



## CTT12 – KỸ THUẬT LẬP TRÌNH

### BÀI TẬP VỀ NHÀ

#### BTVN-01: CON TRỎ

#### I. Thông tin chung

Mã số bài tập:	BTVN-01
Thời lượng dự kiến:	5 – 8 tiếng
Deadline nộp bài:	
Hình thức:	Bài tập cá nhân
Hình thức nộp bài:	Nộp qua Moodle môn học
GV phụ trách:	Võ Hoài Việt
Thông tin liên lạc với GV:	<a href="mailto:vhviet@fit.hcmus.edu.vn">vhviet@fit.hcmus.edu.vn</a>

#### II. Chuẩn đầu ra cần đạt

Bài tập này nhằm mục tiêu đạt được các chuẩn đầu ra sau:

- Hiểu được khái niệm con trỏ
- Sử dụng được con trỏ

#### III. Mô tả bài tập

**Bài 1:** Sinh viên biên dịch và chạy đoạn code bên dưới. Trả lời các câu hỏi sau:

```
int _tmain(int argc, _TCHAR* argv[])
{
    int a = 1, b = 3;
    printf("Giá trị: a = %d, j = %d\n", a, b);
    printf("Địa chỉ: a = %p, b = %p\n", &a, &b);
    return 0;
}
```

Hãy cho biết giá trị và địa chỉ của biến sau 5 lần chạy và cho nhận xét về kết quả trên.

Lần chạy	Biến a		Biến b	
	Giá trị	Địa chỉ	Giá trị	Địa chỉ
1				
2				
3				
4				
5				
Nhận xét				

**Bài 2:** Sinh viên biên dịch và chạy đoạn code bên dưới. Trả lời các câu hỏi sau:

```
int _tmain(int argc, _TCHAR* argv[])
{
    int a = 1; // line 1
    int *pointer; // line 2
    float b = 0.00; // line 2
    printf("i = %p \n", &a); // line 4
    pointer = &a; // line 5
    printf("p = %p\n", pointer); // line 6
    return 0;
}
```

- 2.1 Cho biết giá trị và địa chỉ của biến *a* và *pointer*.
- 2.2 Có thể gán trực tiếp giá trị cho biến *pointer* hay không? Ví dụ *pointer* = 10000
- 2.3 Có thể gán địa chỉ của biến kiểu *b* vào biến *pointer* không (*pointer* = *&b*)? Tại sao?
- 2.4 Biến *pointer* có địa chỉ của nó trong bộ nhớ không? Nếu có, hãy cho biết địa chỉ của nó. Dùng lệnh `printf("p = %p\n", &pointer);`

**Bài 3:** Sinh viên biên dịch và chạy đoạn code bên dưới. Trả lời các câu hỏi sau:

```
int _tmain(int argc, _TCHAR* argv[])
{
    char a = 'A', *pa;
    float x = 2.5F, *px;
    int i = 1, *m;
    pa = &a;
    px = &x;
    m = &i;
    printf("pa = %p\npx = %p\n m = %p\n", pa, px, m);
    return 0;
}
```

- 3.1 Cho biết các biến *a*, *pa*, *px*, *m*, *x*, *i* được lưu trữ ở đâu và giá trị của chúng.
- 3.2 Biến nào lưu ký tự và biến nào lưu địa chỉ đến vùng nhớ chứa ký tự?
- 3.3 Ta có thể lưu địa chỉ của biến *x* trong *pa* không (*pa* = *&x*)? Có thể lưu địa chỉ của biến *i* trong *pa* không (*pa* = *&i*)? Nếu không, thông báo lỗi gì sẽ xuất hiện.

**Bài 4:** Sinh viên biên dịch và chạy đoạn code bên dưới. Trả lời các câu hỏi sau:

```
int _tmain(int argc, _TCHAR* argv[])
{
    char a = 'A', b = 'B', *pa, *pb; // line 1
    pa = &a; // line 2
    pb = pa; // line 3
    printf("pa = %p, pb = %p\n", pa, pb); // line 4
    pa = &b; // Line 5
    printf("pa = %p, pb = %p\n", pa, pb); //line 6
    return 0;
}
```

- 4.1 Có thể gán biến con trỏ cho một biến con trỏ khác không, như trong line 3?
- 4.2 Cho biết địa chỉ của biến *a* và *b*.
- 4.3 Cho biết địa chỉ và giá trị của biến *pa* và *pb* thay đổi như thế nào trong đoạn code trên. Giải thích.

Bài 5: Sinh viên sử dụng con trỏ để làm các bài tập sau đây

### 5.1 Mảng một chiều

- Nhập vào một mảng gồm  $n$  phần tử nguyên ( $1 \leq n \leq 10,000$ ) và một số nguyên  $X$ :
  - Những phần tử nào (in ra cả giá trị và vị trí của phần tử đó) là ước số của  $X$ ?
  - Dồn những phần tử là ước số của  $X$  về cuối mảng. Sắp xếp các phần tử không phải là ước số của  $X$  theo thứ tự tăng dần (giữ nguyên vị trí các phần tử là ước số của  $X$ ).
- Nhập vào một mảng các số nguyên dương gồm  $n$  phần tử ( $1 \leq n \leq 15$ ):
  - Đếm số phần tử tận cùng là 6 và chia hết cho 6 trong mảng
  - Tính trung bình cộng các số nguyên tố hiện có trong mảng
  - Cho biết trong mảng có bao nhiêu số nguyên tố phân biệt
- Cho mảng  $A$  gồm  $n < 1000$  phần tử nguyên  $|A[i]| \leq 10,000$ . Viết hàm thực hiện các công việc sau:
  - Trích những phần tử trong  $A$  không phải số nguyên tố ra mảng  $B$
  - Sắp giảm các số nguyên trong mảng  $B$
  - Xóa những số nguyên tố trong mảng  $A$
- Nhập 2 dãy số nguyên  $A, B$  gồm  $m, n$  phần tử ( $1 \leq n, m \leq 25$ ):
  - Xuất ra những phần tử có trong  $A$  mà không có trong  $B$
  - Ghép  $A, B$  thành  $C$  sao cho  $C$  không có phần tử trùng nhau

### 5.2 Mảng hai chiều

- Nhập xuất ma trận số nguyên
- Tính tổng các phần tử dương trong ma trận
- Đếm số lượng số nguyên tố trong ma trận
- Tìm số lớn nhất trên biên ma trận.
- Tìm số dương nhỏ nhất trong ma trận
- Liệt kê các dòng có chứa các giá trị âm trong ma trận.
- Liệt kê các dòng chứa toàn số chẵn trong ma trận.
- Đếm số lượng giá trị “Yên ngựa” trên ma trận. Một phần tử được gọi là “yên ngựa” khi nó lớn nhất trên dòng và nhỏ nhất trên cột.
  - Đếm số lượng giá trị “Hoàng hậu” trên ma trận. Một phần tử được gọi là hoàng hậu khi nó lớn nhất trên dòng, trên cột và hai đường chéo đi qua nó
  - Tính tổng các phần tử cực trị trong ma trận. Một phần tử gọi là cực trị khi nó lớn hơn các phần tử xung quanh hoặc nhỏ hơn các phần tử xung quanh.
  - Tìm chữ số xuất hiện nhiều nhất trong ma trận
  - Sắp xếp các giá trị nằm trên biên ma trận tăng dần theo chiều kim đồng hồ

### Bài 6: Định nghĩa cấu trúc dữ liệu PhanSo

Thực hiện các yêu cầu sau:

#### 6.1 Viết hàm Nhập và Xuất phân số

#### 6.2 Viết hàm tính Tổng, Hiệu, Tích, Thương của 2 phân số

#### 6.3 Viết hàm Rút gọn (tối giản) phân số

#### 6.4 Viết hàm so sánh 2 phân số $a$ và $b$ . Hàm so sánh trả về 1 nếu $a > b$ , 0 nếu $a = b$ và -1 nếu $a < b$

Bài 7: Dựa vào cấu trúc dữ liệu PhanSo ở bài 7, viết chương trình xử lý các thao tác sau trên dãy các PhanSo:

#### 7.1 Viết hàm Nhập và Xuất dãy phân số có $N$ phần tử

#### 7.2 Viết hàm tính tổng các phân số có trong dãy

7.3 Viết hàm thay thế các phân số không tối giản có trong dãy trên bằng chính phân số tối giản tương ứng

7.4 Viết hàm Sắp xếp dãy phân số trên tăng dần

7.5 Cho trước một phân số K (do người dùng nhập vào), viết chương trình kiểm tra xem phân số K có trong dãy trên hay không? Nếu có trả về 1, ngược lại trả về 0.

Bài 8: Khai báo cấu trúc biểu diễn đường tròn trong mặt phẳng 2D

8.1 Viết hàm nhập và xuất ra màn hình cấu trúc đường tròn.

8.2 Viết hàm tính chu vi và diện tích của đường tròn.

8.3 Viết hàm kiểm tra tương quan giữa một điểm đối và một đường tròn.

8.4 Viết hàm nhập và xuất mảng đường tròn.

8.5 Viết hàm sắp xếp mảng đường tròn tăng dần theo diện tích.

#### IV. Các yêu cầu & quy định chi tiết cho bài nộp

- Bài nộp được nén .RAR hoặc .ZIP và được nộp trên moodle. Với cấu trúc tên tập tin theo tứ tự mã số sinh viên SV1\_SV2.RAR hoặc SV1\_SV2.ZIP ( Ví dụ: 0912496\_0912407.RAR)

#### V. Hướng dẫn chi tiết

#### VI. Cách đánh giá

STT	Tên kết quả	Tỉ lệ điểm	Ghi chú
1	Mã nguồn	80%	Cung cấp các thư viện và mã nguồn đầy đủ để biên dịch.
2	Phong cách lập trình	20%	Cấu trúc chương trình rõ ràng, hàm/ biến đặt tên dễ hiểu và gọi nhớ và tuân thủ các qui tắc lập trình.

#### VII. Tài liệu tham khảo

Slide bài giảng lý thuyết

#### VIII. Các quy định khác

- Chương trình phải có hướng dẫn sử dụng (Không có hướng dẫn sử dụng sẽ bị trừ 50% số điểm của phần phần chương trình).
- Tất cả các bài làm sai quy định đều bị 0 điểm cho mỗi bài.
- Hai bài giống nhau từ 80% trở lên sẽ bị 0 điểm cho cả hai bất kể ai là tác giả.
- Các trường hợp sử dụng mã nguồn không ghi rõ nguồn tham khảo sẽ bị điểm 0 cho tất cả các bài và các tác giả).
- Các bài làm xuất sắc sẽ được điểm cộng.
- Không nhận bài nộp trễ qua mail.