

NHẬP MÔN LẬP TRÌNH (23CLC05)

Buổi 8 – 20/11/23

Chuỗi có quan trọng không?

- Chuỗi là phương tiện nhập xuất cơ bản của chương trình
- Chương trình nhận dữ liệu từ người dùng qua chuỗi nhập từ bàn phím (sau đó được chuyển đổi sang các kiểu dữ liệu khác như số nguyên, số thực, ...)
- Chương trình xuất dữ liệu ra màn hình ở dạng chuỗi (đã được chuyển đổi từ các kiểu dữ liệu khác)
- Chuỗi là dạng dữ liệu rất quan trọng, thậm chí, ở góc độ nào đó, còn quan trọng và phổ biến hơn số

Xử lý chuỗi bằng cách nào?

- Xử lý chuỗi tương tự như mảng một chiều với lưu ý
 - Mảng một chiều lưu trữ tương minh chiều dài mảng
 - Chuỗi dùng kí tự NULL (mã 0) đánh dấu kết thúc chuỗi
- Thư viện chuẩn C cung cấp rất nhiều hàm thao tác, xử lý kí tự, chuỗi trong stdio.h, string.h, ctype.h, stdlib.h
 - <https://www.cplusplus.com/reference/cstdio/>
 - <https://www.cplusplus.com/reference/cstring/>
 - <https://www.cplusplus.com/reference/ctype/>
 - <https://www.cplusplus.com/reference/stdlib/>
- Dạng chuỗi này còn được gọi là **chuỗi-C** (C-string)

C++ có hỗ trợ gì thêm giúp làm việc với chuỗi?

- Thư viện chuẩn C++ cung cấp một kiểu dữ liệu để lưu trữ và thao tác với chuỗi là **lớp string** (string class)
- Kiểu này cùng với các hàm thao tác trên nó được để trong không gian tên std và được `#include` từ file tiêu đề `string`
 - <https://www.cplusplus.com/reference/string/string/>
- Dạng chuỗi này còn được gọi là **chuỗi-C++** (C++ - string)
- (Xem Thầy minh họa kiểu string)

Cộng điểm

- Làm bài tập 4.4.5 (Tài liệu C)

Lỗi tràn bộ đệm là gì?

- **Truy cập quá chỉ số mảng** (array index out of bounds) là việc dùng chỉ số mảng không hợp lệ để truy cập vào phần tử không nằm trong mảng
- C/C++ không hỗ trợ tự động việc kiểm tra truy cập quá chỉ số mảng mà lập trình viên phải tự quản lý lấy (Python có hỗ trợ). Truy cập quá chỉ số mảng là lỗi rất nguy hiểm cần để ý và tránh!
- **Buffer overflow** là lỗi ghi dữ liệu ra ngoài vùng nhớ thao tác (thường là vùng nhớ được cấp phát của một mảng hay chuỗi)
- https://en.wikipedia.org/wiki/Buffer_overflow
- (Xem Thầy minh họa)

Giải lao!!!

- Giải lao đến 9h35

Luyện tập về nhà!

- Tìm hiểu **bài toán chuỗi con chung dài nhất** (longest common substring problem) và viết hàm giải bài toán này
- Tìm cách làm “hiệu quả” nhất!

Ôn tập các kiểu dữ liệu

- **Kiểu dữ liệu** (data type) là nhóm các dữ liệu giống nhau về ý nghĩa, mục đích sử dụng và dạng thức
- Các dữ liệu của cùng một kiểu được tổ chức và xử lý giống nhau trên máy và trong chương trình
- Một kiểu dữ liệu được xác định bởi:
 - tập các dữ liệu cụ thể có thể có của kiểu
 - tập các thao tác xử lý có thể dùng trên dữ liệu của kiểu
 - cách định dạng và lưu trữ/truyền nhận dữ liệu của kiểu trên máy

Ôn tập các kiểu dữ liệu

- Các kiểu dữ liệu cơ bản (đã học):
 - số nguyên (integer)
 - số thực (real)
 - luận lý (boolean)
 - kí tự (character)
- Các kiểu dữ liệu nâng cao (đã học):
 - mảng (array)
 - chuỗi (string)
 - C++: vector, string

Ôn tập các kiểu dữ liệu

- Đọc thật kĩ **Bài 3.5** (Tài liệu C) và tài liệu tóm lược

Viết chương trình xác định số lớn nhất

- Viết chương trình cho nhập n **số nguyên** và xác định số lớn nhất
- **Viết dễ!**
- Viết chương trình cho nhập n **số thực** và xác định số lớn nhất
- **Viết dễ!**
- Viết chương trình cho nhập n **phân số** và xác định số lớn nhất
- **Viết khó?!**
- (Xem chương trình Thầy viết)

Kiểu dữ liệu phức hợp là gì?

- Các kiểu dữ liệu đơn giản (số nguyên, số thực, luận lý, kí tự) giúp ta xử lý dữ liệu đơn lẻ
- Mảng và chuỗi giúp ta xử lý tập nhiều dữ liệu cùng kiểu
- Các kiểu dữ liệu phức hợp giúp ta xử lý các dữ liệu gồm nhiều phần có thể không cùng kiểu

Kiểu dữ liệu phức hợp là gì?

- **Kiểu dữ liệu phức hợp** (composite data type) là kiểu dữ liệu được xây dựng từ các kiểu dữ liệu khác
- Đây cũng là những **kiểu do người dùng** (lập trình viên) **định nghĩa** (user-defined data type)
- Hai kiểu dữ liệu phức hợp quan trọng là:
 - **kiểu cấu trúc** (struct)
 - **kiểu lớp** (class) (kiểu lớp sẽ được học kĩ trong môn khác)

Kiểu cấu trúc là gì?

- **Kiểu cấu trúc** (struct) là kiểu dữ liệu gồm nhiều **thành phần** (component, member):
 - các thành phần có thể khác kiểu nhau
 - các thành phần có thể có kiểu là kiểu cấu trúc khác
- Kiểu cấu trúc còn được gọi là **bản ghi** (record) với các thành phần thường được gọi là **trường** (field)

Minh họa kiểu cấu trúc trên bộ nhớ

```
struct PayRoll
{
    int empNumber;
    string name;
    double hours, payRate, grossPay;
};

PayRoll deptHead, foreman, associate;
```

Members

Structure variable name

deptHead

empNumber
name
hours
payRate
grossPay

| | |
|-----------|----------------------|
| empNumber | <input type="text"/> |
| name | <input type="text"/> |
| hours | <input type="text"/> |
| payRate | <input type="text"/> |
| grossPay | <input type="text"/> |

foreman

empNumber
name
hours
payRate
grossPay

| | |
|-----------|----------------------|
| empNumber | <input type="text"/> |
| name | <input type="text"/> |
| hours | <input type="text"/> |
| payRate | <input type="text"/> |
| grossPay | <input type="text"/> |

associate

empNumber
name
hours
payRate
grossPay

| | |
|-----------|----------------------|
| empNumber | <input type="text"/> |
| name | <input type="text"/> |
| hours | <input type="text"/> |
| payRate | <input type="text"/> |
| grossPay | <input type="text"/> |

Làm thế nào định nghĩa kiểu cấu trúc?

- **Định nghĩa kiểu cấu trúc** là việc:
 - khai báo kiểu (bao gồm khai báo các thành phần)
 - khai báo các thao tác trên kiểu
 - và cài đặt các thao tác
- Nên tổ chức module cho các kiểu cấu trúc phức tạp với việc tách riêng phần khai báo kiểu/thao tác ra khỏi phần cài đặt

Làm thế nào truy cập các thành phần của cấu trúc?

- **Toán tử truy cập thành phần** (toán tử `.`) (member access operator) giúp truy cập đến thành phần tương ứng của một dữ liệu kiểu cấu trúc
- Thành phần được truy cập có vai trò như một **l-value** (giống như biến thông thường)

Có thể “lồng ghép” các “dạng” dữ liệu với nhau không?

- Để xử lý nhiều dữ liệu cấu trúc cùng kiểu, ta có thể dùng mảng cấu trúc
- Cấu trúc cũng có thể có các thành phần là mảng, chuỗi
- **Cấu trúc lồng** (nested struct) là các cấu trúc có chứa thành phần là cấu trúc khác
- Cấu trúc lồng thường được dùng để tổ chức và xử lý các dạng dữ liệu “lồng chứa” nhiều mức

Cộng điểm

- Thêm thao tác cộng vào kiểu PhanSo

Viết chương trình xác định số lớn nhất

- Viết chương trình cho nhập n **phân số** và xác định số lớn nhất
- **Viết đẹp: dùng nạp chồng toán tử (operator overloading) của C++**
- (Xem chương trình Thầy viết)

Cộng điểm

- Nạp chồng toán tử + cho kiểu PhanSo