỗi lần chạy nhân dồn i vào biến $tich$

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n, tich = 1;
6
7      cout << "N = "; cin >> n;
8
9      for (int i = 1; i <= n; i = i+1)
10         tich = tich * i;
11
12     cout << "n! = " << tich << endl;
13 }
```

Phần 4

Ví dụ về vòng lặp dùng điều kiện

Ví dụ: nhập liệu và tính tổng

```
#include <iostream>
using namespace std;
int main() {
    int tong = 0, n;
    do {
        cout << "Nhap mot so: "; cin >> n;
        tong = tong + n;
    } while (n != 0);
    cout << "Tong cac so vua nhap = " << tong;
}
```


Phần 5

Biểu thức logic

Biểu thức logic

- Các biểu thức logic là cơ sở để xây dựng điều kiện dừng lặp
- Giá trị logic có 2 loại: **false** (sai) và **true** (đúng)
 - Số nguyên có thể dùng lẫn lộn với kiểu logic, trong đó giá trị 0 tương đương với false và ngược lại
- Các phép toán logic:
 - Phép **NOT** (phép “đảo” - !)
 - Phép **AND** (phép “và” - &&)
 - Phép **OR** (phép “hoặc” - ||)
 - Phép **XOR** (phép “hoặc nghịch đảo” - ^)

Biểu thức logic

- Các phép so sánh: có kết quả kiểu logic
 - Bằng nhau: `==`
 - Khác nhau: `!=`
 - Lớn hơn: `>`
 - Lớn hơn hoặc bằng: `>=`
 - Nhỏ hơn: `<`
 - Nhỏ hơn hoặc bằng: `<=`
- Nên dùng cặp ngoặc để làm rõ thứ tự tính toán
 - `(a + 5 < 0) || (a >= b) && (a != c)`
 - `((a + 5) < 0) || ((a >= b) && (a != c))`

Phép toán AND

- Tiếng Anh: AND
- Tiếng Việt: VÀ
- Trong ngôn ngữ C/C++: &&

“chỉ đúng khi cả 2 vế đều đúng”

- Ví dụ:

$(a > b) \ \&\& \ (a > c)$

$((x \% 2) == 0) \ \&\& \ ((x \% 5) == 0)$

Phép toán OR

- Tiếng Anh: OR
- Tiếng Việt: HOẶC
- Trong ngôn ngữ C/C++: ||

“chỉ sai nếu cả 2 vế đều sai”

- Ví dụ:

$(a == 1) \ || \ (a == 3)$

$(a > (b+c)) \ || \ (b > (a+c)) \ || \ (c > (a+b))$

Phép toán XOR

- Tiếng Anh: XOR
- Tiếng Việt: HOẶC NGHỊCH ĐẢO
- Trong ngôn ngữ C/C++: ^

“sai nếu 2 vế có giá trị giống nhau”

- Ví dụ:
 $(a > 10) \wedge (b > 10)$
 $(a > b) \wedge (a \leq b)$

Bảng chân lý của các phép logic

x	y	x && y	x y	x ^ y
True	True	True	True	False
True	False	False	True	True
False	True	False	True	True
False	False	False	False	False

Phần 6

Cú pháp các cấu trúc lặp

Lặp FOR

- Cú pháp:

```
for (<khởi tạo>; <điều kiện>; <thay đổi>) {  
    <công việc>  
}
```

- Quá trình thực hiện:

1. Chạy khối <khởi tạo>

2. Kiểm tra <điều kiện>, nếu sai thì dừng lặp

3. Thực hiện <công việc>

4. Thực hiện <thay đổi>

5. Chuyển về bước 2

Lặp FOR

- Ví dụ: in các số từ 1 đến 20 ra màn hình

```
for (int x = 1; x <= 20; x = x + 1) {  
    cout << x << endl;  
}
```

- Quá trình thực hiện:

1. Khai báo biến `x` và gán giá trị `x = 1`

2. Kiểm tra `x <= 20` không? Nếu không thì dừng

3. In giá trị `x` ra màn hình

4. Tăng `x` lên 1

5. Chuyển về bước 2

Lặp FOR: vài ví dụ

// tính tổng các số từ 1 đến 100

```
for (int i = 1, tong = 0; i <= 100; i++) {  
    tong = tong + i;  
}
```

// tính tổng các số lẻ trong khoảng từ 1 đến n

```
for (int i = 1, tong = 0; i <= n; i = i + 2) {  
    tong = tong + i;  
}
```

// đoán thử xem vòng lặp này thực hiện điều gì

```
for (a = n - 1; 0 != (n % a); a = a - 1) {}
```

Lặp WHILE

- Cú pháp:

```
while (<điều kiện>) {  
    <công việc>  
}
```

- Quá trình thực hiện:

1. Kiểm tra <điều kiện>, nếu sai thì dừng lặp
2. Thực hiện <công việc>
3. Chuyển về bước 1

- Ví dụ:


```
a = n - 1;  
while (0 != (n % a)) { a = a - 1; }
```

Lặp DO-WHILE

- Cú pháp:

```
do {  
    <công việc>  
} while (<điều kiện>;
```

- Quá trình thực hiện:

1. Thực hiện <công việc>
 2. Kiểm tra <điều kiện>, nếu sai thì dừng lặp
 3. Chuyển về bước 1
- 

- Ví dụ:

```
a = n;  
do { a = a - 1; } while (0 == (n % a));
```

Phần 7

Bài tập

Bài tập

1. Nhập số n nguyên dương, tính và in các số chính phương từ 1^2 đến n^2 ra màn hình, mỗi số một dòng
2. Tính giá trị số A dưới đây
$$A = 1 \times 2 + 2 \times 3 + \dots + 99 \times 100$$
3. Nhập số n nguyên dương, tính giá trị số Y dưới đây
$$Y = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$
4. Nhập số n chẵn và tính giá trị của số Z sau đây
$$Z = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots - \frac{1}{n}$$