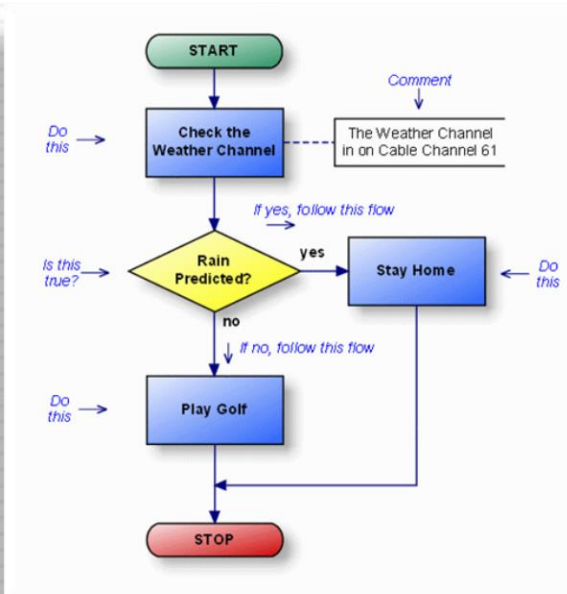


NHẬP MÔN LẬP TRÌNH

CHƯƠNG 3: CÂU LỆNH LẶP (LOOP STATEMENT)



GV: Phạm Nguyễn Sơn Tùng

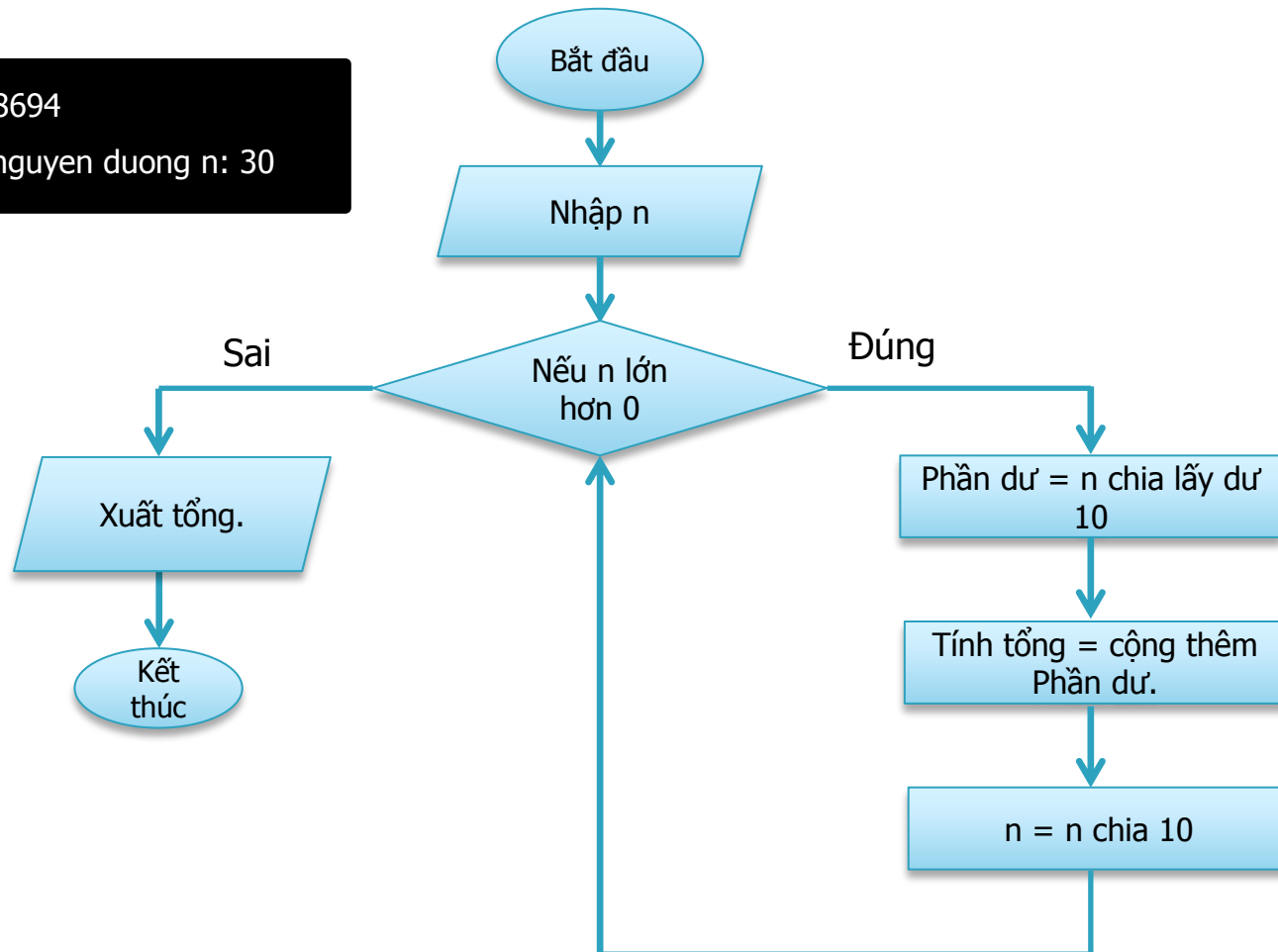
Email: pnstung@fit.hcmus.edu.vn

BÀI TOÁN DẪN NHẬP

Bài toán: Tính tổng các chữ số của số nguyên dương n .

Nhap so n : 38694

Tong chu so nguyen duong n : 30



CÚ PHÁP VÀ CÁCH SỬ DỤNG

Loại 1: Vòng lặp **for**.

```
for(<giá trị khởi đầu>; <điều kiện lặp>; <bước nhảy>)  
{  
    //Thực hiện các lệnh khi điều kiện lặp còn đúng  
}
```

Ghi chú:

- **<giá trị khởi đầu>**: đặt một biến có giá trị khởi đầu.
- **<điều kiện lặp>**: xét điều kiện nếu đúng thì tiếp tục lặp lại thực hiện các lệnh.
- **<bước nhảy>**: tăng giá trị của biến ban đầu lên theo một quy tắc nào đó.
- Nếu trong dấu ngoặc { } chỉ có 1 dòng lệnh thì có thể bỏ dấu ngoặc.
- Có thể xuất hiện nhiều vòng lặp for lồng vào nhau. Tạo nên nhiều vòng lặp khác nhau.

VÍ DỤ MINH HỌA

Ví dụ 1: Tính tổng giá trị các số từ 1 đến n, giá trị n do người dùng nhập vào.

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

int main()
{
    int n = 0;
    int tong = 0;
    cout << "Nhap gia tri n: ";
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        tong = tong + i;
    }
    cout << "Tong gia tri tu 1 den n la: " << tong;
    return 0;
}
```

MỘT SỐ LƯU Ý KHI SỬ DỤNG VÒNG LẶP FOR

Lưu ý 1: Vòng lặp for chúng ta có thể bỏ phần **<giá trị khởi đầu>**

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

int main()
{

}
```

MỘT SỐ LƯU Ý KHI SỬ DỤNG VÒNG LẶP FOR

Lưu ý 2: Vòng lặp for chúng ta có thể bỏ phần **<điều kiện lặp>**

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

int main()
{

}
}
```

break: Nếu chương trình gặp lệnh này sẽ thoát ra khỏi vòng lặp, đang chứa câu lệnh break gần nhất.

MỘT SỐ LƯU Ý KHI SỬ DỤNG VÒNG LẶP FOR


Lưu ý 3: Vòng lặp for chúng ta có thể bỏ phần **<bước nhảy>**

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

int main()
{

}

}
```



MỘT SỐ LƯU Ý KHI SỬ DỤNG VÒNG LẶP FOR

Lưu ý 4: Vòng lặp for chúng ta có thể bỏ phần <bước nhảy> một cách tùy biến. Ví dụ như bài chỉ kêu tính tổng các số lẻ.

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

int main()
{

}

}
```



continue: Nếu chương trình gặp lệnh này, sẽ bỏ bước lặp hiện tại và qua vòng lặp tiếp theo.

MỘT SỐ LƯU Ý KHI SỬ DỤNG VÒNG LẶP FOR

Lưu ý 5: Có nhiều <giá trị khởi đầu>; <điều kiện lặp>; <bước nhảy> bên trong vòng lặp for. Ví dụ in các cặp giá trị tăng dần của i và giảm dần của j.

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

int main()
{

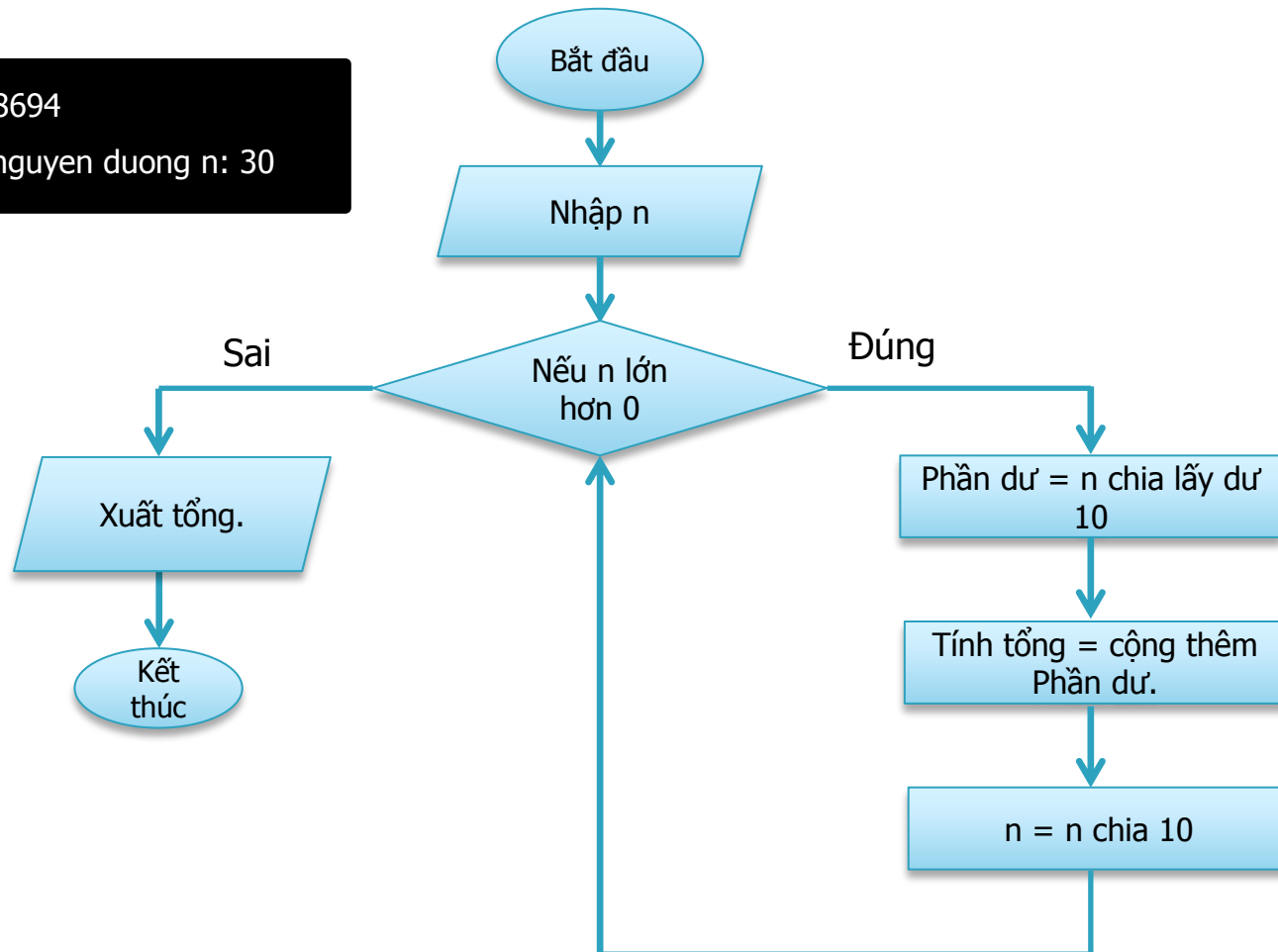
}
}
```

BÀI TOÁN DẪN NHẬP

Bài toán: Tính tổng các chữ số của số nguyên dương n .

Nhap so n : 38694

Tong chu so nguyen duong n : 30



MỘT SỐ LƯU Ý KHI SỬ DỤNG VÒNG LẶP FOR

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

int main()
{
    int n = 0;
    cout << "Nhap gia tri n: ";
    cin >> n;
    for (int i = 1, j = n; i <= n; i++, j--)
    {
        cout << i << "-" << j << endl;
    }
    return 0;
}
```

CÚ PHÁP VÀ CÁCH SỬ DỤNG

Loại 2: Vòng lặp **while**.

```
while(điều kiện lặp 1 && điều kiện lặp 2 || điều kiện lặp 3 ...)  
{  
    //Thực hiện các lệnh khi điều kiện lặp còn đúng  
}
```

Ghi chú:

- Khác với **for** vòng lặp while <giá trị khởi đầu> và <bước nhảy> có thể nằm trong hoặc ngoài vòng lặp.
- Không được thêm ; phía sau dấu ngoặc của while.

VÍ DỤ MINH HỌA

Ví dụ 1: Tính tổng giá trị các số từ 1 đến n, giá trị n do người dùng nhập vào.

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

int main()
{
    int n = 0;
    int tong = 0;
    cout << "Nhap gia tri n: ";
    cin >> n;
    int i = 1;
    while (i <= n)
    {
        tong = tong + i;
        i++;
    }
    cout << "Tong gia tri tu 1 den n la: " << tong;
    return 0;
}
```

MỘT SỐ LƯU Ý KHI SỬ DỤNG VÒNG LẶP WHILE

Lưu ý 1: Vòng lặp while chúng ta có thể bỏ phần <điều kiện lặp>

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

int main()
{

}
}
```

MỘT SỐ LƯU Ý KHI SỬ DỤNG VÒNG LẶP WHILE

Lưu ý 2: Vòng lặp while có thể lồng nhau hoặc lồng với vòng lặp khác

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

int main()
{

}
}
```

CÚ PHÁP VÀ CÁCH SỬ DỤNG

Loại 3: Vòng lặp **do - while**.

```
do
{
    //Thực hiện các lệnh theo yêu cầu và xét điều kiện sau
} while(điều kiện lặp 1 && điều kiện lặp 2 || điều kiện lặp 3 ...);
```

Ghi chú:

- Khác với vòng lặp while, vòng lặp do – while thực hiện 1 lệnh đầu tiên sau đó mới bắt đầu xét điều kiện.
- Khác với **for** vòng lặp do-while <giá trị khởi đầu> và <bước nhảy> có thể nằm trong hoặc ngoài vòng lặp.
- Phải có dấu **;** phía sau dấu ngoặc của while.

VÍ DỤ MINH HỌA

Ví dụ 1: Tính tổng giá trị các số từ 1 đến n, giá trị n do người dùng nhập vào.

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

int main()
{
    int n = 0;
    int tong = 0;
    cout << "Nhap gia tri n: ";
    cin >> n;
    int i = 0;
    do
    {
        tong = tong + i;
        i++;
    } while (i <= n);
    cout << "Tong gia tri tu 1 den n la: " << tong;
    return 0;
}
```


MỘT SỐ LƯU Ý KHI SỬ DỤNG VÒNG LẶP DO-WHILE

Lưu ý 1: Vòng lặp do-while chúng ta có thể bỏ phần **<điều kiện lặp>**

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

int main()
{

}
}
```



break: Nếu chương trình gặp lệnh này sẽ thoát ra khỏi vòng lặp, đang chứa câu lệnh break gần nhất.

MỘT SỐ LƯU Ý KHI SỬ DỤNG VÒNG LẶP DO - WHILE

Lưu ý 2: Vòng lặp while có thể lồng nhau hoặc lồng với vòng lặp khác

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

int main()
{

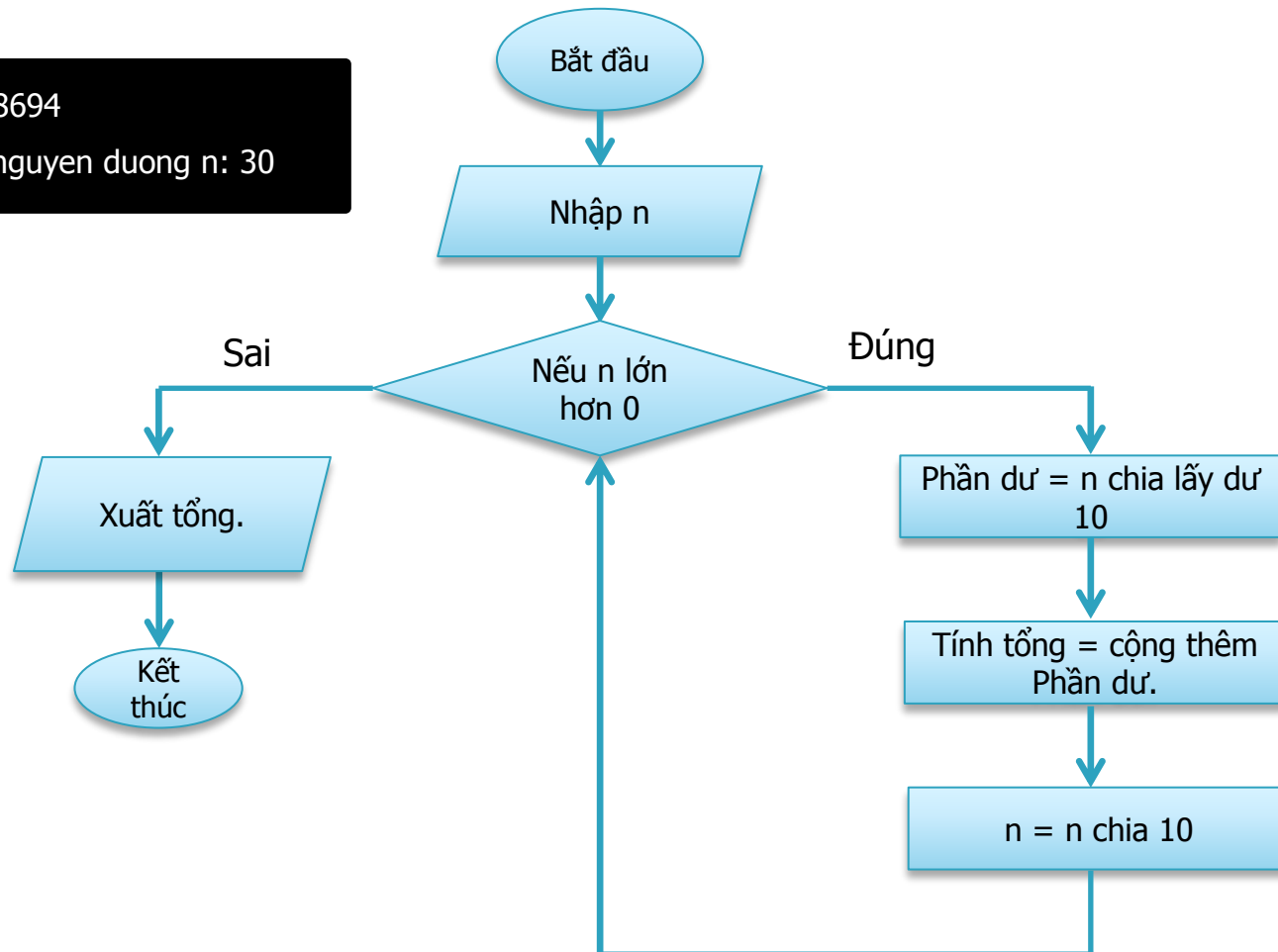
}
}
```

BÀI TOÁN DẪN NHẬP

Bài toán: Tính tổng các chữ số của số nguyên dương n .

Nhap so n : 38694

Tong chu so nguyen duong n : 30



BÀI TẬP ÁP DỤNG

Bài 1: Nhập một số nguyên dương n. Xuất ra số ngược lại. Ví dụ: Nhập 8709 → Xuất 9087.

Bài 2: Nhập vào số nguyên dương N. Tính tổng:

$$S = 1 + \frac{1}{1*2} + \frac{1}{2*3} + \dots + \frac{1}{(N-1)*N}$$

Bài 3: Tính tổng:

$$S = 1 + 1*2 + 1*2*3 + \dots + 1*2*3*N$$

Bài 4: Nhập vào số N và số K, tìm k nhỏ nhất sao cho:

$$1 + 2 + 3 + 4 + \dots + k > n$$

Bài 5: Nhập một số nguyên dương n. Kiểm tra n có toàn chẵn hoặc toàn lẻ hay không?.