

# ĐỀ THI CUỐI KỲ

Môn **NHẬP MÔN LẬP TRÌNH**

Lớp **CLC-Khóa 2019**

Ngày thi **30/12/2019**

Thời gian **90 phút**

Học kỳ **1**

Năm học **2019-2020**

## LƯU Ý

- Đề thi gồm **6** bài. Điểm tối đa của bài kiểm tra này là **100** điểm.
- Tổng số điểm của các bài trong đề thi lớn hơn 100. Sinh viên được phép làm nhiều bài nhất có thể.
- Sinh viên **ĐƯỢC** sử dụng tài liệu là **một** tờ giấy A4 do cá nhân viết tay, không photo.
- Sinh viên cần tuân thủ các hướng dẫn trình bày mã nguồn: viết mã nguồn rõ ràng, ngay thẳng, đặt tên biến, tên hàm,... có ý nghĩa.

## ĐỀ BÀI

**BÀI 1** (15 điểm) Cho mảng số nguyên  $a$  gồm  $n$  phần tử  $(a_0, a_1, \dots, a_{n-1})$ . Hãy viết hàm tìm vị trí  $i$  xuất hiện đầu tiên trên mảng đã cho thỏa điều kiện:

$$a_0 + a_1 + \dots + a_{i-1} = a_{i+1} + a_{i+2} + \dots + a_{n-1}$$

Nếu không tìm được giá trị  $i$  nào thỏa điều kiện thì hàm sẽ trả ra giá trị  $-1$ .

Ví dụ:

- Với mảng  $a = \{7, 1, 2, 3, 5\}$ , hàm sẽ trả ra giá trị 2 (vì  $a_0 + a_1 = 7 + 1 = a_3 + a_4 = 3 + 5 = 8$ ).
- Với mảng  $a = \{7, 1, 8, 2, 3, 5\}$ , hàm sẽ trả ra giá trị  $-1$ .

**BÀI 2** Các sinh viên học môn *Nhập môn lập trình* của khóa 2019 được yêu cầu viết một hàm để tìm giá trị lớn thứ hai trên một mảng hai chiều các số nguyên  $A$  gồm  $nRows$  dòng và  $nCols$  cột.

Giả sử có hai hàm được cài đặt sẵn chính xác và đúng đắn với nhiệm vụ được mô tả như sau:

```
int SecondLargest(int A[], int N);  
int Collargest(int Matrix[][COLSIZE], int nRows, int nCols, int arrLargest[]);
```

- Hàm **SecondLargest** tìm phần tử có giá trị lớn thứ hai của một mảng số nguyên  $A$  gồm  $N$  phần tử. Ví dụ, với mảng số nguyên  $A$  gồm các phần tử  $\{9, 9, 1, 2, 6, 1, 8\}$  thì hàm **SecondLargest** trả ra giá trị 8.
- Hàm **Collargest** tìm giá trị lớn nhất của các cột của mảng hai chiều các số nguyên  $Matrix$  kích thước  $nRows$  dòng và  $nCols$  cột. Kết quả các giá trị lớn nhất được lưu trong mảng một chiều **arrLargest** có  $nCols$  phần tử.

Một sinh viên của lớp, bạn Văn Toàn, thực hiện yêu cầu của đề bài, đã cài đặt đoạn chương trình như sau:

```
//COLSIZE là hằng số, xác định số lượng cột tối đa
int SecondLargest_Matrix(int Matrix[][COLSIZE], int nRows, int nCols)
{
    int arrLargest[COLSIZE];
    int second_largest;
    //Tìm các giá trị lớn nhất ứng với mỗi cột của mảng hai chiều Matrix
    Collargest(Matrix, nRows, nCols, arrLargest);
    //Tìm phần tử lớn thứ hai dựa trên mảng arrLargest
    second_largest = SecondLargest(arrLargest, nCols);
    return second_largest;
}
```

Đọc kỹ đoạn chương trình đã cho, bạn cần thực hiện các nhiệm vụ sau đây:

- (a) (10 điểm) Hàm `SecondLargest_Matrix` của bạn Văn Toàn có thực hiện chính xác (hoàn chỉnh) theo yêu cầu của đề bài (tìm giá trị lớn thứ hai trên một mảng hai chiều các số nguyên  $A$  gồm  $nRows$  dòng và  $nCols$  cột) hay không? Giải thích vắn tắt (thông qua các ví dụ, minh họa, lý lẽ,...) cho câu trả lời của bạn.
- (b) (10 điểm) Cài đặt hàm `SecondLargest_Matrix` của riêng bạn với yêu cầu bắt buộc phải sử dụng hàm `SecondLargest` đã cho.

**BÀI 3** (20 điểm) Viết hàm tạo ra một chuỗi ký tự từ một mảng hai chiều các phần tử nguyên gồm  $nRows$  dòng và  $nCols$  cột. Biết rằng, các phần tử trong cùng một dòng sẽ cách nhau bằng một ký tự khoảng trắng ( ' ') và các dòng nối tiếp nhau từ dòng đầu tiên đến dòng cuối cùng, giữa các dòng sẽ cách nhau bằng một ký tự xuống dòng ( '\n' ).

Ví dụ, mảng hai chiều gồm 4 dòng, 3 cột với nội dung như sau

```
12  12  2019
11  11  2020
-1   -2  -3
45  67  89
```

sẽ tạo được chuỗi kết quả " 12 12 2019\n 11 11 2020\n -1 -2 -3\n 45 67 89\n"

(Gợi ý về một số hàm xử lý trên chuỗi C-String được nêu ở trang cuối của đề thi này).

**BÀI 4** Giờ làm việc của công ty FIT-HCMUS từ 8:00 sáng đến 18:00 chiều. Công ty muốn chấm công cho tất cả nhân viên hai lần mỗi ngày (lần 1: giờ đi làm buổi sáng, và lần 2: giờ ra về buổi chiều). Mỗi nhân viên sau khi chấm công sẽ được tính công cho ngày đó theo quy tắc sau:

- Nếu buổi sáng chấm công trước ( $\leq$ ) 8:00 và buổi chiều chấm công sau ( $\geq$ ) 18:00 thì được tính là một (01) ngày công.
- Nếu buổi sáng chấm công sau ( $>$ ) 8:00 (gọi là *đi trễ*) hoặc buổi chiều chấm công trước ( $<$ ) 18:00 (gọi là *về sớm*) thì được tính là nửa (0.5) ngày công.

- Nếu đi làm trễ và ra về sớm (vi phạm cả hai) thì được tính là không (0) ngày công.

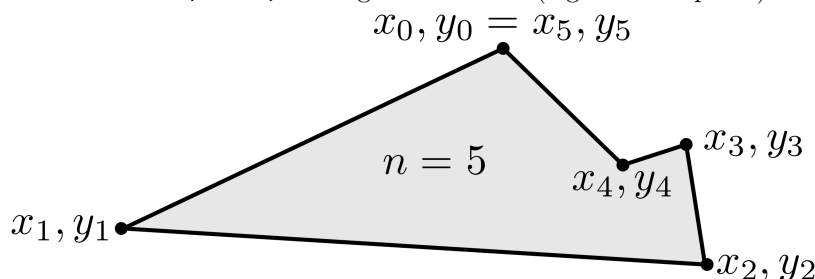
Hãy thực hiện các yêu cầu sau đối với một nhân viên của công ty trong  $N$  ngày làm việc.

- (10 điểm) Viết hàm nhập vào giờ chấm công (thông tin giờ, phút) buổi sáng và buổi chiều.
- (10 điểm) Dựa vào dữ liệu chấm công ở câu  $a$ , viết hàm xác định số ngày đi trễ, số ngày về sớm và số ngày vừa đi trễ vừa về sớm của nhân viên đó.
- (10 điểm) Dựa vào dữ liệu chấm công từ câu  $a$ , viết hàm tính số ngày công cho nhân viên đó.

**BÀI 5** Đa giác là một đối tượng quen thuộc trong hình học Euclide. Một định nghĩa về đa giác được nêu phía dưới và minh họa bằng hình 1.

*Đa giác là một đường gấp khúc phẳng khép kín, nghĩa là gồm những đoạn thẳng nối tiếp nhau cùng nằm trên một mặt phẳng và khép kín (điểm nối đầu trùng với điểm nối cuối). Phần mặt phẳng giới hạn bởi đường đa giác được gọi là hình đa giác.*

Hình 1: Một ví dụ về đa giác có 5 đỉnh. (Nguồn: Wikipedia)



Bạn hãy thực hiện những yêu cầu sau:

- (5 điểm) Đề xuất và khai báo (các) kiểu cấu trúc (**struct**) thích hợp để biểu diễn một đa giác như mô tả phía trên.
- (15 điểm) Sử dụng cấu trúc đã khai báo và cài đặt hàm tính chu vi của một đa giác (là độ dài đường bao quanh hình đa giác đó).  
(Lưu ý: Trong trường hợp cần sử dụng một (hoặc nhiều) hàm chưa có sẵn, sinh viên phải cài đặt thêm.)

**BÀI 6** Một chương trình quản lý danh bạ điện thoại chứa các thông tin liên hệ trong một tập tin văn bản có tên là `contacts.txt`. Mỗi dòng của tập tin này là một thông tin liên hệ gồm 3 thông tin: (Họ) Tên ghi nhớ - `HoTen`, Địa chỉ - `DiaChi`, Số điện thoại - `SoDienThoai`. Các thông tin này được đặt theo trật tự như sau: đầu tiên là `HoTen`, kế đến là `DiaChi`, cuối cùng là `SoDienThoai`, biết rằng thông tin `DiaChi` được đặt trong cặp ký tự ( và ).

Một phần của tập tin `contacts.txt` được minh họa như dưới đây:

```
Thoa(227 Nguyen Van Cu, Quan 5)0987654321
Dung Le(135 Tran Hung Dao, Quan 1)0912345678
Nguyen Van Huy(137 Nguyen Chi Thanh, Quan 5)0984123123
Kim Trang(KP6, Linh Trung)02838967366
```

Là người tham gia xây dựng chương trình quản lý danh bạ điện thoại này, bạn hãy:

- (a) (5 điểm) Đề xuất và khai báo kiểu cấu trúc (**struct**) thích hợp để chứa một thông tin liên hệ theo mô tả phía trên.
- (b) (10 điểm) Cài đặt hàm đọc toàn bộ danh bạ (gồm tất cả thông tin liên hệ) có trong tập tin **contacts.txt**.
- (c) (10 điểm) Cài đặt hàm lọc trong danh bạ những thông tin liên hệ có số điện thoại bắt đầu bằng chuỗi "098" rồi ghi xuống tập tin văn bản có tên là **output.txt**.  
Ví dụ, tập tin **output.txt** gồm những thông tin liên hệ được lọc từ dữ liệu **contacts.txt** (được mô tả phía trên).

Thoa(227 Nguyen Van Cu, Quan 5)0987654321

Nguyen Van Huy(137 Nguyen Chi Thanh, Quan 5)0984123123

HẾT

## GỢI Ý - Một số gợi ý hàm xử lý trên chuỗi ký tự C-String

### 1. Sao chép chuỗi

Thư viện: **cstring**. Hàm: **strcpy()**

Ví dụ:

```
char strSource[50] = "Hello World"; //chuỗi nguồn
char strDest[50]; //chuỗi đích
strcpy(strDest, strSource); //strDest: "Hello World"
```

### 2. Ghép chuỗi

Thư viện: **cstring**. Hàm: **strcat()**

Ví dụ:

```
char strSource[50] = "Hello World"; //chuỗi nguồn
char strDest[50] = "!"; //chuỗi gắn thêm
strcat(strSource, strDest); //strSource: "Hello World!"
```

### 3. Đổi số thành chuỗi

Thư viện: **string**. Hàm **to\_string()** và **c\_str()**

Ví dụ:

```
char strDest[50];
int num = 50;
strcpy(strDest, to_string(num).c_str()); //strDest: "50"
```