NGÔN NGỮ LẬP TRÌNH C++

Silde 2.3: Char & String

Giảng viên: Th.S Bùi Văn Kiên

1. Xâu kí tự trong C++

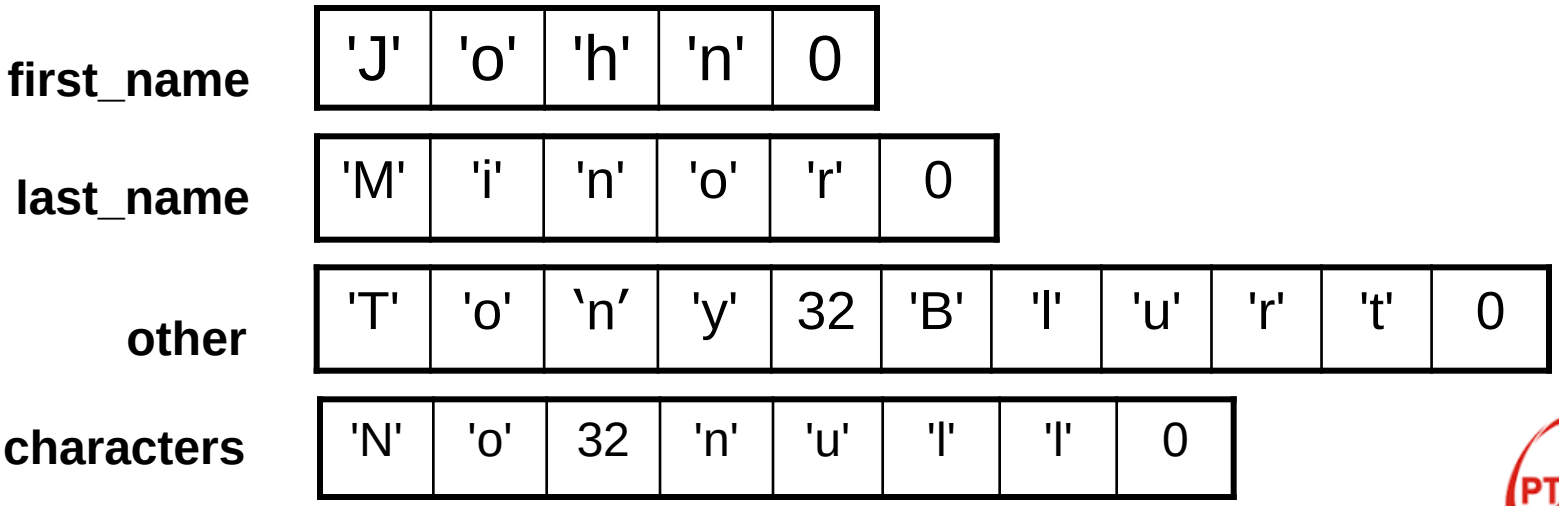
Mảng các kí tự:

- Nguyên bản từ ngôn ngữ C
- Một xâu có thể được coi là một mảng một chiều, trong đó mỗi phần tử là một ký tự. Ký tự kết thúc là ‘\0’
  - char s1[200];
  - cin >> s1; nhập xâu thông thường
  - cin.getline(s1,200): nhập nguyên 1 dòng. Đọc cả chuỗi chứa khoảng trắng cho đến khi gặp ký tự phân cách (mặc định là '\n')
  - strcpy(s1, s2): copy xâu s2 vào s1
  - strcat(s1, s2): nối xâu s2 vào sau xâu s1
  - strlen(s) : tính độ dài xâu s
  - strcmp(s1, s2): so sánh hai xâu s1 và s2 theo thứ tự từ điển.

1. Xâu kí tự trong C++

Mảng các kí tự:

```
char  first_name[5] = { 'J', 'o', 'h', 'n', '\0' };
char  last_name[6]  = "Minor";
char  other[]       = "Tony Blurt";
char  characters[8] = "No null";
```



1. Xâu kí tự trong C++

Chuyển đổi giữa số và kí tự dựa trên mã ASCII

- Khi đọc bằng cin, xâu sẽ ngắt khi gặp kí tự ‘space’ (dấu cách) đầu tiên.
- Input: Nguyen Van An
- Output: Nguyen

- Xử lí: sử dụng cin.getline()

1. Xâu kí tự trong C++

Các lỗi thường gặp:

- Khi đọc bằng cin, xâu sẽ ngắt khi gặp kí tự ‘space’ (dấu cách) đầu tiên.
- Input: Nguyen Van An
- Output: Nguyen

- Xử lí: sử dụng cin.getline()

```
#include <iostream>
using namespace std;
int main(){
        char szFullName[10];
        cin >> szFullName;
        cout << szFullName << endl;

        return 0;
}
```

1. Xâu kí tự trong C++

Các lỗi thường gặp:

- Lỗi tràn mảng
- Input: NguyenVanAn (Có 11 kí tự, + NULL → 12 kí tự)
- Chương trình bị lỗi sau khi nhập input

- Xử lí: tăng kích thước mảng

```
#include <iostream>
using namespace std;
int main(){
        char szFullName[10];
        cin >> szFullName;
        cout << szFullName << endl;

        return 0;
}
```

2. std::string

Kiểu dữ liệu string

Trong C++ có kiểu dữ liệu dành riêng cho xử lý chuỗi là string

- s1 = s2 sao chép xâu s2 vào s1
- s1 = s1 + s2 Nối s2 vào sau s1
- s1 == s2 So sánh xâu s1 có trùng xâu s2 hay không
- s1 < s2 So sánh xâu s1 có đứng trước s2 hay không (từ điển)
- s1 > s2 So sánh xâu s1 có đứng sau s2 hay không (từ điển)
- s1 <= s2 So sánh xâu s1 có đứng trước hoặc trùng s2 (từ điển)
- s1 >= s2 So sánh xâu s1 có đứng sau hoặc trùng s2 (từ điển)
- s1 != s2 So sánh xâu khác nhau hay không

Độ phức tạp thuật toán – Big Oh

- Ký hiệu O (Big-Oh), dung để xác định độ phức tạp thuật toán. Ví dụ:
  - 7n-2 là O(n)
  - $3n^3 + 20n^2 + 10$ là $O(n^3)$
  - 3 log n + 5 là O(log n)

Độ phức tạp thuật toán – Big Oh

Gọi $T_1(n)$ và $T_2(n)$ là thời gian thực hiện của hai giai đoạn chương trình P_1 và P_2 mà $T_1(n) = O(f(n))$; $T_2(n) = O(g(n))$

- **Quy tắc cộng:**

Thời gian thực hiện đoạn P_1 rồi P_2 tiếp theo là:

$$T1(n) + T2(n) = O(max(f(n), g(n))).$$

- **Quy tắc nhân:**

Thời gian thực hiện P_1 và P_2 lồng nhau là:

$$T_1(n)T_2(n) = O(f(n)*g(n))$$

Độ phức tạp thuật toán – Ví dụ

- Các phép gán, đọc, viết, so sánh, if else, switch, return là các phép toán sơ cấp

→ có thời gian thực hiện là: O(1)

- Các vòng lặp for, while, do while, nếu duyệt hết n phần tử thì độ phức tạp là O(n)

Độ phức tạp thuật toán – Ví dụ

Case1:
```
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        sum += i*j;
```
$O(n^2)$

Case 2:
```
for (i=0; i<n; i++)
    for (j=0; i<i; j++)
        for (k=0; k<j; k++)
            ans++;
```
$O(n^3)$

Case 3:
```
for (int i=0; i<n-1; i++)
    for (int j=0; j<i; j++)
        ans += 1;
```
$O(n^2)$

2. std::string

Kiểu dữ liệu string: Cẩn thận với độ phức tạp của các hàm.

STT	Tên hàm (Phương thức)	Ý nghĩa
1	s.size() hoặc s.length()	Trả lại độ dài string s → O(1)
2	getline(cin, s)	Nhập một dòng từ bàn phím cho string s
3	s.erase(n, k)	Xóa k ký tự trong s kể từ vị trí thứ n
4	s.insert(n, s1)	Chèn s1 vào s kể từ vị trí thứ n.
5	s.insert(n, s1, k, m)	Chèn m ký tự kể từ ký tự thứ k trong s1 vào s kể từ vị trí thứ n.
6	s.replace(n, k, s1)	Thay thế k ký tự trong s kể từ vị trí thứ n bằng xâu s1.
7	s.find(s1)	Trả lại vị trí xuất hiện đầu tiên của s1 trong s → O(n)
8	s.rfind(s1)	Trả lại vị trí xuất hiện cuối cùng của s1 trong s → O(n)
9	s.at(int i) hoặc s[i]	Truy nhập đến phần tử thứ i trong string → O(1)

2. std::string

Duyệt string

- for (int i = 0; i < (int)s.size(); i++)

Nhập liệu kết hợp

- Khi kết hợp nhập liệu các dạng dữ liệu khác nhau, có thể xảy ra một số ký tự bị bỏ qua không mong muốn.
- Có thể sử dụng cin.ignore() để khắc phục

2. std::string

Sử dụng stringstream

- C++ cho phép một số stream sử dụng cho các đối tượng std::string, có thể sử dụng các toán tử (<<) và (>>) để làm việc với std::string. String stream cung cấp một buffer để chứa dữ liệu. Tuy nhiên, string stream không kết nối đến các Input/Output channel (như keyboard, monitor, file,…)
- Thư viện:

#include <sstream>

- Ví dụ:

```
stringstream ss;
ss << "Hello" << " " << "PTIT" << endl;
cout << ss.str();
```

2. std::string

Sử dụng stringstream

- Một số phương pháp đưa dữ liệu vào:

```
stringstream ss;
ss.str("Hello PTIT\n");
```

- Hoặc:

```
stringstream ss("Hello PTIT\n“);
```

- Tách từ:

```
string token = "";
while (ss >> token) {
        cout << token << endl;
}
```

3. Bài tập

- Các bài toán chuẩn hoá / xử lý xâu
- Viết chương trình chuẩn hóa tên
- Viết chương trình loại bỏ xâu con
- Các bài toán xử lý số nguyên lớn: iết chương trình tính tổng/hiểu/tích hai số nguyên lớn
- Các bài tập về số đẹp

3. Bài tập

1. Viết chương trình chuẩn hóa tên

Input là một string có thể có nhiều dấu cách, các kí tự đầu tiên của một từ có thể chưa được viết hoa.

→ Cần chuẩn hóa: các từ cách nhau một dấu cách, chữ cái đầu tiên của mỗi từ đều được viết hoa.

- Input:

Nguyen van an

- Output

Nguyen Van An

3. Bài tập

2. Tìm kiếm số 42

Viết liền các số từ 1→N để tạo thành một xâu S cực kì dài.

12345678910111213……..

Nhiệm vụ của bạn là hãy tìm các vị trí xuất hiện xâu con “42”. Lưu ý, chỉ số bắt đầu của xâu S được tính từ 0.

- Input:

Một số nguyên N (N <= 100 000)

- Output

Các vị trí xâu con “42” xuất hiện.

N = 100
Output:
38 73

3. Bài tập

Toán tử s = s + t và s += t trong string

- s = s + t

Tạo ra 1 copy của s (u), u.append(t) rồi gán lại s = u. Độ phức tạp: O(n)

- s += t

Gọi đến hàm inline s.append(t). Độ phức tạp O(1)

QUESTIONS & ANSWERS