

Test Ignore

Hồ Tuấn Thanh

htthanh@fit.hcmus.edu.vn

Stream

□ Xét đoạn code đọc 2 biến n và x như sau.

```
int main(){  
    int n;  
    double x;  
    cout << "Enter n: ";  
    cin >> n;  
    cout << "Enter x: ";  
    cin >> x;  
    return 0;  
}
```

Stream

- ❑ Giả sử dữ liệu người dùng nhập vào như sau.

10

3.14

- ❑ Như vậy, khi chạy chương trình:

- n nhận giá trị 10.
- x nhận giá trị 3.14.

Stream

- ❑ Hãy nghĩ đến con trỏ nhập liệu (cursor) trong Word.
- ❑ Đầu tiên, con trỏ nằm ở trước số 10.

```
10  
3.14
```

- ❑ Sau lệnh đọc `cin >> n;` con trỏ di chuyển về sau số 10 (và n đã nhận giá trị 10).

```
10  
3.14
```

- ❑ Sau lệnh đọc `cin >> x;` con trỏ di chuyển về sau số 3.14 (và x đã nhận giá trị 3.14).

```
10  
3.14
```

Stream

- ❑ Cứ tiếp tục như vậy, con trỏ di chuyển đến khi nào toàn bộ nội dung input đọc xong.
- ❑ Điều này tương tự như đọc file, con trỏ `FILE *f;` hoặc `ifstream f;` cũng thực hiện quá trình tương tự.

Vấn đề khi đọc chuỗi

- ❑ Khi đọc 1 một số (int, long, float, double) rồi đọc một chuỗi, ta thường gặp lỗi “trôi” không dừng lại để người dùng nhập chuỗi.

Chạy chương trình

- ❑ Phần đọc id (số) thì dừng lại đợi người dùng nhập.
- ❑ Phần đọc name (chuỗi) thì không dừng lại đợi người dùng nhập (dân gian gọi là trôi).
 - Lúc in ra, name là chuỗi rỗng (empty string)

Enter id: 19127001

Enter name: STUDENT 19127001 -

Giải thích

- ❑ Quay lại với con trỏ nhập liệu lúc này.
- ❑ Lúc đầu, con trỏ nằm trước số 19127001.

19127001
Ho Tuan Thanh

- ❑ Sau lệnh đọc `cin >> id;` con trỏ di chuyển về sau số 19127001 (và id đã nhận giá trị 19127001).

19127001
Ho Tuan Thanh

Giải thích

- ❑ Lệnh đọc chuỗi `cin.getline(name, 51);` sẽ đọc đến hết dòng và chuyển đến dòng tiếp theo (và name nhận giá trị chuỗi rỗng).

19127001

Ho Tuan Thanh

Khắc phục

- ❑ Như vậy, để khắc phục lỗi này, ta có thể đọc chuỗi name 2 lần, để lần đầu name là chuỗi rỗng, lần sau name sẽ nhận giá trị “Ho Tuan Thanh”.

```
int main(){
    int id;
    char name[51];
    cout << "Enter id: ";
    cin >> id;
    cout << "Enter name: ";
    cin.getline(name, 51);
    cin.getline(name, 51);

    cout << "STUDENT " << id << " - " << name << endl;
    return 0;
}
```

Khắc phục

19127001
Ho Tuan Thanh

19127001
Ho Tuan Thanh

19127001
Ho Tuan Thanh

19127001
Ho Tuan Thanh

Khắc phục

- ❑ Hoặc thông thường, ta dùng lệnh `cin.ignore(1);` để đọc bỏ 1 kí tự (cho con trỏ nhập liệu xuống dòng 2).

```
int main(){
    int id;
    char name[51];

    cout << "Enter id: ";
    cin >> id;
    cin.ignore(1);
    cout << "Enter name: ";
    cin.getline(name, 51);

    cout << "STUDENT " << id << " - " << name << endl;
    return 0;
}
```

Khắc phục (C)

□ Dùng hàm `getc(stdin)` để đọc bỏ 1 kí tự.

```
int main(){
    int id;
    char name[51];

    printf("Enter id: ");
    scanf("%d", &id);
    getc(stdin);
    printf("Enter name: ");
    fgets(name, 51, stdin);
    name[strlen(name) - 1] = '\\0';

    printf("STUDENT %d - %s\\n", id, name);
    return 0;
}
```

Chỉ “trôi” giữa số và chuỗi

- ❑ Vấn đề trôi chỉ gặp phải khi bạn đọc 1 số và sau đó đọc 1 chuỗi.
- ❑ Do đó, nếu áp dụng lệnh đọc bỏ 1 kí tự, giữa 2 chuỗi, sẽ làm mất kí tự đầu tiên của chuỗi thứ 2.

Chỉ “trôi” giữa số và chuỗi

```
int main(){
    char name[51];
    char address[101];
    cout << "Enter name: ";
    cin.getline(name, 51);
    cin.ignore(1);
    cout << "Enter address: ";
    cin.getline(address, 101);
    cout << "STUDENT " << name << " - " << address << endl;
    return 0;
}
```

Enter name: Thanh Ho

Quan 5

Enter address: STUDENT Thanh Ho - **uan 5**

Chỉ “trôi” giữa số và chuỗi

❑ Nhớ đừng dùng `cin.ignore(1);` giữa 2 chuỗi nha!!!

```
int main(){
    char name[51];
    char address[101];
    cout << "Enter name: ";
    cin.getline(name, 51);
    cout << "Enter address: ";
    cin.getline(address, 101);
    cout << "STUDENT " << name << " - " << address << endl;
    return 0;
}
```

Enter name: Thanh Ho

Enter address: Quan 5

STUDENT Thanh Ho – Quan 5

Không dùng cin.ignore(1);

```
int main(){
    int n;
    char name[51];
    char address[101];
    cout << "Enter n: ";
    cin >> n;
    for(int i = 0; i < n; i++){
        cout << "Enter name: ";
        cin.getline(name, 51);
        cout << "Enter address: ";
        cin.getline(address, 101);

        cout << "STUDENT " << i << ": ";
        cout << name << " - " << address << endl;
    }
    return 0;
}
```

Enter n: 2

Enter name: Enter address: Thanh Ho

STUDENT 0: - Thanh Ho

Enter name: Quan 5

Enter address: Dung Nguyen

STUDENT 1: Quan 5 - Dung Nguyen

Program ended with exit code: 0

Dùng cin.ignore(1); sai vị trí

```
int main(){
    int n;
    char name[51];
    char address[101];
    cout << "Enter n: ";
    cin >> n;
    for(int i = 0; i < n; i++){
        cin.ignore(1);
        cout << "Enter name: ";
        cin.getline(name, 51);
        cout << "Enter address: ";
        cin.getline(address, 101);

        cout << "STUDENT " << i << ": ";
        cout << name << " - " << address << endl;
    }
    return 0;
}
```

```
Enter n: 2
Enter name: Thanh Ho
Enter address: Quan 5
STUDENT 0: Thanh Ho - Quan 5
Dung Nguyen
Enter name: Enter address: Quan 10
STUDENT 1: ung Nguyen - Quan 10
```

Dùng `cin.ignore(1);` đúng vị trí

```
int main(){
    int n;
    char name[51];
    char address[101];
    cout << "Enter n: ";
    cin >> n;
    cin.ignore(1);
    for(int i = 0; i < n; i++){
        cout << "Enter name: ";
        cin.getline(name, 51);
        cout << "Enter address: ";
        cin.getline(address, 101);

        cout << "STUDENT " << i << ": ";
        cout << name << " - " << address << endl;
    }
    return 0;
}
```

```
Enter n: 2
Enter name: Thanh Ho
Enter address: Quan 5
STUDENT 0: Thanh Ho - Quan 5
Enter name: Dung Nguyen
Enter address: Quan 10
STUDENT 1: Dung Nguyen - Quan 10
```

