

NHẬP MÔN LẬP TRÌNH (23CLC05)

Buổi 6 – 06/11/23

Tham số có nhất thiết cần đối số?

Python	C++
<pre>def power(x=1, n=0): y = 1 for i in range(n): y *= x return y print(power(2, 64)) print(power(2, 0)) print(power(2)) print(power())</pre>	<pre>#include <iostream> using namespace std; double power(double x = 1, int n = 0) { double y = 1; for (int i = 1; i <= n; i++) y *= x; return y; } int main() { cout << power(2, 64) << endl; cout << power(2, 0) << endl; cout << power(2) << endl; cout << power() << endl; }</pre>

Đối số mặc định là gì?

- Các tham số có thể được khai báo nhận **đối số mặc định** (default argument) là giá trị mà tham số sẽ nhận nếu lời gọi hàm không cung cấp đối số tương ứng
- Các tham số này còn được gọi là **tham số tùy chọn** (optional parameter)
- Python, C++ có hỗ trợ đối số mặc định; C thì không
- “Để an toàn”, các đối số mặc định nên là các “biểu thức hằng”
- Chỉ nên dùng đối số mặc định khi nó có ý nghĩa (chẳng hạn có một giá trị rất hay dùng hoặc ngầm định dùng cho tham số)

Các đối số có nhất thiết được truyền theo đúng thứ tự các tham số?

Python

```
for power(x=1, n=0):  
    y = 1  
    for i in range(n):  
        y *= x  
    return y  
  
print("2^64 = ", power(2, 64))  
print("64^2 = ", power(64, 2))  
print("2^64 = ", power(n=64, x=2))  
print("2^64 = ", power(x=2, n=64))  
print("2^64 = ", power(2, n=64))  
print("64^0 = ", power(64))  
print("1^64 = ", power(n=64))  
# print("2^64 = ", power(x = 2, 64)) # Syntax error
```

Truyền đối số theo tên là gì?

- Có 2 cách cơ bản để **truyền (gán) giá trị của đối số cho tham số** (argument passing) là
 - (1) **theo vị trí** (passing argument by position), dựa trên thứ tự của đối số tương ứng với thứ tự của tham số
 - (2) **theo tên** (passing argument by name), dựa trên tên của tham số đi kèm với đối sốCách đầu tiện lợi hơn nhưng cách sau rõ ràng hơn
- Hầu hết các ngôn ngữ lập trình đều hỗ trợ cách truyền đối số cho tham số theo vị trí; một số ngôn ngữ hỗ trợ cách truyền đối số cho tham số theo tên (Python hỗ trợ, C/C++ không hỗ trợ)
- Nên chọn thứ tự và tên các tham số một cách hợp lý

Nạp chồng hàm là gì?

- **Nạp chồng hàm** (function overloading) là kĩ thuật cho phép định nghĩa nhiều hàm cùng tên
 - Khi định nghĩa hàm: các hàm phải khác nhau số lượng hoặc kiểu tham số
 - Khi dùng hàm: tùy theo số lượng và kiểu của đối số mà hàm tương ứng được gọi thực thi
- C không hỗ trợ, C++ có hỗ trợ
- Python không có khai báo kiểu nên khái niệm nạp chồng hàm “không rõ ràng”!
- (Xem minh họa)

Làm sao viết hàm hoán đổi giá trị 2 biến?

- **Cộng điểm:** Viết hàm `swap(int x, int y)` hoán đổi giá trị `x` với `y`
- Chẳng hạn

```
int a = 10;  
int b = 20;  
cout << a << " " << b;      // 10 20  
swap(a, b);  
cout << a << " " << b;      // 20 10
```

Làm sao viết hàm hoán đổi giá trị 2 biến?

Bad!	Good!
<pre>void swap(int x, int y) { int t = x; x = y; y = t; }</pre>	<pre>void swap(int &x, int &y) { int t = x; x = y; y = t; }</pre>

Truyền dữ liệu bằng giá trị và bằng tham chiếu là gì?

- Trong C/C++, chỉ có giá trị của đối số được truyền (gán) cho tham số (passing data by value)
 - ☞ các thay đổi trên tham số không ảnh hưởng đến đối số!
- Trong C++, để “truy cập” và thay đổi đối số ta có thể dùng tham số dạng **tham chiếu** (reference): là tham số có dấu & trước tên.

Mẫu số liệu và thống kê là gì?

- **Một mẫu số liệu** (sample) $\{x\}$ **cỡ** (size) n là một danh sách gồm n số thực
 x_1, x_2, \dots, x_n
- Ví dụ, mẫu số liệu $\{a\}$ cỡ $n = 5$ gồm các số $\{1, 3, 5, 2, 3\}$
- Một **thống kê trên mẫu** (sample statistic) là một con số tính từ mẫu, ví dụ
$$\min(x) = \min\{x_1, x_2, \dots, x_n\}$$
$$\max(x) = \max\{x_1, x_2, \dots, x_n\}$$
$$\text{range}(x) = \max(x) - \min(x)$$
- Chẳng hạn, với mẫu số liệu $\{a\}$ ở trên ta có: $\min(a) = 1, \max(a) = 5, \text{range}(a) = 5 - 1 = 4$

Trung bình mẫu là gì?

- Một thống kê rất hay được dùng trên mẫu là **trung bình mẫu** (sample mean) được tính là

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

- Chẳng hạn, với mẫu số liệu $\{a\}$ ở trên ta có

$$\bar{a} = \frac{1 + 3 + 5 + 2 + 3}{5} = 2.8$$

- Trung bình mẫu cho biết giá trị trung tâm (hay trọng tâm) mà các số liệu của mẫu “xoay quanh” đó

Tính trung bình mẫu như thế nào?

- Viết chương trình cho nhập mẫu số liệu và tính trung bình mẫu
- **Viết dễ!** (Cộng điểm)

Phương sai mẫu là gì?

- Một thống kê hay được dùng khác trên mẫu là **phương sai mẫu** (sample variance) được tính là

$$S_x^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_n - \bar{x})^2}{n}$$

- Chẳng hạn, với mẫu số liệu $\{a\}$ ở trên ta có

$$S_a^2 = \frac{(1 - 2.8)^2 + (3 - 2.8)^2 + (5 - 2.8)^2 + (2 - 2.8)^2 + (3 - 2.8)^2}{5} \\ = 1.76$$

- Phương sai cho thấy mức độ “phân tán” của các số liệu
 - Phương sai nhỏ cho thấy các số liệu gần nhau
 - Phương sai lớn cho thấy các số liệu xa nhau

Tính phương sai mẫu như thế nào?

- Viết chương trình cho nhập mẫu số liệu và tính phương sai mẫu
- **Viết khó!**

Giải lao!!!

- Giải lao đến 10h05

Mảng và mảng tĩnh là gì?

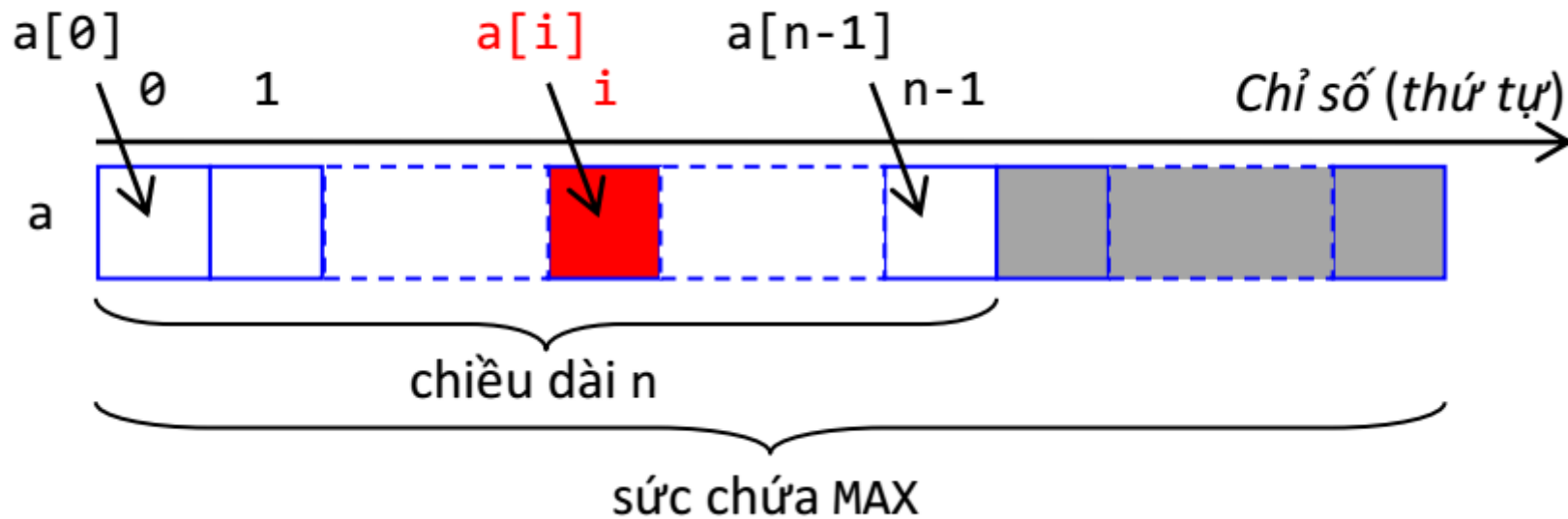
- **Mảng** (array) là một **cấu trúc dữ liệu** (data structure) giúp lưu trữ và xử lý nhiều dữ liệu cùng kiểu
- Mỗi dữ liệu trong mảng được gọi là **phần tử** (element) của mảng
- **Mảng tĩnh** (static array) là mảng được cấp phát trước với lượng bộ nhớ cố định
- Số phần tử tối đa mà mảng (tĩnh) có thể chứa được gọi là **sức chứa** của mảng
- Số phần tử thực sự trong mảng được gọi là **chiều dài** (length) của mảng

Làm sao truy cập các phần tử của mảng?

- Sau khi mảng được cấp phát, các phần tử của mảng được bố trí liên tiếp theo thứ tự trên bộ nhớ
- Hình dung **mảng một chiều** (one-dimensional array, 1D array, array) như một hàng các dữ liệu
- Các phần tử của mảng được quản lý bằng **chỉ số** (index, location)
- Chỉ số mảng trong C/C++/Python bắt đầu từ 0

Hình dung mảng một chiều!

Hình 4.1.1 – Hình minh họa mảng (một chiều) a chiều dài n sức chứa MAX



Xử lý ngoại tuyến và trực tuyến là gì?

- **Xử lý ngoại tuyến** (offline) là cách thức “nhận và lưu trữ toàn bộ dữ liệu rồi mới xử lý”
- **Xử lý trực tuyến** (online) là cách thức “xử lý liền trên từng dữ liệu nhận được”
- Xử lý offline thường đơn giản hơn xử lý online
- Mảng thường được dùng để lưu trữ và xử lý offline

Cộng điểm

- Tìm hiểu thống kê **trung vị mẫu** (sample median) và viết hàm tính thống kê này