

## Bài thực hành 12

(phiên bản 2)

### Mục tiêu

- Con trỏ và quản lý bộ nhớ
- Cấp phát động
- Số học con trỏ
- Liên hệ với class
- Phân biệt copy nông và copy sâu

### Bài tập

**Bắt buộc giải phóng tất cả các biến được cấp phát động (bằng toán tử delete) khi không còn cần đến chúng.**

#### Câu 1. [word\_order\_reverse.cpp]

Hãy viết chương trình thực hiện: Đọc vào một dòng văn bản (đọc đến khi gặp dấu xuống dòng). Đưa các từ trong dòng văn bản này vào một mảng string cấp phát động. Đảo ngược thứ tự các phần tử trong mảng rồi in ra màn hình dòng đảo ngược.

Ví dụ xâu nhập vào là `This is an example.` thì cấp phát động mảng 4 phần tử string, xâu in ra màn hình là `example. an is This`

#### Câu 2. [word\_reverse.cpp]

Hãy viết chương trình thực hiện: Đọc vào một dòng văn bản và đưa các từ vào mảng string cấp phát động (như câu 1). Đảo ngược thứ tự các kí tự trong mỗi từ rồi in ra màn hình xâu kết quả.

Ví dụ xâu nhập vào là `This is an example.` thì cấp phát động mảng 4 phần tử string, xâu in ra màn hình là `sihT si na .elpmaxe`

#### Câu 3. [array\_input1.cpp]

Viết chương trình đọc vào số nguyên dương  $n$ , cấp phát động một mảng  $n$  phần tử kiểu `int`, sau đó đọc tiếp  $n$  số nguyên vào mảng này. In lại dãy số nguyên vừa đọc.

#### Câu 4. [array\_input2.cpp]

a. Viết hàm thực hiện nhập vào mảng `int` cấp phát động như mô tả ở câu 3. Nguyên mẫu của hàm này là

```
void inputArray(int*& p, int& n);
```

Trong đó

- tham số tham chiếu `p` là biến con trỏ để lưu địa chỉ của mảng cấp phát động
- tham số tham chiếu `n` là để lưu số số nguyên nhập vào.

Trong `main()`, ta có thể gọi tới hàm `inputArray` bằng các lệnh

```
int size = 0;
int* arr;
inputArray(arr, size);
```

b. Viết hàm in mảng nguyên.

c. Dùng các hàm vừa định nghĩa để nhập 1 dãy số nguyên và in lại chúng.

#### Câu 5. [array\_input3.cpp]

Viết hàm `inputArrayExtend` đọc các số nguyên (tới khi người dùng nhập EOF) vào một mảng cấp phát động `p`.

Ban đầu cấp phát mảng `p` kích thước bằng `maxSize = 2`, số phần tử `n = 0`.

Với mỗi số nguyên `x` đọc từ bàn phím, hãy kiểm tra xem mảng `p` đã đầy chưa (nếu kích thước của mảng là `maxSize = 2` và người dùng đang nhập số thứ 3 thì phép kiểm tra này trả về `true`). Nếu đã đầy thì tăng gấp đôi kích thước mảng `p` bằng cách:

- cấp phát động một mảng mới `q` với kích thước bằng `maxSize * 2`
- sao toàn bộ dữ liệu từ mảng `p` sang mảng `q`
- thêm phần tử mới `x` vào `q`
- giải phóng bộ nhớ hiện đang bị quản lý bởi `p`
- gán `p = q`

Dùng hàm vừa định nghĩa để nhập 1 dãy số nguyên và in lại chúng.

### Câu 6. [mycstring.cpp]

Hãy định nghĩa các hàm `mystrlen`, `mystrcpy` và `mystrcat` thực hiện công việc giống các hàm `strlen` (tính độ dài chuỗi C), `strcpy` (sao chuỗi C) và `strcat` (nối 2 chuỗi C) của thư viện `<cstring>`. Không được dùng các hàm xử lý chuỗi ký tự có sẵn của C++ trong các hàm nói trên.

Nguyên mẫu các hàm thư viện:

```
size_t strlen ( const char * str );  
char * strcpy ( char * destination, const char * source );  
char * strcat ( char * destination, const char * source );
```

*Tham khảo:*

<http://cplusplus.com/reference/library/cstring/strlen/>

<http://cplusplus.com/reference/library/cstring/strcpy/>

<http://cplusplus.com/reference/library/cstring/strcat/>