# **Bài 7. Con trỏ**

**Mục tiêu:**

1. *Luyện tập sử dụng con trỏ và địa chỉ của các biến*
2. *Sử dụng con trỏ khi thao tác với mảng.*

***Giới hạn:*** *không dùng các thư viện stl (chẳng hạn vector, algorithm)*

**Yêu cầu nộp bài***: Toàn bộ 7 bài phần A,* ***và****. 3 câu hỏi phần B. Lời giải cho B viết vào file readme.txt*

***Lưu ý: chép bài hoặc cho chép bài sẽ dẫn đến trượt môn học!***

## Thực hành

1. ***Lỗi sử dụng con trỏ.*** Hãy chạy chương trình thử nghiệm các đoạn code lỗi đã cho trong bài giảng.
2. ***Mảng truyền vào hàm dạng con trỏ***. Hãy viết một chương trình demo rằng khi truyền tham số, mảng được truyền vào hàm ở dạng con trỏ chứ không phải dạng mảng.   
   Gợi ý: khai báo một mảng một chiều A, in kích thước của mảng bằng toán tử sizeof. Sau đó, viết thêm một hàm f nhận tham số kiểu mảng (thử cả mảng không xác định cũng như có xác định kích thước) và in ra kích thước của mảng. So sánh kết quả của sizeof đối với mảng A từ bên trong hàm f và bên ngoài - nơi khai báo mảng A.
3. ***Truy nhập mảng***. Viết môt hàm count\_even(int\*, int) nhận tham số là một mảng và kích thước, và trả về số số chẵn trong mảng. Viết hàm main demo việc sử dụng hàm cho một mảng (có thể hardcode - khởi tạo tại chỗ không cần nhập) cho các trường hợp gọi hàm count\_even() cho đoạn 5 phần tử đứng đầu mảng và đoạn 5 phần tử đứng cuối mảng.
4. ***Tìm kiếm nhị phân.*** Hãy viết lại chương trình tìm kiếm nhị phân mà bạn đã làm (đệ quy hay vòng lặp đều được) nhưng hàm tìm kiếm truy nhập mảng hoàn toàn bằng con trỏ.
5. ***Dangling references***. Hãy viết chương trình thử nghiệm lỗi con trỏ truy nhập vùng bộ nhớ không hợp lệ. Thử cho hàm main() gọi weird\_string() và in kết quả trả về. Xem trình biên dịch cảnh báo những gì và bạn có nhận xét gì về output của hàm main?

char\* weird\_string() {

char c[] = “123345”;  
 return c;   
}

Bài học cần rút ra từ thử nghiệm này là không được trả về địa chỉ của một vùng nhớ không hợp lệ. Mảng c là biến địa phương của hàm weird\_string, khi truy nhập mảng c khi hàm weird\_string đã kết thúc, ta đang truy nhập vào vùng bộ nhớ stack đã bị thu hồi, không thể đảm bảo dữ liệu cũ vẫn còn ở đó, thậm chí có thể dẫn đến các lỗi hệ thống nghiêm trọng khác.

1. ***Tham số dòng lệnh của hàm main()*** . Cho chương trình thử nghiệm dưới đây:

int main(int argc, const char \* argv[]) {

cout << "Number of command-line arguments: " << argc << endl;

for (int i = 0; i < argc; i++) {

cout << "Argument #" << i << ": \_" << argv[i] << "\_\n";

}

return 0;

}

Đó là một ví dụ demo việc lấy tham số dòng lệnh. Bạn cần chạy chương trình từ cửa sổ cmd. Chẳng hạn nếu biên dịch ra file test-cmd.exe. Hãy thử chạy các lệnh sau từ dấu nhắc cmd để quan sát kết quả của chương trình:

test-cmd

test-cmd first second 3rd

test-cmd 1

Hãy sửa bài A8 (BT06) để nhận N từ tham số dòng lệnh hoặc sửa domin (BT05) để nhận kích thước bảng mìn và số mìn từ tham số dòng lệnh. Ví dụ bạn có thể chạy dò mìn với kích thước 3 dòng 4 cột 2 quả mìn bằng lệnh sau tại cửa sổ cmd:

domin 3 4 2

Tất nhiên, chương trình không được yêu cầu nhập kích thước và số mìn nữa.

1. ***Hàm xử lý xâu***. Sử dụng hàm strcmp (<http://www.cplusplus.com/reference/cstring/strcmp/>) hãy viết một chương trình nhận 2 tham số từ dòng lệnh là hai xâu kí tự, đếm xem xâu thứ nhất xuất hiện bao nhiêu lần trong xâu thứ hai. In kết quả ra màn hình.

## Câu hỏi

1. Cho đoạn lệnh sau.

char c = ‘A', d = ‘B';  
char\* p1 = &c;  
char\* p2 = &d;  
char\* p3;

Giả sử địa chỉ của c là 0x1234, địa chỉ của d là 0x5678. Hỏi output của đoạn lệnh sau là gì?

p3 = &d;

cout << “\*p3 = “ << \*p3 << “, p3 = “ << p3 << endl;

//\*p3 = B, p3 = 0x5678

p3 = p1;

cout << “\*p3 = “ << \*p3 << “, p3 = “ << p3 << endl;

//\*p3 = A, p3 = 0x1234

\*p1 = \*p2;

cout << “\*p1 = “ << \*p1 << “, p1 = “ << p1 << endl;

\*p1 = B, p1 = 0x5678

1. Cho các lệnh sau.

int \*p;  
int i;  
int k;  
i = 4;  
k = i;  
p = &i;

(Những) lệnh nào trong số các lệnh dưới đây sẽ gán cho i giá trị 5?

k = 5;

\*k = 5;  
p = 5;

\*p = 5;

1. Giải thích lỗi sau:

char c = ‘C';  
double \*p = &c;

con trỏ tới kiểu double bị gán trỏ tới kiểu char

## Bài tập

1. Viết hàm mô phỏng quá trình đổ 1 quân xúc xắc (trả về giá trị trong khoảng từ 1-6),  
   Sử dụng hàm đổ xúc xắc trên viết chương trình mô phỏng trò chơi sau: Hai người chơi, lần lượt đổ 1 xúc xắc, tổng điểm được cộng gộp vào cho mỗi người chơi. Ai có tổng điểm BẰNG 100 trước là thắng. Đổi lượt đổ xúc xắc cho người chơi kia theo 2 cách:  
    A. Mỗi người chơi lần lượt đổ xúc xắc.  
    B. Người có tổng điểm nhỏ hơn sẽ được đổ xúc xắc.  
   Ví dụ với cách đổi lượt thứ nhất:  
   xuc xac nguoi choi 1: 2 Tong nguoi 1: 2  
   xuc xac nguoi choi 2: 4 Tong nguoi 2: 4  
   xuc xac nguoi choi 1: 5 Tong nguoi 2: 7  
   xuc xac nguoi choi 2: 1 Tong nguoi 1: 5  
     
   Ví dụ với cách đổi lượt thứ hai:  
   xuc xac nguoi choi 1: 2 Tong nguoi 1: 2  
   xuc xac nguoi choi 2: 4 Tong nguoi 2: 4  
   xuc xac nguoi choi 1: 5 Tong nguoi 1: 7  
   xuc xac nguoi choi 2: 1 Tong nguoi 2: 5  
   xuc xac nguoi choi 2: 1 Tong nguoi 2: 6
2. Viết chương trình giả lập trò chơi “Rùa & Thỏ”. Đường đua là một con dốc bao gồm 70 vị trí (tương ứng với 70 vị trí trên màn hình). Điểm đích của đường đua là đỉnh dốc, tương đương vị trí 70. Cả rùa và thỏ bắt đầu từ chân dốtc, tương đương vị trí 0. Rùa và thỏ cùng bắt đầu cuộc đua. Sau mỗi lượt, vị trí của rùa và thỏ trên đường đua được thay đổi ngẫu nhiên như sau:  
   Vật Đua Loại chuyển động Tỷ lệ Bước chuyển động  
   Rùa Tiến dài 50% Tiến 3 (+3)  
    Tiến ngắn 30% Tiến 1 (+1)  
    Trượt 20% Trượt 6 (-6)  
   Thỏ Ngủ 20% Đứng im (0)  
    Tiến dài 20% Tiến 9 (+9)  
    Trượt dài 10% Trượt 12 (-12)  
    Tiến ngắn 30% Tiến 1 (+1)  
    Trượt ngắn 20% Trượt 2 (-2)  
   Chú ý: chú ý nếu trượt quá vạch xuất phát thì đưa rùa/thỏ trở lại vạch xuất phát (không có vị trí <0).  
   Viết hàm giả lập chuyển động cho rùa & thỏ như ở bảng trên.  
   Thông báo khi cuộc đua bắt đầu và kết thúc. Thông báo người thắng cuộc. Thông báo vị trí của rùa & thỏ sau mỗi lượt, in ra R (rùa) & T (thỏ). In ra vị trí bắt đầu và kết thúc của đường đua trong suốt cuộc đua. In ra chi tiết mỗi bước chuyển động của rùa và thỏ.  
   Nâng cao: sử dụng con trỏ hàm để viết hàm giả lập chuyển động cho rùa & thỏ:

void chay(int& viTri, int (\*buoc)());

1. Viết chương trình giả lập trò chơi “Tiến Lên Kiểu Úc” cho 4 người chơi.   
   Bộ bài gồm 52 quân, mỗi quân bài có thông tin về số (2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A) và chất (♠, ♣, ♦, ♥). Mỗi quân bài có quyền ưu tiên dựa trên số rồi chất (nếu số giống nhau). Mức ưu tiên tăng dần theo số: 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A; và theo chất ♠, ♣, ♦, ♥. Ví dụ 4♦ ưu tiên cao hơn 4♣ và 5♠ ưu tiên cao hơn 4♥. Ngoài ra nên có thông tin khóa cho mỗi quân bài (dựa trên khóa có thể xác định chính xác quân bài theo số & chất). Ví dụ khóa 0 cho quân 2♠, khóa 51 cho quân A♥.   
    1. Viết hàm xác định quân bài dựa trên khóa và quyền ưu tiên giữa hai quân bài dựa trên khóa:  
    string inQuanBai(int khoa);  
    bool uuTienHon(int khoa1, int khoa2);  
   **Bước 1 – Tráo bộ bài:** 52 quân bài được sắp xếp một cách ngẫu nhiên (có thể sử dụng khóa của quân bài).  
    2. Viết hàm tráo bài:   
    void traoBai(int \*boBai);  
   **Bước 2 – Chia bài:** Chia hết bộ bài (52 quân bài) sau khi đã được tráo cho 4 người chơi, mỗi người lần lượt nhận một quân bài từ bộ bài đã tráo. Sau khi chia bài, mỗi người chơi có 13 quân bài.  
    3. Viết hàm chia bài và hàm in bộ bài của mỗi người chơi  
    void chiaBai(int \*boBai, int \*\*boBaiNguoiChoi);  
    void inBoBaiNguoiChoi(int \*boBaiNguoiChoi);  
   **Bước 3 – Chơi bài:** Người chơi có 2♠ (quân bài có quyền ưu tiên thấp nhất) được chơi đầu tiên & người chơi đó phải chơi quân bài này.  
   Người chơi tiếp theo (xác định dựa trên thứ tự ở Bước 2 – Chia bài) chọn một quân bài ngẫu nhiên để chơi. Quân bài này được phép chơi ra nếu có quyền ưu tiên cao hơn quân bài người trước vừa chơi. Nếu không người đó sẽ mất lượt chơi tại vòng này.  
   Người mất lượt sẽ không được chơi trong vòng cho tới khi vòng chơi kết thúc. Vòng chơi kết thúc khi chỉ còn lại một người chơi. Khi đó vòng mới bắt đầu (có 4 người chơi) với việc người cuối cùng còn lại trong vòng trước chơi một quân bài bất kỳ.  
   Ván bài kết thúc ngay sau khi có một người chơi hết bài (không còn quân bài nào) và là người thắng ván đó.  
   Tính điểm: mỗi quân bài còn lại của 3 người thua trong ván được tính là 1 điểm. Toàn bộ số điểm này được cộng cho người thắng ván đó.  
   Ván bài mới bắt đầu lại từ việc tráo bài, chia bài, chơi bài (2♠) cho tới khi ai được 100 điểm đầu tiên thì trò chơi kết thúc.  
    4. Viết hàm xác định người chơi đầu tiên bắt đầu mỗi ván (2♠)   
    int aiCo2Bich(int \*\*inBoBaiNguoiChoi);  
    5. Viết hàm chọn một quân bài bất kỳ từ bộ bài của người chơi.  
    int chonBai(int \*boBaiNguoiChoi);  
    6. Viết hàm cho phép chơi quân bài vừa chọn.  
    void choiBai(int \*boBaiNguoiChoi, int quanBaiVuaChoi);  
    7. Viết hàm loại một người chơi khỏi vòng chơi & hàm xác định người chơi tiếp theo của vòng chơi.  
    void loaiNguoiChoi(int\*, int&, int);  
    int nguoiChoiTiepTheo(int\*, int, int);  
    8. Viết hàm kiểm tra xem đã có người chơi nào hết bài chưa.  
    bool chuaAiHetBai(int \*\*boBaiNguoiChoi);  
     
   In ra các thông tin chi tiết trong quá trình chơi bài:  
   • Các quân bài hiện tại của mỗi người chơi  
   • Quân bài vừa chơi là quân gì (ai chơi).  
   • Người chơi tiếp theo là ai, chơi quân gì, có mất lượt không.  
   • Tổng điểm của mỗi người khi ván bài kết thúc.  
   • ………

D. Câu hỏi trắc nghiệm

1. Câu lệnh nào sau đây thực hiện gán giá trị của n cho con trỏ \*p  
   int n = 5; int \*p;  
   i. p = n;  
   ii. p = &n;  
   iii. \*p = n;  
   iv. \*p = &n;
2. Câu lệnh nào sau đây thực hiện chỉ con trỏ \*p1 vào vị trí con trỏ \*p2 đang chỉ  
   i. \*p1 = p2;  
   ii. p1 = \*p2;  
   iii. p1 = p2;  
   iv. \*p1 = \*p2
3. Đối với phép toán con trỏ, phát biểu nào sau đây có ý nghĩa lập trình   
   i. Phép nhân trên con trỏ kiểu int  
   ii. Phép cộng trừ trên con trỏ kiểu int  
   iii. Phép gán với con trỏ kiểu int  
   iv. Tất cả các phép toán với con trỏ kiểu int
4. Phát biểu nào sau đây chính xác   
   i. Phép cộng trừ với các con trỏ ở mảng khác nhau   
   ii. Phép cộng trừ với các con trỏ ở cùng mảng  
   iii. Tất cả các phép toán với các con trỏ ở cùng mảng (trong giới hạn mảng)  
   iv. Tất cả các phát biểu trên
5. Phát biểu nào sau đây đúng với con trỏ tới hàm  
   i. Có thể được truyền vào trong trong  
   ii. Có thể được trả về bới hàm  
   iii. Có thể được lưu trong mảng  
   iv. Tất cả các phát biểu trên
6. Khai báo hàm nào sau đây không đúng  
   i. void bubble(int [], const int, bool (\*) (int, int));  
   ii. void bubble(int [], const int, bool \* (int, int));  
   iii. void bubble(int [], const int, (bool \*) (int, int));  
   iv. void bubble(int [], const int, bool (\* (int, int)));
7. Kết quả đoạn mã sau là gì:  
    int a[6] = {1, 2, 3};  
    int \*b; b = &a[0];  
    for (int i=0; i<3; i++)  
    switch (i%3) {  
    case 0: \*(b+i) = a[i]; break;  
    case 1: \*(b++) = a[i]; break;  
    case 2: \*(++b) = a[i]++; break;  
    }  
    for (int i=0; i<3; i++) cout << a[i] << " ";  
   i. 0 2 3  
   ii. 2 2 0  
   iii. 2 2 3  
   iv. 3 0 3