

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI CÔNG NGHỆ THÔNG TIN  
KHOA CÔNG NGHỆ PHẦN MỀM

MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG  
BÀI TẬP THỰC HÀNH 1

**OBJECT-ORIENTED PROGRAMMING  
EXERCISE 1**

CỦ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

TP. HỒ CHÍ MINH, NĂM 2024



**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI CÔNG NGHỆ THÔNG TIN  
KHOA CÔNG NGHỆ PHẦN MỀM**

**TRẦN VẠN TẤN - 23521407**

**MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG  
BÀI TẬP THỰC HÀNH 1**

**OBJECT-ORIENTED PROGRAMMING  
EXERCISE 1**

**CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH**

**GIẢNG VIÊN HƯỚNG DẪN  
CN. NGUYỄN NGỌC QUÍ**

**TP. HỒ CHÍ MINH, NĂM 2024**



## **NHẬN XÉT CỦA GIẢNG VIÊN**



# LỜI CẢM ƠN

*TP. Hồ Chí Minh, ngày 30 tháng 9 năm 2024*

*Tác giả*

*Trần Văn Tấn - 23521407*

# Mục lục

<b>NHẬN XÉT CỦA GIẢNG VIÊN</b>	<b>i</b>
<b>LỜI CẢM ƠN</b>	<b>iii</b>
<b>Mục lục</b>	<b>iv</b>
<b>Danh sách hình vẽ</b>	<b>v</b>
<b>Danh sách bảng</b>	<b>vi</b>
<b>Danh sách từ viết tắt</b>	<b>vii</b>
<b>Tóm tắt đề tài</b>	<b>1</b>
<b>1 Nội dung bài làm</b>	<b>3</b>
1.1 Bài tập 1 . . . . .	3
1.2 Bài tập 2 . . . . .	4
1.3 Bài tập 3 . . . . .	6
1.4 Bài tập 4 . . . . .	7
1.5 Bài tập 5 . . . . .	8
1.6 Bài tập 6 . . . . .	11
1.7 Bài tập 6 <sub>2</sub> . . . . .	12
1.8 Bài tập 7 . . . . .	15
1.9 Bài tập 8 . . . . .	18



# Danh sách hình vẽ

1.1	.....	4
1.2	.....	4
1.3	.....	5
1.4	.....	5
1.5	.....	6
1.6	.....	7
1.8	.....	8
1.9	.....	8
1.10	.....	10
1.11	.....	10
1.12	.....	12
1.13	.....	14
1.14	.....	15
1.15	.....	17
1.16	.....	20

# **Danh sách bảng**

## **Danh sách từ viết tắt**



## **Tóm tắt đề tài**



# Chương 1

## Nội dung bài làm

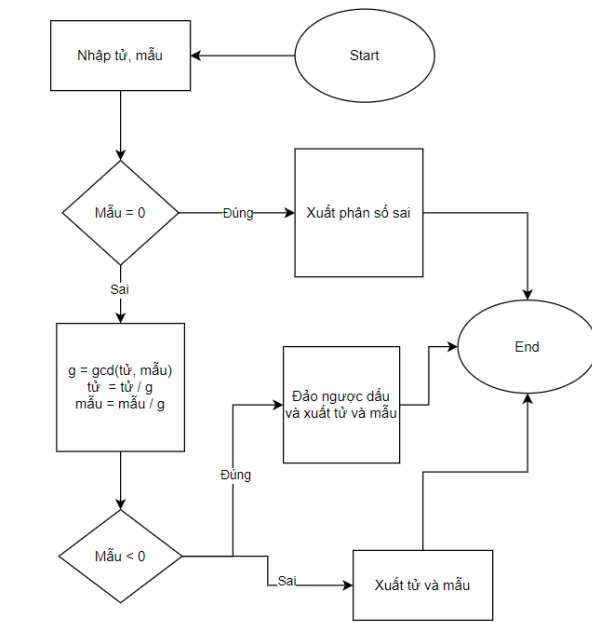
### 1.1 Bài tập 1

**Đề bài:** Viết chương trình nhập vào một phân số, rút gọn phân số và xuất kết quả.

**Mô tả Input và Output bài toán:**

- Input: Hai số nguyên  $a, b$  biểu diễn phân số  $\frac{a}{b}$
- Output: Thông báo lỗi hoặc phân số  $\frac{a}{b}$  đã được rút gọn
- Solution: Nếu mẫu số(giá trị  $b$ ) được nhập vào bằng 0 thì báo lỗi vì không thể có phân số như vậy. Nếu các số được nhập vào là hợp lệ thì ta tìm  $g = \gcd(a, b)$ , khi đó sẽ tồn tại 2 số nguyên  $p, q$  sao cho  $a = g \cdot p, b = g \cdot q$ . Ta có  $g$  là nhân tử chung nên ta rút gọn  $g$ , ta có được kết quả là phân số  $\frac{p}{q}$ , có  $\gcd(p, q) = 1$  nên tối giản.

**Flow Chart:**



Hình 1.1

Code: [Lab 1 bài 1](#)

Kiểm thử các test case:

```

D:\UIT\LTHDT\Lab_1>baitap1.exe
Hay nhap vao phan so: 100 0
Mau so khong the bang 0
D:\UIT\LTHDT\Lab_1>baitap1.exe
Hay nhap vao phan so: 36 8
Phan so toi gian: 9/2
D:\UIT\LTHDT\Lab_1>baitap1.exe
Hay nhap vao phan so: 6 -15
Phan so toi gian: -2/5
  
```

Hình 1.2

## 1.2 Bài tập 2

**Đề bài:**Viết chương trình nhập vào hai phân số, tìm phân số lớn nhất và xuất kết quả.

**Mô tả Input và Output bài toán:**

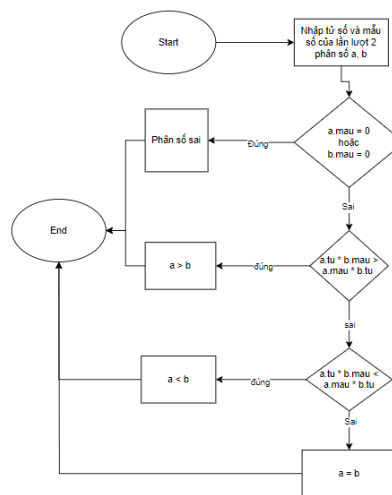
- Input: Gồm 4 số nguyên lần lượt là tử và mẫu của 2 phân số a, b



## 1.2. Bài tập 2

- Output: Phân số lớn hơn hoặc báo lỗi
- Solution: Mẫu của 1 trong 2 phân số bằng 0 thì ta báo lỗi. Nếu input hợp lệ thì ta so sánh 2 phân số a và b. In ra số lớn hơn là a hoặc b hoặc phân số a bằng phân số b

**Flow Chart:**



Hình 1.3

**Code:** [Lab 1 bài 2](#)

**Kiểm thử các test case:**

```
D:\UIT\LTHDT\Lab_1>baitap2.exe
Hay nhap vao phan so thu nhat: 100 0
Hay nhap vao phan so thu hai: 6 29
Mau so khong the bang 0
D:\UIT\LTHDT\Lab_1>baitap2.exe
Hay nhap vao phan so thu nhat: 36 8
Hay nhap vao phan so thu hai: 9 2
Phan so a bang phan so b
D:\UIT\LTHDT\Lab_1>baitap2.exe
Hay nhap vao phan so thu nhat: 32 27
Hay nhap vao phan so thu hai: 16 59
Phan so a lon hon phan so b
D:\UIT\LTHDT\Lab_1>baitap2.exe
Hay nhap vao phan so thu nhat: 45 98
Hay nhap vao phan so thu hai: 107 34
Phan so a nho hon phan so b
```

Hình 1.4

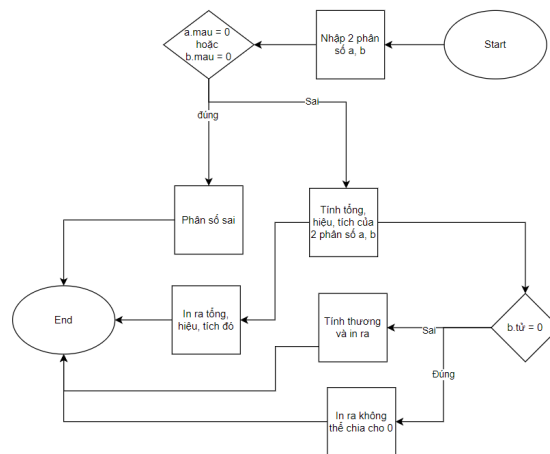
## 1.3 Bài tập 3

**Đề bài:**Viết chương trình nhập vào hai phân số. Tính tổng, hiệu, tích, thương giữa chúng và xuất kết quả.

**Mô tả Input và Output bài toán:**

- Input: Gồm 4 số nguyên lần lượt là tử và mẫu của 2 phân số a, b
- Output: Tổng, hiệu, tích, thương của 2 phân số a, b hoặc báo lỗi
- Solution: Mẫu của 1 trong 2 phân số bằng 0 thì ta báo lỗi. Nếu input đã hợp lệ thì ta tính giá trị tổng, hiệu, tích của 2 phân số a, b trước. Sau đó kiểm tra giá trị của phân số b, nếu phân số b bằng 0 thì thông báo không thể thực hiện được phép chia giữa 2 số a, b, vì chia cho 0 là không thể, nếu không thì ta tính thương của 2 số a, b.

**Flow Chart:**



Hình 1.5

**Code:** [Lab 1 bai 3](#)

**Kiểm thử các test case:**

```

D:\UIT\LTHDT\Lab_1>baitap3.exe
Nhap phan so thu nhat: 7 13
Nhap phan so thu hai: 8 17
Tong la: 223/221
Hieu la: 15/221
Tich la: 56/221
Thuong la: 119/104

D:\UIT\LTHDT\Lab_1>baitap3.exe
Nhap phan so thu nhat: 7 0
terminate called after throwing an instance of 'std::invalid_argument'
  what(): Mau so khong the bang 0

D:\UIT\LTHDT\Lab_1>baitap3.exe
Nhap phan so thu nhat: 6 9
Nhap phan so thu hai: 0 3
Tong la: 2/3
Hieu la: 2/3
Tich la: 0/1
terminate called after throwing an instance of 'std::invalid_argument'
  what(): Mau so khong the bang 0

```

Hình 1.6

## 1.4 Bài tập 4

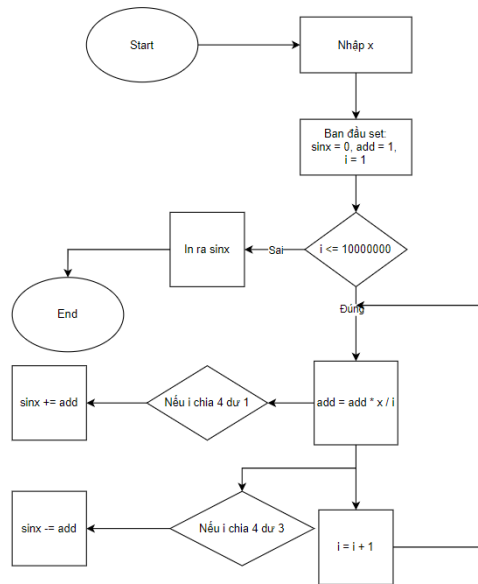
**Đề bài:**Lập chương trình tính  $\sin(x)$  với độ chính xác 0.00001 theo công thức:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

**Mô tả Input và Output bài toán:**

- Input: Số thực x
- Output:  $\sin(x)$  với độ chính xác 0.00001
- Solution: Ban đầu ta cho đặt biến số thực  $\sin x = 0.0$ , đặt biến add sẽ là lượng ta thêm vào hoặc bớt ra trong quá trình chạy vòng for. Cho chạy i từ 1 đến 10000000(số này đủ lớn để làm cho  $\sin(x)$  có độ chính xác cao), ta thực hiện tính toán trên biến add, duy trì  $\text{add} = \frac{x^i}{i!}$ . Theo công thức ta nhận thấy nếu i chia 4 dư 1 thì ta cộng add vào  $\sin x$ , nếu i chia 4 dư 3 thì ta trừ add khỏi  $\sin x$ . Với add, ta có thể tính dựa trên công thức  $\text{add}_i = \frac{\text{add}_{i-1} * x}{i}$ . Sau khi chạy xong vòng lặp thì thiết lập độ chính xác 5 chữ số sau dấu phẩy thập phân, rồi in ra  $\sin x$

**Flow Chart:**



Hình 1.8

Code: [Lab 1 bài 4](#)

Kiểm thử các test case:

```

D:\UIT\LTHDT\Lab_1>baitap4.exe
1
0.841471
D:\UIT\LTHDT\Lab_1>baitap4.exe
3.14
0.001593
D:\UIT\LTHDT\Lab_1>baitap4.exe
3.141592654
-0.000000
D:\UIT\LTHDT\Lab_1>
  
```

Hình 1.9

## 1.5 Bài tập 5

**Đề bài:** Viết chương trình nhập vào một ngày.

- Tìm ngày kế tiếp và xuất kết quả
- Tìm ngày trước ngày này và xuất kết quả.
- Tính xem ngày đó là ngày thứ bao nhiêu trong năm và xuất kết quả.

**Mô tả Input và Output bài toán:**

- Input: 3 số nguyên d, m, y lần lượt biểu thị ngày, tháng, năm hiện tại
- Output: Ngày trước đó, ngày kế tiếp và số thứ tự của ngày bao nhiêu trong năm.
- Solution: Kiểm tra xem ngày đó có hợp lệ hay không, nếu không thì dừng chương trình và thông báo. Nếu đầu vào hợp lệ. Ta cần xây dựng mảng thang[], với thang[i] là số ngày của tháng thứ i trong năm. Đối với tháng 2 thì số ngày còn phụ thuộc vào năm nhuận hay không. Ta xét ngày kế tiếp như sau:
  - Nếu d/m/y là ngày cuối trong tháng thì đáp án là  $1/m + 1/y$
  - Nếu d/m/y là ngày cuối trong năm(31/12) thì đáp án là  $1/1/y+1$
  - Không thuộc 2 trường hợp trên thì xuất ra  $d + 1/m/y$

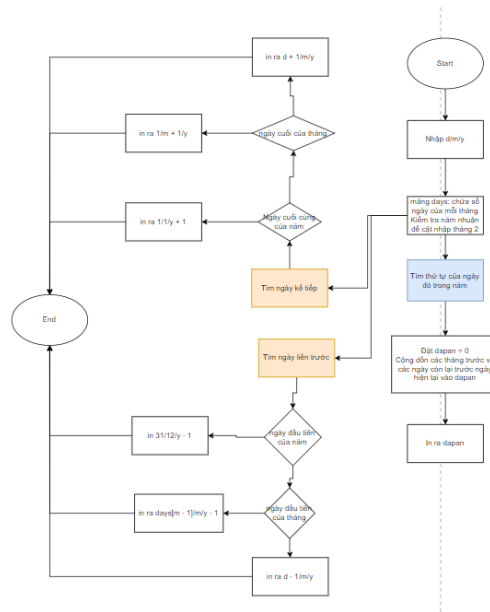
Xét ngày trước đó:

- Nếu d/m/y là ngày đầu trong tháng thì xuất ra  $\text{thang}[m - 1]/m - 1/y$
- Nếu d/m/y là ngày đầu trong năm (1/1) thì đáp án là  $31/12/y - 1$
- Không thuộc 2 trường hợp trên thì xuất ra  $d - 1/m/y$

Xét ngày thứ mấy trong năm:

- Tính tổng các ngày trước đó trong tháng cộng với số ngày của tất cả các tháng trước đó. Vd ngày 05/09/2005, đáp án sẽ là số ngày đã qua trong tháng(9) + số ngày của tất cả các tháng trước:(tháng 1, 2, 3, 4, 5, 6, 7, 8 trong năm 2005).
- Vậy ta sẽ có được vị trí, số thứ tự của ngày này trong năm

**Flow Chart:**



Hình 1.10

Code: Lab 1 bai 5

Kiểm thử các test case:

```

D:\UIT\LTHTD\Lab_1>baitap5.exe
Nhap ngay thang nam: 33 09 2005
terminate called after throwing an instance of 'std::invalid_argument'
what(): Ngay thang khong hop le

D:\UIT\LTHTD\Lab_1>baitap5.exe
Nhap ngay thang nam: 31 12 2005
Ngay trước là: 30/12/2005
Ngay sau là: 1/1/2006
Đây là ngày thứ: 365

D:\UIT\LTHTD\Lab_1>baitap5.exe
Nhap ngay thang nam: 31 7 2005
Ngay trước là: 30/7/2005
Ngay sau là: 1/8/2005
Đây là ngày thứ: 212

D:\UIT\LTHTD\Lab_1>baitap5.exe
Nhap ngay thang nam: 29 10 2005
Ngay trước là: 28/10/2005
Ngay sau là: 30/10/2005
Đây là ngày thứ: 302

D:\UIT\LTHTD\Lab_1>baitap5.exe
Nhap ngay thang nam: 1 1 2005
Ngay trước là: 31/12/2004
Ngay sau là: 2/1/2005
Đây là ngày thứ: 1
    
```

Hình 1.11

## 1.6 Bài tập 6

**Đề bài:** Cho hai mảng A và B chứa các số nguyên. Hãy xây dựng một chương trình để xác định xem có bao nhiêu lần mảng A xuất hiện liên tiếp dưới dạng một dãy con trong mảng B.

Cụ thể:

- Mảng A được coi là xuất hiện trong mảng B nếu tất cả các phần tử của A xuất hiện liên tiếp theo đúng thứ tự trong B.

- Các vị trí xuất hiện của A trong B có thể chồng lấn lên nhau. Yêu cầu:

1. Viết một thuật toán hiệu quả để đếm số lần xuất hiện của A trong B.

2. Đảm bảo rằng chương trình có thể xử lý các trường hợp mảng lớn và mảng con xuất hiện nhiều lần liên tiếp trong mảng cha.

3. Trong phần xuất kết quả, liệt kê tất cả các vị trí bắt đầu (chỉ số) của các lần xuất hiện của A trong B.

Ví dụ: Với  $A = [1, 2]$ ,  $B = [1, 2, 1, 2, 3, 1, 2]$ , mảng A xuất hiện 3 lần tại các chỉ số bắt đầu là 0, 2, và 5.

Đầu vào:

- Dòng đầu tiên chứa số phần tử của mảng A và mảng B.
- Dòng thứ hai chứa các phần tử của mảng A.
- Dòng thứ ba chứa các phần tử của mảng B. Đầu ra:
- Số lần mảng A xuất hiện trong mảng B.
- Danh sách các chỉ số bắt đầu của những lần xuất hiện.

**Mô tả Input và Output bài toán:**

- Input:

- Dòng đầu tiên chứa  $n_a$ ,  $n_b$  lần lượt là kích thước của mảng a và b
- Dòng thứ hai chứa  $n_a$  số nguyên biểu thị mảng a
- Dòng thứ ba chứa  $n_b$  số nguyên biểu thị mảng b

- Output:

- Dòng đầu tiên chứa k là số lượng đoạn con trong a giống với b
- Dòng thứ hai chứa k số nguyên lần lượt là vị trí bắt đầu của đoạn con giống nhau trong a và b

- Solution: Ta khởi tạo một biến  $cnt = 0$  để lưu số lượng và một vector  $idx[]$  để lưu vị trí. Chạy từ 0 đến  $a.size() - b.size()$  để kiểm tra đoạn con từ  $a[i, i + b.size() - 1]$  có giống với  $b$  không rồi update vào  $cnt$  và  $idx$ , Ta sẽ kiểm tra nhanh trong độ phức tạp  $O(1)$  bằng Hash. Hai đoạn con giống nhau sẽ có mã hash giống nhau. Ta sẽ hash 2 mảng  $a, b$  với  $base = 137$  hoặc là một số nguyên tố bất kì nào khác. Tương tự là số nguyên tố cho MOD, có thể  $MOD = 1e9 + 7$ . Xác suất sẽ là  $1/10^9$  cho 1 lần kiểm tra. Nếu muốn chính xác cao hơn ta có thể hash 2, 3, 4,... số nguyên tố khác nhau.

**Code:** [Lab 1 bài 6](#)

**Kiểm thử các test case:**

```
D:\UIT\LTHDT\Lab_1>baitap6.exe
11 3
1 2 3 1 2 1 2 3 1 2 3
1 2 3
3
1 6 9
D:\UIT\LTHDT\Lab_1>baitap6.exe
10 7
1 2 3 4 5 6 7 6 5 4
1 2 3 4 5 6 7
1
1
```

Hình 1.12

## 1.7 Bài tập 6<sub>2</sub>

**Đề bài:** Viết chương trình quản lý thông tin của nhiều học sinh, bao gồm họ tên, điểm toán, điểm văn, và thêm điểm ngoại ngữ. Chương trình cần thực hiện các chức năng sau:

1. Nhập thông tin cho  $n$  học sinh, bao gồm:
  - Họ tên (phải kiểm tra độ dài và định dạng, không chứa ký tự số).
  - Điểm Toán, Văn và Ngoại ngữ (phải là số thực trong khoảng từ 0 đến 10, nếu nhập sai phải yêu cầu nhập lại).
2. Tính điểm trung bình của mỗi học sinh.
 

Điểm trung bình =  $(2 * \text{Điểm Toán} + \text{Điểm Văn} + \text{Điểm Anh}) / 4$

Trong đó, điểm Toán có trọng số cao hơn.
3. Phân loại học sinh dựa trên điểm trung bình:



- Xuất sắc: Điểm trung bình từ 9 trở lên.
- Giỏi: Điểm trung bình từ 8 đến dưới 9.
- Khá: Điểm trung bình từ 6.5 đến dưới 8.
- Trung bình: Điểm trung bình từ 5 đến dưới 6.5.
- Yếu: Điểm trung bình dưới 5.

4. Tìm kiếm và xuất thông tin của học sinh có điểm trung bình cao nhất.

5. Tìm kiếm học sinh theo tên (cho phép tìm kiếm với từ khóa không phân biệt hoa thường và không cần nhập tên đầy đủ).

### Mô tả Input và Output bài toán:

- Input: Nhập thông tin của tất cả n học sinh. Nhập dữ liệu theo yêu cầu dựa trên các học sinh cần đưa vào.
- Output: Nếu đầu vào không hợp lệ, thông báo thông tin không hợp lệ. Nếu đầu vào hợp lệ thì in ra:
  - Điểm trung bình, phân loại học sinh.
  - Học sinh có điểm trung bình cao nhất và danh sách học sinh có điểm toán thấp nhất
  - Danh sách học sinh phù hợp với tên tìm kiếm
- Solution: Xây dựng struct HocSinh để lưu thông tin của 1 học sinh.
  - Từ dữ liệu Input để kiểm tra tính hợp lệ:
    - . Kiểm tra tên: Với các giá trị ASCII nằm giữa "A" và "Z" thì chúng hợp lệ.
    - Tìm kiếm trong n học sinh, học sinh nào có điểm trung bình cao nhất. Để biết danh sách học sinh có điểm Toán thấp nhất, ta duyệt qua n học sinh, duy trì biến, toanthapnhat là điểm Toán thấp nhất của các học sinh đến hiện tại. Sau đó ta duyệt lại lần nữa và kiểm tra điểm Toán của học sinh hiện tại có bằng với toanthapnhat hay không.
    - . Kiểm tra điểm: Với  $\min(\text{điểm}) < 0$  hoặc  $\max(\text{điểm}) > 0$  thì không hợp lệ.
    - Danh sách học sinh phù hợp với tên tìm kiếm sẽ được xây dựng bằng cách: . Ta sẽ so khớp tên cần tìm với tên của tất cả các bạn học sinh, tên của học sinh ta cần tìm khả năng cao nhất sẽ là tên mà ta so khớp được nhiều ký tự giống nhau nhất

**Code:** [Lab 1 bài 6<sub>2</sub>](#)

**Kiểm thử các test case:**

```
D:\UIT\LTHTD\Lab_1>baitap6_2.exe
Nhap so luong hoc sinh: 5
Nhap thong tin hoc sinh thu 1
Nhap ten hoc sinh: Van Tan
Nhap diem toan, van, anh: 4 4 5
Nhap thong tin hoc sinh thu 2
Nhap ten hoc sinh: Phuong Nam
Nhap diem toan, van, anh: 5 5 6
Nhap thong tin hoc sinh thu 3
Nhap ten hoc sinh: Minh Huy
Nhap diem toan, van, anh: 7 7 8
Nhap thong tin hoc sinh thu 4
Nhap ten hoc sinh: Dinh Phuoc
Nhap diem toan, van, anh: 8 8 9
Nhap thong tin hoc sinh thu 5
Nhap ten hoc sinh: Long Vu
Nhap diem toan, van, anh: 10 10 10
Thong tin hoc sinh thu 1
Ten hoc sinh: Van Tan
Diem toan: 4, Diem van: 4, Diem anh: 5
Diem trung binh: 4.25
Phan loai: Yeu
Thong tin hoc sinh thu 2
Ten hoc sinh: Phuong Nam
Diem toan: 5, Diem van: 5, Diem anh: 6
Diem trung binh: 5.25
Phan loai: Trung binh
Thong tin hoc sinh thu 3
Ten hoc sinh: Minh Huy
Diem toan: 7, Diem van: 7, Diem anh: 8
Diem trung binh: 7.25
Phan loai: Kha
Thong tin hoc sinh thu 4
Ten hoc sinh: Dinh Phuoc
Diem toan: 8, Diem van: 8, Diem anh: 9
Diem trung binh: 8.25
Phan loai: Gioi
Thong tin hoc sinh thu 5
Ten hoc sinh: Long Vu
Diem toan: 10, Diem van: 10, Diem anh: 10
Diem trung binh: 10
Phan loai: Xuat sac
Hoc sinh co DTB cao nhat la: Long Vu
Nhap hoc sinh can tim kiem:
```

Hình 1.13

```

D:\UIT\LTHDT\Lab_1>baitap6_2.exe
Nhap so luong hoc sinh: 2
Nhap thong tin hoc sinh thu 1
Nhap ten hoc sinh: @@@
Nhap diem toan, van, anh: -1 -1 -1
terminate called after throwing an instance of 'char const*'

D:\UIT\LTHDT\Lab_1>baitap6_2.exe
Nhap so luong hoc sinh: 2
Nhap thong tin hoc sinh thu 1
Nhap ten hoc sinh: Van Tan
Nhap diem toan, van, anh: 5 5 5
Nhap thong tin hoc sinh thu 2
Nhap ten hoc sinh: Phuong Nam
Nhap diem toan, van, anh: 10 10 10
Thong tin hoc sinh thu 1
Ten hoc sinh: Van Tan
Diem toan: 5, Diem van: 5, Diem anh: 5
Diem trung binh: 5
Phan loai: Trung binh
Thong tin hoc sinh thu 2
Ten hoc sinh: Phuong Nam
Diem toan: 10, Diem van: 10, Diem anh: 10
Diem trung binh: 10
Phan loai: Xuat sac
Hoc sinh co DTB cao nhat la: Phuong Nam
Nhap hoc sinh can tim kiem: Phu
Hoc sinh co ten phu hop nhat voi Phu la: Phuong Nam
Danh sach hoc sinh co diem toan thap nhat la:
Van Tan

```

Hình 1.14

## 1.8 Bài tập 7

**Đề bài:** Hãy xây dựng một chương trình quản lý thông tin chuyến bay. Mỗi chuyến bay bao gồm các thông tin sau:

- Mã chuyến bay: Chuỗi ký tự dài tối đa 5 ký tự, không được chứa khoảng trắng hay ký tự đặc biệt.
  - Ngày bay: Dữ liệu ngày khởi hành, yêu cầu kiểm tra tính hợp lệ của ngày nhập (ngày phải tồn tại và hợp lệ theo lịch).
  - Giờ bay: Dữ liệu giờ khởi hành, được nhập theo định dạng 24 giờ (từ 00:00 đến 23:59).
  - Nơi đi: Chuỗi ký tự dài tối đa 20 ký tự. Phải là tên địa danh hợp lệ, không được chứa số hoặc ký tự đặc biệt.
  - Nơi đến: Chuỗi ký tự dài tối đa 20 ký tự. Tương tự như nơi đi, yêu cầu kiểm tra định dạng
- Yêu cầu:**

1. Viết các hàm nhập và xuất dữ liệu cho một chuyến bay.
  - Kiểm tra và báo lỗi nếu người dùng nhập sai các định dạng yêu cầu.
2. Viết thêm các chức năng sau:
  - Tìm kiếm chuyến bay theo mã chuyến bay, nơi đi hoặc nơi đến.

- Sắp xếp danh sách chuyến bay theo thứ tự ngày và giờ khởi hành (từ sớm đến muộn).

3. Hiển thị danh sách tất cả các chuyến bay khởi hành từ một nơi cụ thể trong một ngày được chỉ định.

4. Đếm số lượng chuyến bay từ một nơi đi đến một nơi đến nhất định.

Ví dụ chi tiết:

Giả sử bạn có danh sách các chuyến bay với thông tin đầy đủ, chương trình cần:

- Kiểm tra xem mã chuyến bay “VN123” có tồn tại không.
- Sắp xếp danh sách các chuyến bay theo ngày giờ khởi hành.
- Liệt kê tất cả các chuyến bay khởi hành từ “Hà Nội” vào ngày 01/01/2024.
- Đếm số chuyến bay từ “Hà Nội” đến “TP. Hồ Chí Minh”

### **Mô tả Input và Output bài toán:**

- Input: Nếu đầu vào không hợp lệ, thông báo thông tin không hợp lệ. Nếu đầu vào hợp lệ thì ta có dữ liệu vào:
  - Số lượng chuyến bay.
  - Thông tin chi tiết cho mỗi chuyến bay bao gồm mã chuyến bay, ngày bay, giờ bay, nơi đi, và nơi đến.
  - Mã chuyến bay: Xâu gồm tối đa 5 kí tự, Không chứa kí tự đặc biệt, dấu cách
  - Ngày bay: Khởi hành, tính hợp lệ (ngày phải tồn tại trên lịch)
  - Giờ bay: Là một định dạng 24 giờ(00:00 đến 23:59).
  - Nơi đi, nơi khởi hành: Xâu dài tối đa 20 ký tự. Phải là tên địa điểm có tồn tại trên bản đồ, không chứa các kí tự ngoài lề hay đặc biệt.
  - Nơi đến: Xâu gồm tối đa 20 kí tự. Kiểm tra định dạng xâu giống nơi đi.
- Output:
  - Danh sách chuyến bay đã sắp xếp theo ngày giờ khởi hành.
  - Kết quả tìm kiếm và các báo cáo theo yêu cầu.
  - Mã chuyến bay, nơi đi, nơi đến.
  - Chuyến bay được sắp xếp theo giờ bay và thứ tự bay.
  - Đưa ra các chuyến bay xuất phát trong ngày nêu được tìm kiếm ngày cụ thể.
  - Giữa 2 địa điểm có bao nhiêu chuyến bay tính trên thời gian tổng thể

- Solution: Lưu thông tin của 1 chuyến bay vào struct
  - Kiểm tra:
    - . Mã chuyến bay, nơi đi, nơi đến, theo từng kí tự
    - . Định dạng ngày và giờ, có tồn tại hay không
    - . Xây 1 hàm trong struct để kiểm tra, tìm kiếm chuyến bay theo nơi đi, nơi đến.
  - Ta dùng hàm sort trong C++, và ta chỉ việc xây dựng hàm so sánh cho 2 chuyến bay khác nhau.
  - Đối với việc lấy chuyến bay xuất phát từ 1 địa điểm và đến một địa điểm khác trong ngày chỉ định. Ta duyệt hết tất cả các chuyến bay rồi kiểm tra, phân phối.

### Lab 1 bài 7

#### Kiểm thử các test case:

```
D:\UIT\LTHDT\Lab_1>baitap7.exe
Nhap so luong chuyen bay: 3
Nhap chuyen bay thu 1
Nhap ma chuyen bay: TAN00
Nhap noi den: DaNang
Nhap noi di: Hue
Nhap ngay: 01/10/2024
Nhap gio: 08:40
Nhap chuyen bay thu 2
Nhap ma chuyen bay: TAN01
Nhap noi den: HCM
Nhap noi di: DaNang
Nhap ngay: 10/10/2024
Nhap gio: 04:50
Nhap chuyen bay thu 3
Nhap ma chuyen bay: TAN02
Nhap noi den: Hue
Nhap noi di: HCM
Nhap ngay: 05/09/2005
Nhap gio: 10:14
Danh sach ma chuyen bay sap xep theo gio khoi hanh:
TAN02: 05/09/2005
TAN00: 01/10/2024
TAN01: 10/10/2024
Nhap ma chuyen bay, noi den hoac noi di can tim: HCM
Ma chuyen bay: TAN02
Noi den: Hue - Noi di: HCM
Ngay: 05/09/2005 - Gio: 10:14
Ma chuyen bay: TAN01
Noi den: HCM - Noi di: DaNang
Ngay: 10/10/2024 - Gio: 04:50
Tim chuyen bay di tu noi di va ngay khoi hanh
Nhap noi di: Hue
Nhap ngay: 01/10/2024
Danh sach chuyen bay co noi di la Hue va ngay la 01/10/2024
Ma chuyen bay: TAN00
Noi den: DaNang - Noi di: Hue
Ngay: 01/10/2024 - Gio: 08:40
Tim chuyen bay di tu noi di den noi den
Nhap noi di: HCM
Nhap noi den: Hue
Ma chuyen bay: TAN02
Noi den: Hue - Noi di: HCM
Ngay: 05/09/2005 - Gio: 10:14
```

Hình 1.15

## 1.9 Bài tập 8

**Đề bài:** Xây dựng chương trình quản lý sổ tiết kiệm với các yêu cầu mở rộng. Mỗi sổ tiết kiệm bao gồm các thông tin:

- Mã số: Chuỗi ký tự dài tối đa 5 ký tự (không chứa khoảng trắng hoặc ký tự đặc biệt).
- Loại tiết kiệm: Chuỗi ký tự dài tối đa 10 ký tự (ví dụ: ngắn hạn, dài hạn, không kỳ hạn...).
- Họ tên khách hàng: Chuỗi ký tự dài tối đa 30 ký tự (không chứa số hoặc ký tự đặc biệt).
- Chứng minh nhân dân: Số nguyên 4 byte (kiểm tra độ dài và định dạng hợp lệ).
- Ngày mở sổ: Kiểu dữ liệu ngày, phải hợp lệ (kiểm tra ngày, tháng, năm).
- Số tiền gửi: Số thực, phải là số dương.

Yêu cầu:

1. Khai báo kiểu dữ liệu để biểu diễn thông tin của một sổ tiết kiệm.
2. Viết các hàm nhập dữ liệu với các yêu cầu:
  - Kiểm tra độ hợp lệ của mã số (độ dài tối đa và không chứa ký tự đặc biệt).
  - Kiểm tra CMND phải có đúng 9 hoặc 12 chữ số.
  - Kiểm tra ngày mở sổ hợp lệ (ngày phải tồn tại theo lịch).
  - Số tiền gửi phải là số dương, nhập sai phải yêu cầu nhập lại.
3. Viết hàm xuất dữ liệu sổ tiết kiệm.
4. Chức năng nâng cao:
  - Cập nhật lãi suất: Người dùng có thể nhập vào lãi suất (tính theo /năm) và chương trình sẽ tính toán số tiền lãi tương ứng theo loại tiết kiệm.
    - Loại tiết kiệm ngắn hạn: áp dụng lãi suất trong khoảng thời gian nhỏ hơn hoặc bằng 6 tháng.
    - Loại tiết kiệm dài hạn: áp dụng lãi suất cho kỳ hạn từ 6 tháng trở lên.
  - Tính toán tiền lãi dựa trên thời gian gửi (ngày mở sổ so với ngày hiện tại).
  - Chức năng rút tiền: Người dùng có thể chọn rút một phần hoặc toàn bộ số tiền trong sổ tiết kiệm. Nếu rút trước hạn, chương trình sẽ cảnh báo và tính lại lãi suất với lãi suất thấp hơn (ví dụ: 0.5
5. Tìm kiếm và liệt kê:
  - Tìm kiếm sổ tiết kiệm theo CMND hoặc mã số.
  - Liệt kê tất cả các sổ tiết kiệm mở trong một khoảng thời gian xác định.
6. Sắp xếp danh sách các sổ tiết kiệm:
  - Sắp xếp danh sách theo số tiền gửi giảm dần.
  - Sắp xếp danh sách theo ngày mở sổ tăng dần.

### Mô tả Input và Output bài toán:

- Input: Nếu đầu vào không hợp lệ thì thông báo thông tin không hợp lệ. Nếu thông tin hợp lệ thì có dữ liệu:
  - Thông tin của n sổ tiết kiệm: bao gồm:
    - Mã số
    - Loại tiết kiệm
    - Họ tên khách hàng
    - Chứng minh nhân dân
    - Ngày mở sổ
    - Số tiền gửi
- Output:
  - Thông tin về sổ tiết kiệm
  - Theo CMND liệt kê thông tin sổ tiết kiệm(in ra mã số)
  - Tất cả thông tin về STK
  - Sort theo số tiền gửi giảm hoặc tăng.
  - Chức năng: cật nhập số dư sau khi rút tiền
- Solution: Sử dụng struct để xử lý:
  - Kiểm tra tính hợp lệ:
    - . Kiểm tra mã số, loại tiết kiệm, họ tên, CMND, kiểm tra từng kí tự và độ dài
    - . Kiểm tra ngày mở sổ như bài 7
  - Cật nhập số tiền
  - Tìm kiếm theo CMND hoặc mã số ngân hàng
  - Viết hàm để sort lại theo ngày mở sổ tăng dần hoặc giảm dần

**Code:** [Lab 1 bai 8](#)

**Kiểm thử các test case:**

```
D:\UIT\LTHDT\Lab_1>baitap8.exe
Nhap so luong so tiet kiem: 3
Nhap thong tin so tiet kiem thu 1
Nhap ma so: aa000
Nhap loai tiet kiem: Ky han
Nhap ten: Long Vu
Nhap CMND: 000000000
Nhap ngay mo so: 05/09/2005
Nhap so tien: 1000
Nhap thong tin so tiet kiem thu 2
Nhap ma so: aa001
Nhap loai tiet kiem: Khong ky han
Nhap ten: Van Tan
Nhap CMND: 123456789
Nhap ngay mo so: 30/09/2005
Nhap so tien: 100
Nhap thong tin so tiet kiem thu 3
Nhap ma so: aa002
Nhap loai tiet kiem: Khong ky han
Nhap ten: Minh Huy
Nhap CMND: 999999999
Nhap ngay mo so: 30/09/2024
Nhap so tien: 1000
Danh sach so tiet kiem:
Ma so: aa000
Loai tiet kiem: Ky han
Ten khách hàng: Long Vu
CMND: 000000000
Ngày mở sổ: 05/09/2005
Số tiền: 6115.91

Ma so: aa001
Loai tiet kiem: Khong ky han
Ten khách hàng: Van Tan
CMND: 123456789
Ngày mở sổ: 30/09/2005
Số tiền: 100

Ma so: aa002
Loai tiet kiem: Khong ky han
Ten khách hàng: Minh Huy
CMND: 999999999
Ngày mở sổ: 30/09/2024
Số tiền: 1000

Nhap ma so hoac CMND can tim: 999999999
Cac STK thoa man la:
Ma so: aa002
CMND: 999999999

STK sắp xếp theo số tiền giảm dần:
Ma so: aa000
Số tiền: 6115.91
```

Hình 1.16