

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

Đồng Quốc Thắng

MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
BÀI TẬP THỰC HÀNH 1

OBJECT-ORIENTED PROGRAMMING
EXERCISE 1

CỬ 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

Đồng Quốc Thắng = 23521421

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

Mục lục

| | |
|---|-----------|
| Mục lục | i |
| 1 Bài 1 | 1 |
| 1.1 Flow Chart | 1 |
| 1.2 Input, Output và Solution | 1 |
| 1.3 Code | 2 |
| 1.4 Testcase | 3 |
| 2 Bài 2 | 5 |
| 2.1 Flow Chart | 5 |
| 2.2 Input, Output và Solution | 5 |
| 2.3 Code | 5 |
| 2.4 Testcase | 7 |
| 3 Bài 3 | 9 |
| 3.1 Flow Chart | 9 |
| 3.2 Input, Output và Solution | 9 |
| 3.3 Code | 9 |
| 3.4 Testcase | 12 |
| 4 Bài 4 | 13 |
| 4.1 Flow Chart | 13 |
| 4.2 Input, Output và Solution | 14 |
| 4.3 Code | 14 |
| 4.4 Testcase | 15 |
| 5 Bài 5 | 17 |
| 5.1 Flow Chart | 18 |
| 5.2 Input, Output và Solution | 19 |

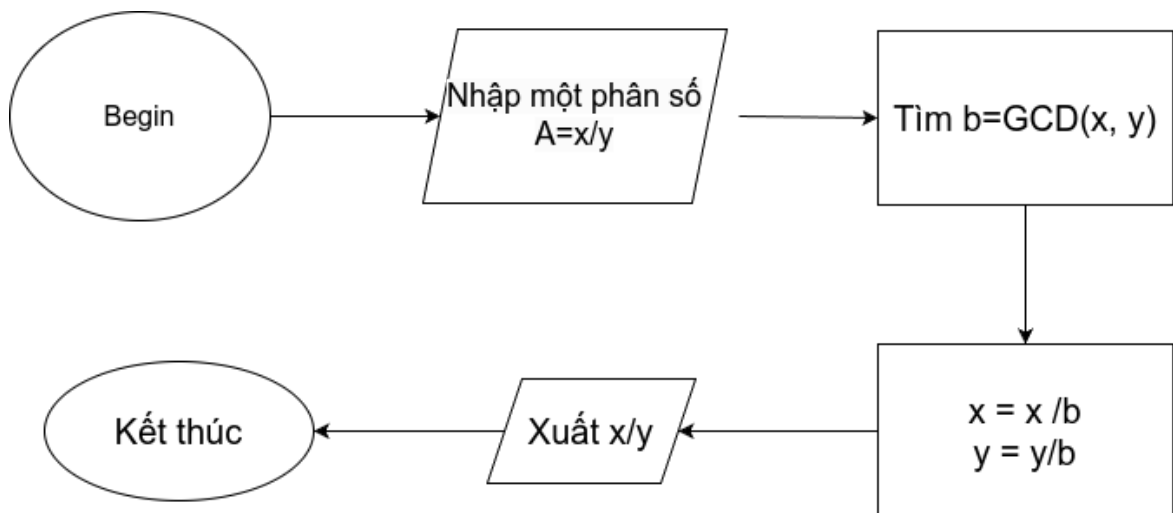
| | | |
|----------|-------------------------------------|-----------|
| 5.3 | Code | 19 |
| 5.4 | Testcase | 21 |
| 6 | Bài 6.1 | 23 |
| 6.1 | Flow Chart | 24 |
| 6.2 | Input, Output và Solution | 24 |
| 6.3 | Code | 24 |
| 6.4 | Testcase | 28 |
| 7 | Bài 6.2 | 29 |
| 7.1 | Flow Chart | 30 |
| 7.2 | Input, Output và Solution | 30 |
| 7.3 | Code | 31 |
| 7.4 | Testcase | 31 |
| 8 | Bài 7 | 35 |
| 8.1 | Flow Chart | 36 |
| 8.2 | Input, Output và Solution | 36 |
| 8.3 | Code | 37 |
| 8.4 | Testcase | 37 |
| 9 | Bài 8 | 41 |
| 9.1 | Flow Chart | 42 |
| 9.2 | Input, Output và Solution | 42 |
| 9.3 | Code | 43 |
| 9.4 | Testcase | 43 |

Chương 1

Bài 1

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

1.1 Flow Chart



1.2 Input, Output và Solution

Input: 2 số nguyên x, y là tử số và mẫu số của phân số đó

Output: Một phân số tối giản(sẽ output ra 0 nếu như tử số là 0, output ra 1 nếu tử số bằng mẫu số và output ra -1 nếu độ lớn tử số bằng mẫu số và khác dấu)

Solution: Ta lần lượt chia tử số và mẫu số cho ước chung lớn nhất của nó, ngoài ra còn xét 4 trường hợp là tử số = 0, mẫu số = 0, và độ lớn tử số và mẫu số bằng nhau nhưng cùng dấu/khác dấu

1.3 Code

```
#include <bits/stdc++.h>
using namespace std;

struct PhanSo {
    int Tu_so;
    int Mau_so;
};

void RutGon(PhanSo &ps) {
    int gcd = __gcd(ps.Tu_so, ps.Mau_so);
    ps.Tu_so = ps.Tu_so / gcd;
    ps.Mau_so = ps.Mau_so / gcd;
};

void Out(PhanSo ps) {
    if (abs(ps.Tu_so) == abs(ps.Mau_so) && abs(ps.Tu_so) == 1) {
        if (ps.Tu_so * ps.Mau_so > 0) {
            cout << 1 << endl;
        } else {
            cout << -1 << endl;
        }
    } else if (ps.Tu_so == 0) {
        cout << 0 << endl;
    } else {
        cout << ps.Tu_so << "/" << ps.Mau_so << endl;
    }
}

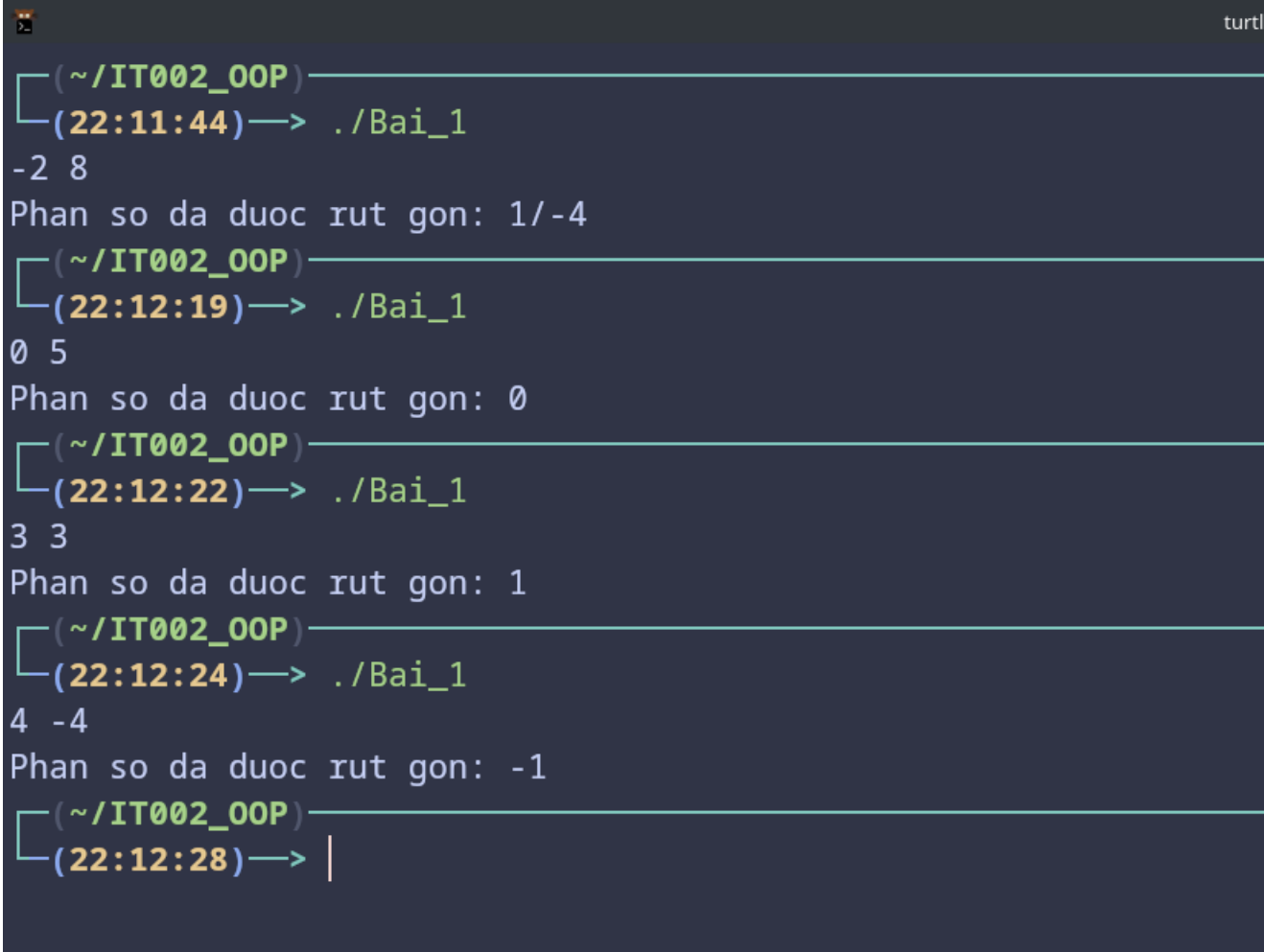
int main() {
    PhanSo ps;
    cin >> ps.Tu_so >> ps.Mau_so;

    if (ps.Mau_so == 0) {
```



```
        cout << "Mau so bang 0" << endl;  
    }  
    RutGon(ps);  
    cout << "Phan so da duoc rut gon: ";  
    Out(ps);  
    return 0;  
}
```

1.4 Testcase



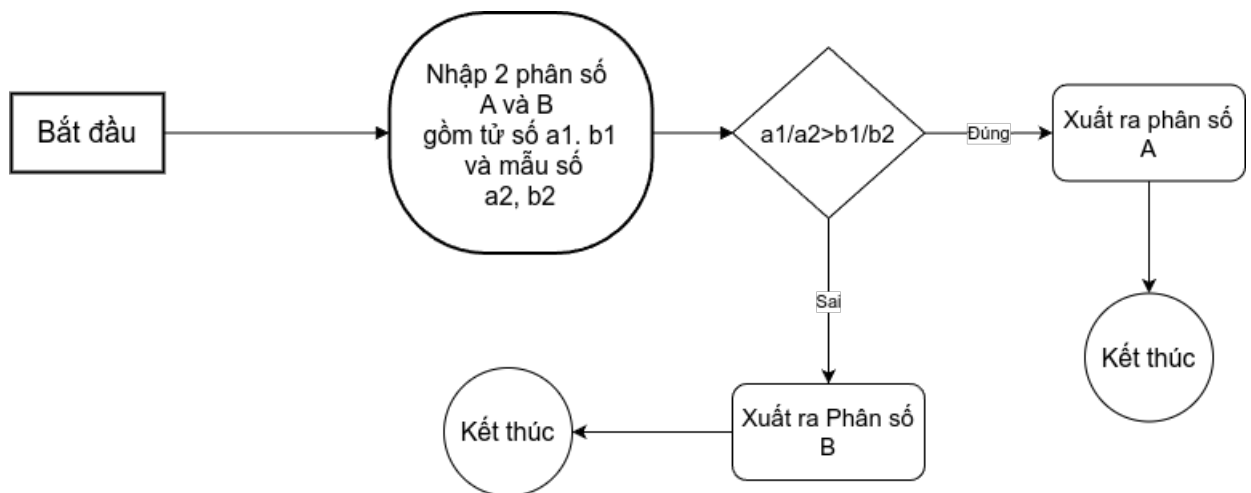
```
(~/IT002_00P) —  
(22:11:44) —> ./Bai_1  
-2 8  
Phan so da duoc rut gon: 1/-4  
(~/IT002_00P) —  
(22:12:19) —> ./Bai_1  
0 5  
Phan so da duoc rut gon: 0  
(~/IT002_00P) —  
(22:12:22) —> ./Bai_1  
3 3  
Phan so da duoc rut gon: 1  
(~/IT002_00P) —  
(22:12:24) —> ./Bai_1  
4 -4  
Phan so da duoc rut gon: -1  
(~/IT002_00P) —  
(22:12:28) —> |
```


Chương 2

Bài 2

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

2.1 Flow Chart



2.2 Input, Output và Solution

Input: Nhập vào 2 phân số A và B

Output: Xuất ra phân số lớn hơn

Solution: Nếu $A > B$ thì in ra A, nếu $A \leq B$ thì in ra B. Báo lỗi nếu mẫu số bằng 0

2.3 Code

```
#include <bits/stdc++.h>
using namespace std;

struct PhanSo {
    int Tu_so;
    int Mau_so;
};

int main() {
    PhanSo ps1, ps2;

    cin >> ps1.Tu_so >> ps1.Mau_so;
    cin >> ps2.Tu_so >> ps2.Mau_so;

    float x1 = ps1.Tu_so / ps1.Mau_so;
    float x2 = ps2.Tu_so / ps2.Mau_so;

    if (ps1.Mau_so == 0 || ps2.Mau_so == 0) {
        cout << "Mau so bang 0" << endl;
        return 0;
    }

    if (x1 > x2) {
        cout << "Phan so lon hon: " << ps1.Tu_so << "/" << ps1.Mau_so << endl;
    } else {
        cout << "Phan so lon hon: " << ps2.Tu_so << "/" << ps2.Mau_so << endl;
    }

    return 0;
}
```

2.4 Testcase

```
(~/IT002_OOP)
(12:34:02)→ ./Bai_2
1 2 3 4
Phan so lon hon: 3/4
(~/IT002_OOP)
(12:34:07)→ ./Bai_2
-3 4 -1 2
Phan so lon hon: -1/2
(~/IT002_OOP)
(12:34:12)→ ./Bai_2
0 1 2 3
Phan so lon hon: 2/3
(~/IT002_OOP)
(12:34:17)→ ./Bai_2
1 0 1 1
Mau so bang 0
(~/IT002_OOP)
(12:34:20)→ |
```


Chương 3

Bài 3

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

3.1 Flow Chart



3.2 Input, Output và Solution

Input: Nhập vào 2 phân số A và B

Output: In ra tổng, hiệu, thương, tích của 2 phân số này

Solution: Sử dụng các phép biến đổi cơ bản để tính tổng, hiệu, thương, tích của 2 phân số đã cho. Báo lỗi nếu mẫu số được nhập vào là 0.

3.3 Code

```
#include <bits/stdc++.h>
using namespace std;
```

```
struct PhanSo {
    int Tu_so;
    int Mau_so;
};

void RutGon(PhanSo &ps) {
    int gcd = __gcd(ps.Tu_so, ps.Mau_so);
    ps.Tu_so = ps.Tu_so / gcd;
    ps.Mau_so = ps.Mau_so / gcd;
}

void Out(PhanSo ps) {
    if (abs(ps.Tu_so) == abs(ps.Mau_so) && abs(ps.Tu_so) == 1) {
        if (ps.Tu_so * ps.Mau_so > 0) {
            cout << 1 << endl;
        } else {
            cout << -1 << endl;
        }
    } else if (ps.Tu_so == 0) {
        cout << 0 << endl;
    } else {
        cout << ps.Tu_so << "/" << ps.Mau_so << endl;
    }
}

int main() {
    PhanSo ps1, ps2;
    PhanSo temp;
    cin >> ps1.Tu_so >> ps1.Mau_so;
    cin >> ps2.Tu_so >> ps2.Mau_so;

    if (ps1.Mau_so == 0 || ps2.Mau_so == 0) {
        cout << "Mau so bang 0" << endl;
        return 0;
    }
}
```



```
temp.Tu_so = ps1.Tu_so * ps2.Mau_so + ps2.Tu_so * ps1.Mau_so;
temp.Mau_so = ps1.Mau_so * ps2.Mau_so;
RutGon(temp);
cout << "Tong 2 phan so vua nhap la: ";
Out(temp);

temp.Tu_so = ps1.Tu_so * ps2.Mau_so - ps2.Tu_so * ps1.Mau_so;
temp.Mau_so = ps1.Mau_so * ps2.Mau_so;
RutGon(temp);
cout << "Hieu 2 phan so vua nhap la: ";
Out(temp);

temp.Tu_so = ps1.Tu_so * ps2.Tu_so;
temp.Mau_so = ps1.Mau_so * ps2.Mau_so;
RutGon(temp);
cout << "Tich 2 phan so vua nhap la: ";
Out(temp);

temp.Tu_so = ps1.Tu_so * ps2.Mau_so;
temp.Mau_so = ps1.Mau_so * ps2.Tu_so;
RutGon(temp);
cout << "Thuong 2 phan so vua nhap la: ";
Out(temp);

return 0;
}
```

3.4 Testcase

```

[~/IT002_OOP]—
[ (12:37:46) —> ./Bai_3
1 2 5 9
Tong 2 phan so vua nhap la: 19/18
Hieu 2 phan so vua nhap la: 1/-18
Tich 2 phan so vua nhap la: 5/18
Thuong 2 phan so vua nhap la: 9/10
[~/IT002_OOP]—
[ (12:37:49) —> ./Bai_3
4 7 -1 2
Tong 2 phan so vua nhap la: 1/14
Hieu 2 phan so vua nhap la: 15/14
Tich 2 phan so vua nhap la: -2/7
Thuong 2 phan so vua nhap la: 8/-7
[~/IT002_OOP]—
[ (12:37:55) —> ./Bai_3
33 99 11 88
Tong 2 phan so vua nhap la: 11/24
Hieu 2 phan so vua nhap la: 5/24
Tich 2 phan so vua nhap la: 1/24
Thuong 2 phan so vua nhap la: 8/3
[~/IT002_OOP]—
[ (12:38:00) —> ./Bai_3
1 0 2 3
Mau so bang 0
[~/IT002_OOP]—
[ (12:38:05) —> |

```

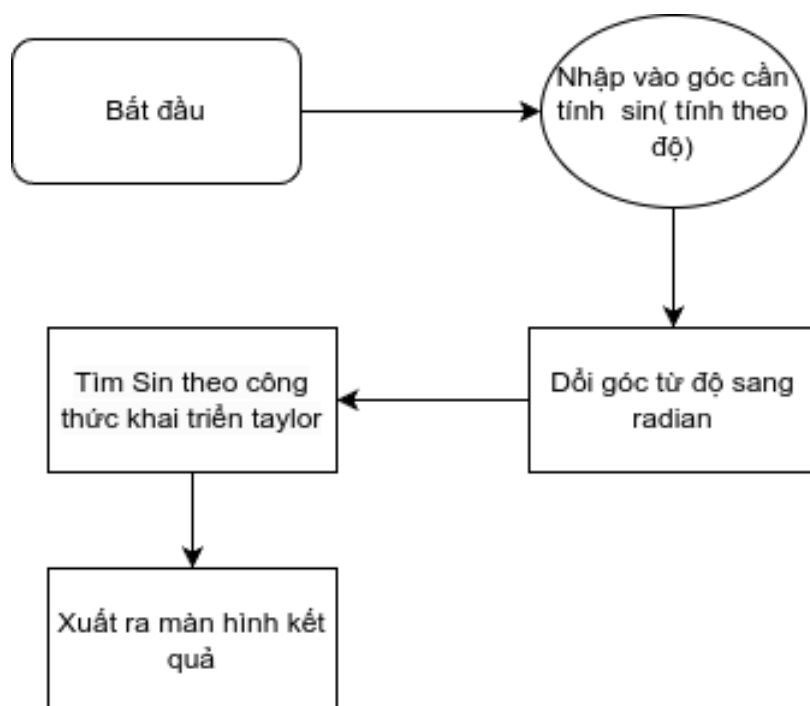
Chương 4

Bài 4

Bài 4: Lập chương trình tính $\sin(x)$ với độ chính xác 0.00001 theo công thức

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

4.1 Flow Chart



4.2 Input, Output và Solution

Input: Nhập vào một góc(tính theo độ)

Output: Xuất ra giá trị sin của góc đó.

Solution: Đổi góc từ độ sang radian, Sử dụng công thức khai triển Taylor trong đề bài. Nếu góc có dạng $x + k * 2 * \pi$ thì giảm đi $k * 2 * \pi$

4.3 Code

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    double x, sinx;
    cin >> x;
    double pi = 2 * acos(0.0);
    x = x - floor(x/360)*360;
    x = x * pi / 180;
    sinx = x;
    int sign = 1;
    double powx = x;
    double sqrx = x * x;
    long long int facx = 1;
    for (int i = 1; i < 10; i++) {
        powx = powx * sqrx;
        sign = sign * -1;
        facx = facx * (2*i + 1) * (2*i);
        sinx += (sign * powx) / facx;
    }

    cout << "Gia tri sin can tim la: " << sinx << endl;

    return 0;
}
```

4.4 Testcase

```
(~/IT002_OOP)——  
(12:52:15)→ ./Bai_4  
45  
Gia tri sin can tim la: 0.707107  
(~/IT002_OOP)——  
(12:52:22)→ ./Bai_4  
12  
Gia tri sin can tim la: 0.207912  
(~/IT002_OOP)——  
(12:52:25)→ ./Bai_4  
1  
Gia tri sin can tim la: 0.0174524  
(~/IT002_OOP)——  
(12:52:28)→ ./Bai_4  
0  
Gia tri sin can tim la: 0  
(~/IT002_OOP)——  
(12:52:29)→ ./Bai_4  
90  
Gia tri sin can tim la: 1  
(~/IT002_OOP)——  
(12:52:32)→ ./Bai_4  
-50  
Gia tri sin can tim la: -0.766091  
(~/IT002_OOP)——  
(12:52:42)→ ./Bai_4  
270  
Gia tri sin can tim la: -1  
(~/IT002_OOP)——  
(12:52:44)→ |
```

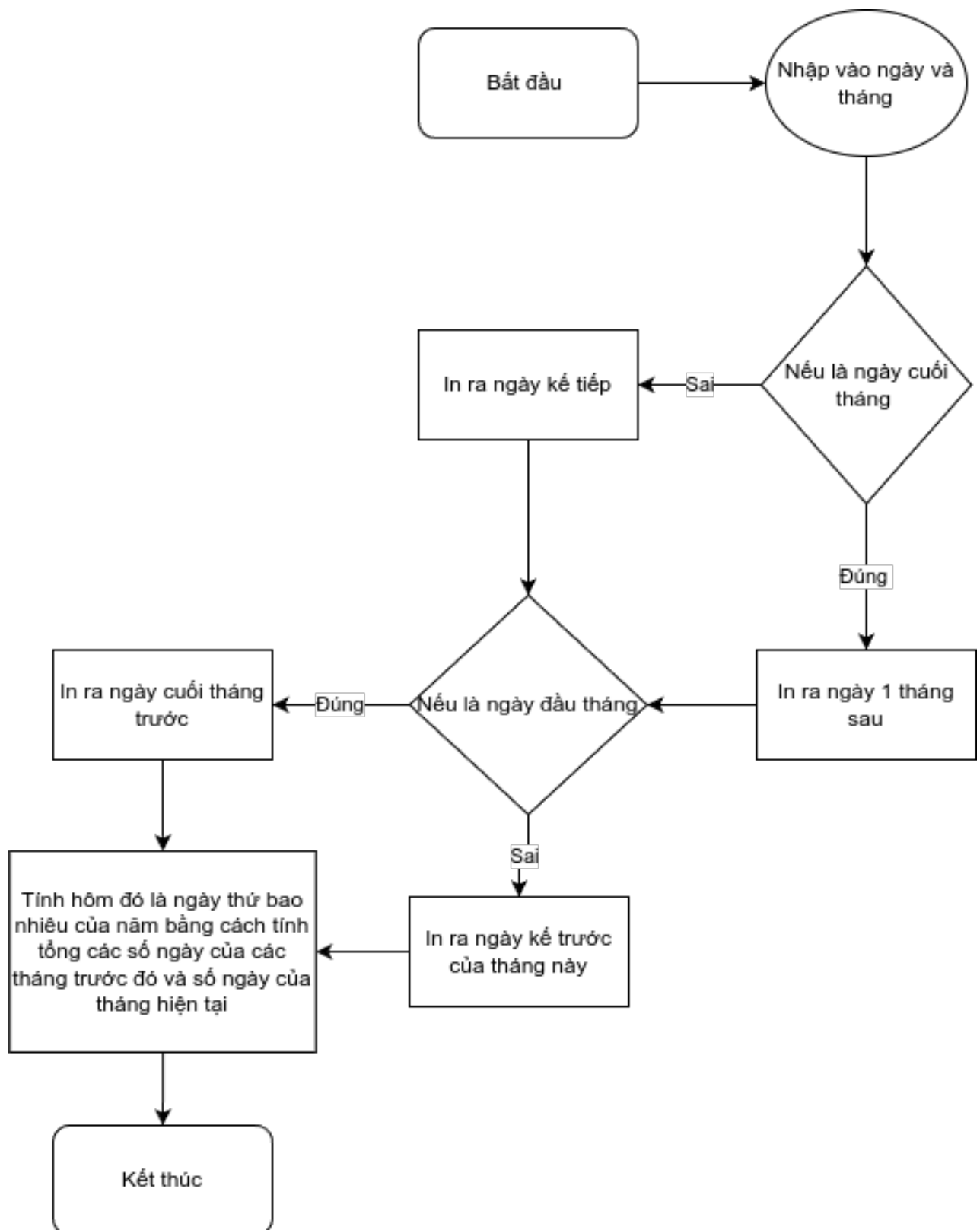

Chương 5

Bài 5

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.

5.1 Flow Chart



5.2 Input, Output và Solution

Input: Ngày và tháng

Output: Ngày trước đó, sau đó và tính ngày ấy là ngày thứ bao nhiêu trong năm

Solution: Xét các trường hợp đặc biệt:

- Nếu ngày đó là ngày đầu tháng thì ngày liền trước phải là ngày cuối tháng trước
- Nếu ngày đó là ngày cuối tháng thì ngày liền sau là ngày đầu tháng sau

Tính số ngày trong năm bằng cách tính tổng tất cả các ngày của các tháng trước và những ngày đã qua của tháng hiện tại.

5.3 Code

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main() {
```

```
    int days_of_month[13] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30,
```

```
    int d, m;
```

```
    cin >> d >> m;
```

```
    if (d == days_of_month[m-1]) {
```

```
        cout << "Ngày kế tiếp là ngày 1" << endl;
```

```
    } else {
```

```
        cout << "Ngày kế tiếp là ngày " << d+1 << endl;
```

```
    }
```

```
    if (d == 1) {
```

```
        if (m == 1) {
```

```
            cout << "Ngày trước là ngày 31" << endl;
```

```
        } else {
```

```
            cout << "Ngày trước là ngày " << days_of_month[m-1] << endl;
```

```
        }
```

```
    } else {
```

```
        cout << "Ngày trước là ngày " << d-1 << endl;
```

```
}

int num_of_days = 0;
for (int i = 0; i < m - 1; i++) {
    num_of_days += days_of_month[i];
}
num_of_days += d;
cout << "Ngày " << d << " thang " << m << " la ngay thu " << num_of_day
return 0;
}
```

5.4 Testcase

```

[~/IT002_OOP]—
(13:21:39)→ ./Bai_5
31 1
Ngày ke tiep la ngay 1
Ngày truoc la ngay 30
Ngày 31 thang 1 la ngay thu 31 trong nam
[~/IT002_OOP]—
(13:21:45)→ ./Bai_5
5 3
Ngày ke tiep la ngay 6
Ngày truoc la ngay 4
Ngày 5 thang 3 la ngay thu 64 trong nam
[~/IT002_OOP]—
(13:21:54)→ ./Bai_5
1 9
Ngày ke tiep la ngay 2
Ngày truoc la ngay 30
Ngày 1 thang 9 la ngay thu 244 trong nam
[~/IT002_OOP]—
(13:22:03)→ ./Bai_5
1 12
Ngày ke tiep la ngay 2
Ngày truoc la ngay 31
Ngày 1 thang 12 la ngay thu 335 trong nam
[~/IT002_OOP]—
(13:22:12)→ ./Bai_5
31 12
Ngày ke tiep la ngay 1
Ngày truoc la ngay 30
Ngày 31 thang 12 la ngay thu 365 trong nam
[~/IT002_OOP]—
(13:22:21)→ |

```


Chương 6

Bài 6.1

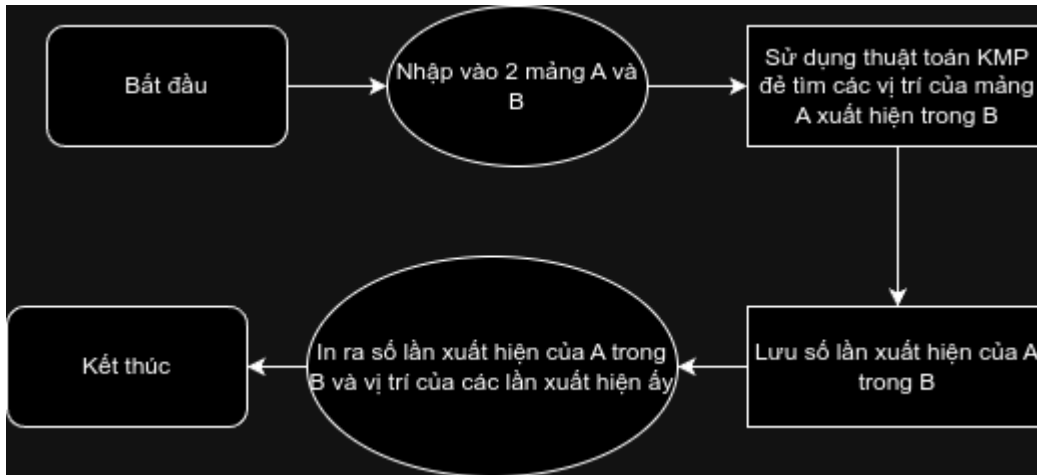
Cho hai mảng A và B chứa các số nguyên. Hãy xây dựng một chương trình để xác định xem có bao nhiêu lần mảng A xuất hiện liên tiếp dưới dạng một dãy con trong mảng B. Cụ thể:

- 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:

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

6.1 Flow Chart



6.2 Input, Output và Solution

Input: 2 giá trị n và m , với n là độ dài mảng A , m là độ dài mảng B . Sau đó là các phần tử lần lượt trong mảng A và trong mảng B

Output: Số lần mảng A xuất hiện trong mảng B và vị trí xuất hiện của A trong B .

Solution: Ta sử dụng thuật toán KMP để tìm số lần A xuất hiện trong B . Vì độ phức tạp của KMP là $O(n+m)$ với n là độ dài của mảng A và m là độ dài của mảng B nên với input lớn thuật toán vẫn có thể thực hiện tốt.

6.3 Code

```

#include <bits/stdc++.h>
using namespace std;

// trong bài này, mảng A sẽ là mảng con và ta cần tìm số lần xuất hiện
// của mảng A trong mảng B

// Hàm dùng để tính Longest Proper Prefix với pat( viết tắt cho pattern )
// là mảng A đã nhập vào
void computeLPSArray(vector<int> pat, int M, vector<int>& lps)
{

```

```

int len = 0;
lps[0] = 0;
int i = 1;

while (i < M) {
    if (pat[i] == pat[len]) {
        len++;
        lps[i] = len;
        i++;
    } else {
        if (len != 0) {
            len = lps[len - 1];
        } else {
            lps[i] = 0;
            i++;
        }
    }
}

// Ham thuc hien search cho cac lan xuat hien cua A trong B
// pat( pattern) la mang A, txt( text) chinh la mang B
vector<int> KMPSearch(vector<int> pat , vector<int> txt)
{
    int M = pat.size();
    int N = txt.size();
    vector<int> lps(M);
    vector<int> result;

    computeLPSArray(pat , M, lps);

    int i = 0;
    int j = 0;

    while ((N - i) >= (M - j)) {
        if (pat[j] == txt[i]) {

```

```
        j++;
        i++;
    }

    if (j == M) {
        result.push_back(i - j + 1);
        j = lps[j - 1];
    } else if (i < N && pat[j] != txt[i]) {
        if (j != 0)
            j = lps[j - 1];
        else
            i = i + 1;
    }
}
return result;
}

int main()
{
    vector<int> pat;
    vector<int> txt;
    int n, m, temp;
    cin >> n >> m;

    for (int i = 0; i < n; i++) {
        cin >> temp;
        pat.push_back(temp);
    }

    for (int i = 0; i < m; i++) {
        cin >> temp;
        txt.push_back(temp);
    }

    vector<int> result = KMPSearch(pat, txt);
```



```
cout << "So lan mang A xuat hien
trong mang B la: " << result.size() << endl;
if (result.size() > 0)
    cout << "Cac vi tri ma mang A xuat hien
    trong mang B la: ";
for (int i = 0; i < result.size(); i++) {
    cout << result[i] - 1 << " ";
}
cout << endl;
return 0;
}
```

6.4 Testcase

```
turtle
[~/IT002_OOP]—
[13:59:11]—> ./Bai_6
3 10
2 3 5
1 2 3 5 2 3 5 2 3 5
So lan mang A xuất hiện trong mang B là: 3
Cac vi tri ma mang A xuất hiện trong mang B là: 1 4 7
[~/IT002_OOP]—
[13:59:30]—> ./Bai_6
5 5
1 2 3 4 5
1 2 3 4 5
So lan mang A xuất hiện trong mang B là: 1
Cac vi tri ma mang A xuất hiện trong mang B là: 0
[~/IT002_OOP]—
[13:59:40]—> ./Bai_6
2 5
1 2
5 6 7 8 9
So lan mang A xuất hiện trong mang B là: 0
[~/IT002_OOP]—
[13:59:56]—> |
```

Chương 7

Bài 6.2

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:

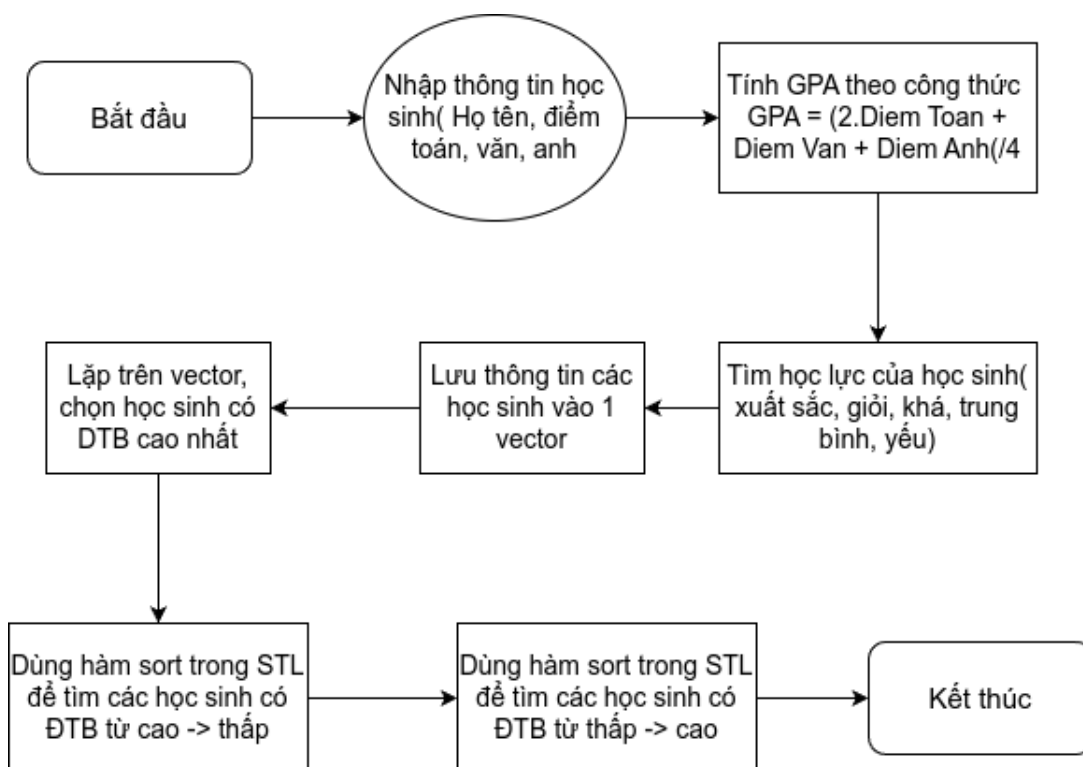
1. Nhập thông tin cho n học sinh, bao gồm:
 - Họ tên (phải kiểm tra độ dài và định dạng, không chứa ký tự số).
 - Điểm Toán, Văn và Ngoại ngữ (phải là số thực trong khoảng từ 0 đến 10, nếu nhập sai phải yêu cầu nhập lại).
 2. Tính điểm trung bình của mỗi học sinh theo công thức: $ETB = \frac{2*DiemToan + DiemVan + DiemAnh}{4}$
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).
 6. Xuất danh sách học sinh có điểm toán thấp nhất.
- Yêu cầu:
- Chương trình cần có giao diện nhập liệu và xuất kết quả rõ ràng.
 - Đảm bảo kiểm tra tính hợp lệ của tất cả các dữ liệu đầu vào.
 - Phân loại học sinh phải chính xác và chương trình phải có tính linh hoạt khi tìm kiếm thông tin theo tên.
- Đầu vào:
- Số lượng học sinh: 3

- Thông tin từng học sinh bao gồm họ tên, điểm toán, điểm văn, điểm ngoại ngữ.

Đầu ra:

- Điểm trung bình và phân loại học sinh.
- Học sinh có điểm trung bình cao nhất.
- Kết quả tìm kiếm theo tên.
- Danh sách học sinh có điểm toán thấp nhất

7.1 Flow Chart



7.2 Input, Output và Solution

Input: Họ và tên học sinh, điểm toán, điểm văn, điểm anh

Output: Điểm trung bình và phân loại học sinh, học sinh có ĐTB cao nhất, thấp nhất

Solution: tạo một struct để lưu các thông tin của học sinh như họ tên, điểm toán, văn, anh.

Sau đó lưu các học sinh vào một vector. Duyệt tuyến tính theo vector để làm các nhiệm vụ đề bài đưa ra.

7.3 Code

Link code: https://github.com/LowTechTurtle/IT002_OOP/blob/main/Lab1/Bai6_2.cpp

7.4 Testcase

Dưới đây là log của output:

Nhap so sinh vien ban muon them vao

3

Nhap Ho va Ten hoc sinh: Nguyen Van @\$

Nhap lai Ho va Ten hoc sinh: Nguyen Van A

Nhap diem mon Toan: 4

Nhap diem mon Van: 7

Nhap diem mon Ngoai ngu: 11

Nhap lai diem mon Ngoai ngu: 1

Nhap Ho va Ten hoc sinh: Le Van B

Nhap diem mon Toan: 5

Nhap diem mon Van: 67

Nhap lai diem mon Van: 6

Nhap diem mon Ngoai ngu: 7

Nhap Ho va Ten hoc sinh: Tran Thi C

Nhap diem mon Toan: 10

Nhap diem mon Van: 3

Nhap diem mon Ngoai ngu: 5

1. Tinh diem trung binh

2. Phan loai HS

3. Tim hoc sinh co diem TB cao nhat

4. Tim hoc sinh theo ten

5. Tim hoc sinh co diem toan thap nhat

1

Ban muon tinh diem TB cua hoc sinh thu may?

2

Xuat diem trung binh va phan loai

Ten: Le Van B

Diem TB: 5.75

Danh Hieu: Trung binh

1. Tinh diem trung binh
2. Phan loai HS
3. Tim hoc sinh co diem TB cao nhat
4. Tim hoc sinh theo ten
5. Tim hoc sinh co diem toan thap nhat

1

Ban muon tinh diem TB cua hoc sinh thu may?

1

Xuat diem trung binh va phan loai

Ten: Nguyen Van A

Diem TB: 3.5

Danh Hieu: Yeu

1. Tinh diem trung binh
2. Phan loai HS
3. Tim hoc sinh co diem TB cao nhat
4. Tim hoc sinh theo ten
5. Tim hoc sinh co diem toan thap nhat

2

Ban muon xem loai hoc luc cua hoc sinh thu may?

3

Xuat diem trung binh va phan loai

Ten: Tran Thi C

Diem TB: 7

Danh Hieu: Kha

1. Tinh diem trung binh
2. Phan loai HS
3. Tim hoc sinh co diem TB cao nhat
4. Tim hoc sinh theo ten
5. Tim hoc sinh co diem toan thap nhat

3

Danh Sach hoc sinh co diem trung binh cao nhat la:

Tran Thi C

1. Tinh diem trung binh
 2. Phan loai HS
 3. Tim hoc sinh co diem TB cao nhat
 4. Tim hoc sinh theo ten
 5. Tim hoc sinh co diem toan thap nhat
- 4

Nhap ten:

Tran Thi C

Xuat diem trung binh va phan loai

Ten: Tran Thi C

Diem TB: 7

Danh Hieu: Kha

1. Tinh diem trung binh
 2. Phan loai HS
 3. Tim hoc sinh co diem TB cao nhat
 4. Tim hoc sinh theo ten
 5. Tim hoc sinh co diem toan thap nhat
- 5

Danh Sach Hoc sinh co diem toan thap nhat la:

Nguyen Van A

1. Tinh diem trung binh
2. Phan loai HS
3. Tim hoc sinh co diem TB cao nhat
4. Tim hoc sinh theo ten
5. Tim hoc sinh co diem toan thap nhat

Chương 8

Bài 7

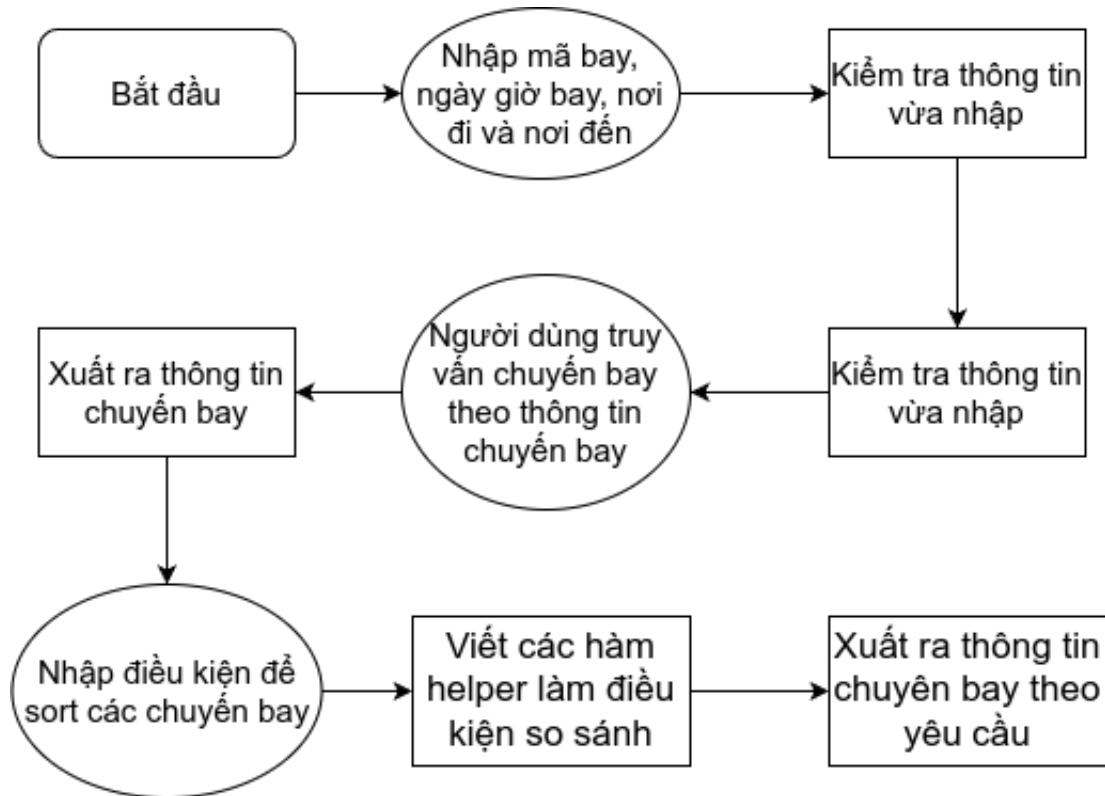
Hãy xây dựng một chương trình quản lý thông tin chuyến bay. Mỗi chuyến bay bao gồm các thông tin sau:

- 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.

8.1 Flow Chart



8.2 Input, Output và Solution

Đầu 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.

Đầu ra:

- Danh sách chuyến bay đã sắp xếp theo ngày giờ khởi hành.
- Kết quả tìm kiếm và các báo cáo theo yêu cầu.

Solution: Dùng struct để lưu ngày bay, dùng 1 struct khác để lưu thời gian bay, và 1 struct khác để lưu mã chuyến bay, ngày giờ bay, nơi đi, nơi đến. Sau đó lưu các chuyến bay vào trong 1 vector. Sau đó lặp trên vector đó để tìm các chuyến bay theo các yêu cầu khác nhau. Có thể dùng hàm sort của STL library để sort các chuyến bay theo các điều kiện khác nhau(dùng function pointer cho argument thứ 3 trong hàm sort)

8.3 Code

Link code: https://github.com/LowTechTurtle/IT002_OOP/blob/main/Lab1/Bai_7.cpp

8.4 Testcase

Dưới đây là log của lần test.

```

    Nhập vào số chuyến bay: 2
Nhập mã chuyến bay: VN!@#
Nhập lại mã chuyến bay: VN123
Nhập ngày bay( nhập ngày, tháng bằng số): 15 17
Nhập lại ngày bay: 15 7
Nhập giờ bay( nhập giờ, phút bằng số): 12 15
Nhập nơi khởi hành: VN
Nhập nơi hạ cánh: US
Nhập mã chuyến bay: US234
Nhập ngày bay( nhập ngày, tháng bằng số): 2 5
Nhập giờ bay( nhập giờ, phút bằng số): 16 20
Nhập nơi khởi hành: US
Nhập nơi hạ cánh: Singapore
1. Tìm chuyến bay bằng mã chuyến bay
2. Tìm chuyến bay bằng nơi đi
3. Tìm chuyến bay bằng nơi đến
4. Tìm chuyến bay từ một nơi cụ thể trong 1 ngày nhất định
5. Tìm chuyến bay bằng nơi đi và nơi đến nhất định
6. Sắp xếp chuyến bay theo ngày giờ khởi hành
1
Nhập mã chuyến bay: US2345
Nhập lại mã chuyến bay: US234
Mã chuyến bay: US234
Ngày bay: 2/5
Giờ bay: 16:20
Nơi khởi hành: US
Nơi hạ cánh: Singapore
1. Tìm chuyến bay bằng mã chuyến bay

```

2. Tim chuyen bay bang noi di
3. Tim chuyen bay bang noi den
4. Tim chuyen bay tu mot noi cu the trong 1 ngay nhat dinh
5. Tim chuyen bay bang noi di va noi den nhat dinh
6. Sap xep chuyen bay theo ngay gio khoi hanh

3

Nhap noi ha canh: US

Ma chuyen bay: VN123

Ngay bay: 15/7

Gio bay: 12:15

Noi khoi hanh: VN

Noi ha canh: US

1. Tim chuyen bay bang ma chuyen bay
2. Tim chuyen bay bang noi di
3. Tim chuyen bay bang noi den
4. Tim chuyen bay tu mot noi cu the trong 1 ngay nhat dinh
5. Tim chuyen bay bang noi di va noi den nhat dinh
6. Sap xep chuyen bay theo ngay gio khoi hanh

4

Nhap noi khoi hanh: VN

Nhap ngay va thang(nhap bang so): 15 7

Ma chuyen bay: VN123

Ngay bay: 15/7

Gio bay: 12:15

Noi khoi hanh: VN

Noi ha canh: US

1. Tim chuyen bay bang ma chuyen bay
2. Tim chuyen bay bang noi di
3. Tim chuyen bay bang noi den
4. Tim chuyen bay tu mot noi cu the trong 1 ngay nhat dinh
5. Tim chuyen bay bang noi di va noi den nhat dinh
6. Sap xep chuyen bay theo ngay gio khoi hanh

5

Nhap noi khoi hanh: VN

Nhap noi ha canh: US

Tim thay 1 chuyen bay

1. Tim chuyen bay bang ma chuyen bay
2. Tim chuyen bay bang noi di
3. Tim chuyen bay bang noi den
4. Tim chuyen bay tu mot noi cu the trong 1 ngay nhat dinh
5. Tim chuyen bay bang noi di va noi den nhat dinh
6. Sap xep chuyen bay theo ngay gio khoi hanh

6

Cac chuyen bay da duoc sort theo ngay thang va gio bay

1. Tim chuyen bay bang ma chuyen bay
2. Tim chuyen bay bang noi di
3. Tim chuyen bay bang noi den
4. Tim chuyen bay tu mot noi cu the trong 1 ngay nhat dinh
5. Tim chuyen bay bang noi di va noi den nhat dinh
6. Sap xep chuyen bay theo ngay gio khoi hanh

Chương 9

Bài 8

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 sổ: 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 sổ (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 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

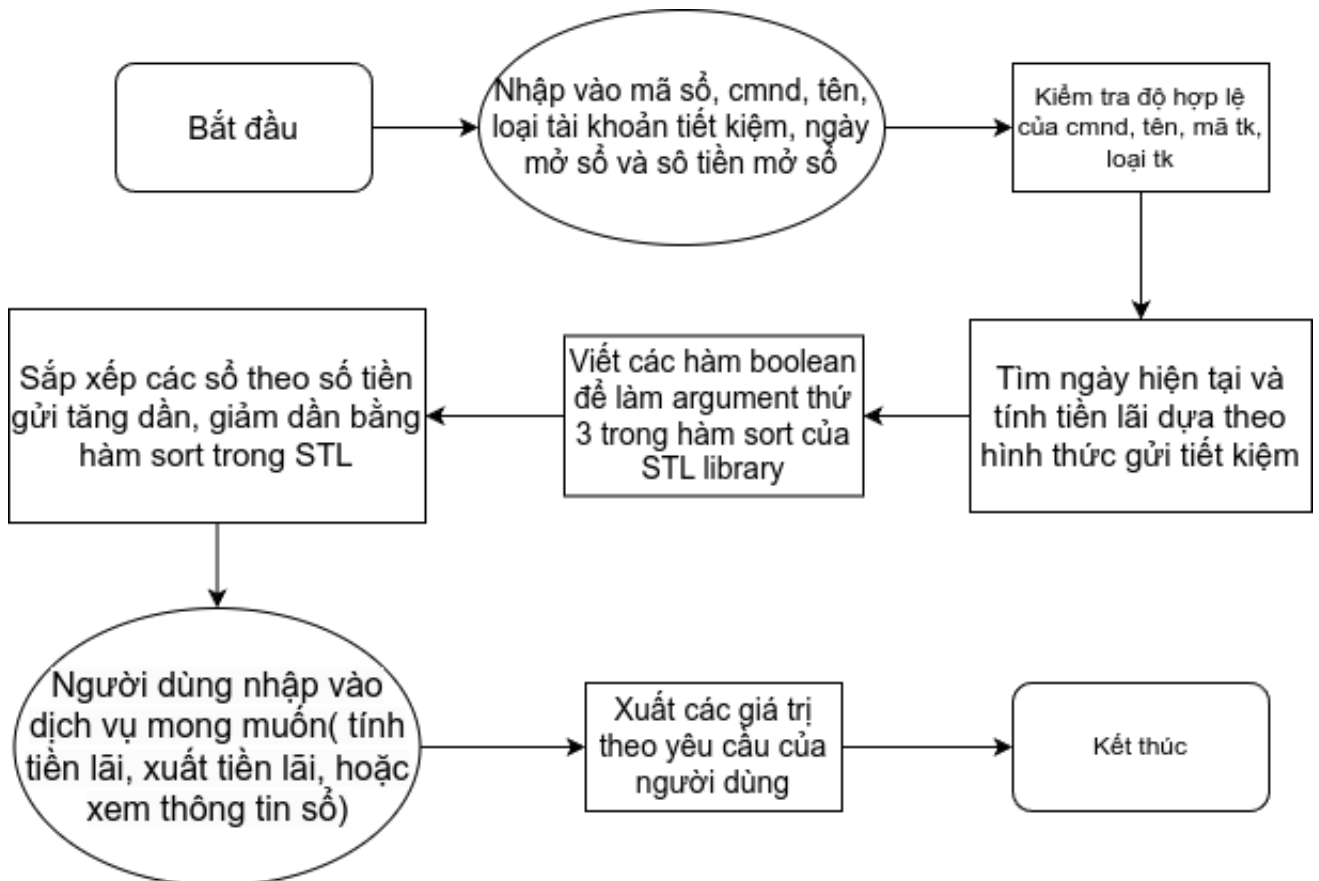
5. Tìm kiếm và liệt kê:

- Tìm kiếm sổ tiết kiệm theo CMND hoặc mã sổ.
- 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.

9.1 Flow Chart



9.2 Input, Output và Solution

Input: Nhập vào các thông tin của sổ tiết kiệm như mã sổ, sổ chứng minh nhân dân, tên, loại tiết kiệm, ngày mở sổ, số tiền mở sổ.

Output: In ra thông tin của người dùng khi được truy vấn. Ngoài ra còn tính tiền lãi từ ngày mở sổ

Solution: Sử dụng 1 struct để lưu ngày tháng năm mở sổ và 1 struct để lưu các thông tin còn lại như mã sổ, cmnd, tên, loại sổ tiết kiệm, và tiền trong sổ. Ta có thể lấy thời gian của thời điểm hiện tại bằng hàm localtime. Ngoài ra lãi suất được nhập vào là số %. Ví dụ muốn nhập tiền lãi là 5%/năm thì sẽ nhập là 5% thay vì 0.05.

9.3 Code

Link code: https://github.com/LowTechTurtle/IT002_OOP/blob/main/Lab1/Bai_8.cpp

9.4 Testcase

Log của output:

```
1: Mo so tiet kiem
2: Tim kiem du lieu so tiet kiem
3: Cap nhat lai suat
4: Tinh tien lai
5: Rut tien
6.1: Sap xep danh sach giam dan theo tien gui
6.2: Sap xep theo ngay mo so tang dan
end: Thoat
1
Nhap ma so :
123
Chon loai tiet kiem (nhap ngan han hoac dai han)
Ngan han (6 thang) (3%/nam)
Dai han (12 thang) (5%/nam)
ngan han
Nhap Ten: Dong Quoc Thang
Nhap chung minh nhan dan: 123456789
Nhap ngay mo so , nhap ngay thang nam :1 1 2023
so tien gui:
200000
```

1: Mo so tiet kiem
2: Tim kiem du lieu so tiet kiem
3: Cap nhat lai suat
4: Tinh tien lai
5: Rut tien
6.1: Sap xep danh sach giam dan theo tien gui
6.2: Sap xep theo ngay mo so tang dan
end: Thoat

2

Go 1 chu so tuong ung voi cac lua chon sau day:

1: In ra tat ca
2: Tim kiem thong tin so
3: In ra cac thong tin so trong 1 khoang thoi gian nhat dinh
1

ma so, ho va ten, loai tiet kiem, cmnd,
thoi gian mo so, so tien

123, Dong Quoc Thang, ngan han, 123456789, 1 1 2023, 200000

1: Mo so tiet kiem
2: Tim kiem du lieu so tiet kiem
3: Cap nhat lai suat
4: Tinh tien lai
5: Rut tien
6.1: Sap xep danh sach giam dan theo tien gui
6.2: Sap xep theo ngay mo so tang dan
end: Thoat

3

Nhap ma so

123

Nhap lai suat

7

Lai suat da duoc cap nhat

1: Mo so tiet kiem
2: Tim kiem du lieu so tiet kiem

3: Cap nhat lai suat
4: Tinh tien lai
5: Rut tien
6.1: Sap xep danh sach giam dan theo tien gui
6.2: Sap xep theo ngay mo so tang dan
end: Thoat

4

Nhap ma so

123

so tien lai:

42000

1: Mo so tiet kiem
2: Tim kiem du lieu so tiet kiem
3: Cap nhat lai suat
4: Tinh tien lai
5: Rut tien
6.1: Sap xep danh sach giam dan theo tien gui
6.2: Sap xep theo ngay mo so tang dan
end: Thoat

5

Nhap ma so can rut tien

123

So tien can rut:

100000

1: Mo so tiet kiem
2: Tim kiem du lieu so tiet kiem
3: Cap nhat lai suat
4: Tinh tien lai
5: Rut tien
6.1: Sap xep danh sach giam dan theo tien gui
6.2: Sap xep theo ngay mo so tang dan
end: Thoat

2

Go 1 chu so tuong ung voi cac lua chon sau day:

1: In ra tat ca
2: Tim kiem thong tin so
3: In ra cac thong tin so trong 1 khoang thoi gian nhat dinh
1

ma so, ho va ten, loai tiet kiem,
cmnd, thoi gian mo so, so tien
123, Dong Quoc Thang, ngan han, 123456789, 1 1 2023, 100000

1: Mo so tiet kiem
2: Tim kiem du lieu so tiet kiem
3: Cap nhat lai suat
4: Tinh tien lai
5: Rut tien
6.1: Sap xep danh sach giam dan theo tien gui
6.2: Sap xep theo ngay mo so tang dan
end: Thoat
6.1
Cac so tiet kiem da duoc sort theo thu tu giam dan

1: Mo so tiet kiem
2: Tim kiem du lieu so tiet kiem
3: Cap nhat lai suat
4: Tinh tien lai
5: Rut tien
6.1: Sap xep danh sach giam dan theo tien gui
6.2: Sap xep theo ngay mo so tang dan
end: Thoat
6.2
Cac so tiet kiem da duoc sort theo thu tu tang dan

1: Mo so tiet kiem
2: Tim kiem du lieu so tiet kiem
3: Cap nhat lai suat
4: Tinh tien lai
5: Rut tien

6.1: Sap xep danh sach giam dan theo tien gui
6.2: Sap xep theo ngay mo so tang dan
end: Thoat
end