

Bài thực hành số 5 : Các câu lệnh lặp

I.ÔN TẬP KIẾN THỨC

1.Câu lệnh if:

<code>for (biểu thức 1 ; biểu thức 2 ; biểu thức 3)</code> <code> câu lệnh</code>	<code>for (biểu thức 1 ; biểu thức 2 ; biểu thức 3)</code> <code>{</code> <code> khối lệnh</code> <code>}</code>
---	--

2.Câu lệnh while:

<code>while (điều kiện)</code> <code> câu lệnh</code>	<code>while (điều kiện)</code> <code>{</code> <code> khối lệnh</code> <code>}</code>
---	--

3.Câu lệnh do while:

<code>do</code> <code> câu lệnh</code> <code>while (điều kiện) ;</code>	<code>do</code> <code> {</code> <code> khối lệnh</code> <code> }</code> <code>while (điều kiện) ;</code>
--	---

II.BÀI TẬP CÓ LỜI GIẢI

Bài 1 : Viết chương trình đếm và in ra số lượng các số nguyên chia hết cho 3 và cho 7 nằm trong đoạn từ 1 đến 100.

Lời giải:

```
#include <iostream>
using namespace std;

int main()
{
    int dem=0;
    for(int i=1; i<=100;i++)
        if(i%3==0 || i%7==0)
            dem++;
    cout<<"So cac so chia het cho 3 hoac 7 la:"<<dem;
```

```
    return 0;
}
```

Bài 2 : Lập chương trình xếp các dấu * thành tam giác cân n dòng với n nhập từ bàn phím

Lời giải:

```
#include <iostream>
using namespace std;

int main()
{
    int n;
    cout << "Nhap kích thước tam giác: "; cin >> n;
    cout << "\n\n";

    int a=(2*n-1);

    for (int i=1;i<=n;i++)
    {
        for (int j=1;j<=a;j++)
        {
            if(j>n-i && j<n+i)
                cout<<"*";
            else cout<<" ";
        }
        cout << "\n";
    }

    return 0;
}
```

Bài 3 : Tìm số n nguyên dương thỏa mãn điều kiện $33n^5 - 317 < 5$

Lời giải:

```
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    int n=1;
    while(3*pow(n,5) - 317<5)
        n++;
    cout<<"Số n="<<n;
    return 0;
}
```

```
}
```

III. THỰC HÀNH

Bài 1 : Xác định số lần lặp trong chương trình sau:

```
#include <iostream>
using namespace std;

int main()
{
    bool a = true, b = true;
    double x;
    while(a || b)
    {
        x= 2*x+1;
        if(x>0)
        {
            a= false;
            if(x>50)
                b= false;
        }
        cout<<"x = "<<x<<endl;
    }
    return 0;
}
```

Bài 2 : Cho biết các giá trị nào sẽ được hiện trên màn hình khi thực hiện chương trình sau:

```
#include <iostream>
using namespace std;

int main()
{
    int i=7;
    while(i>1)
    {
        if(i%2 !=0 )
            i=i*3+1;
        else
            i = i/2;
        cout<<i<<endl;
    }
    return 0;
}
```

Bài 3: Viết chương trình in ra màn hình các số nguyên liên tiếp từ M đến N ($M < N$) theo thứ tự tăng dần và giảm dần bằng các cách có thể.

Gợi ý: Dùng vòng lặp **while** và **for**

Bài 4: Viết chương trình tính tổng các số nguyên liên tiếp từ M đến N ($M < N$)

Bài 5: Viết chương trình tính tổng và giá trị trung bình của một dãy số nguyên dương nhập từ bàn phím cho tới khi gặp số 0 thì dừng lại.

Bài 6 : Viết chương trình cho phép người dùng nhập một số nguyên dương N và in ra màn hình các số chính phương trong N số này. Chương trình có kiểm tra nếu N âm thì yêu cầu nhập lại N cho đến khi hợp lệ.

Gợi ý : Số chính phương là bình phương của một số tự nhiên. Ví dụ số 4 là một số chính phương vì nó là bình phương của 2; các số chính phương trong 40 số tự nhiên đầu tiên là 1, 4, 9, 25 và 36.

Bài 7:

- Viết các chương trình tính các giai thừa của số tự nhiên N theo các công thức sau
 - Giai thừa của N: $N! = 1.2.3 \dots N$
 - Giai thừa lẻ của N: $(2N+1)!! = 1.3.5 \dots (2N+1)$
 - Giai thừa chẵn của N: $(2N)!! = 2.4.5 \dots (2N)$

Gợi ý : Dùng các vòng lặp while và for lồng nhau, chú ý đến miền tác động của biến nguyên.

Bài 8: Viết các chương trình tính và trả về giá trị của các biểu thức sau:

a. $S_1 = 1 - \frac{1}{2} + \frac{1}{3} - \dots + (-1)^{N-1} \frac{1}{N}$

b. $S_2 = 1 - \frac{1}{2!} + \frac{1}{3!} - \dots + (-1)^{N-1} \frac{1}{N!}$

Với N là một số nguyên dương.