

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



MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
BÀI TẬP THỰC HÀNH 1

GVHD: Nguyễn Ngọc Quý

Sinh viên thực hiện: Đoàn Nguyễn Minh Thư - 24521725

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

[illegible]

....., ngày.....tháng.....năm 2025

Người nhận xét

(Ký tên và ghi rõ họ tên)

MỤC LỤC

Bài tập 1.....	6
Bài tập 2.....	11
Bài tập 3.....	14
Bài tập 4.....	22
Bài tập 5.....	24
Bài tập 6.....	30
Bài tập 7.....	33
Bài tập 8.....	38
Bài tập 9.....	45

DANH MỤC HÌNH ẢNH

Figure 1 - Flowchart chương trình rút gọn phân số	6
Figure 2 - Flowchart chương trình tìm phân số lớn nhất	11
Figure 3 - Flowchart chương trình tính toán hai phân số	15
Figure 4 - Flowchart chương trình tính $\sin(x)$	22
Figure 5 - Flowchart chương trình tính ngày	25
Figure 6 - Flowchart chương trình đếm số lần xuất hiện	30
Figure 7 - Flowchart chương trình quản lý học sinh	33
Figure 8 - Flowchart chương trình quản lý thông tin chuyến bay	38
Figure 9 - Flowchart chương trình quản lý sổ tiết kiệm	45

DANH MỤC BẢNG

Table 1 - Code chương trình rút gọn phân số	9
Table 2 - Code chương trình tìm phân số lớn nhất	13
Table 3 - Code chương trình tính toán hai phân số	20
Table 4 - Code chương trình tính $\sin(x)$	24
Table 5 - Code chương trình tính ngày	29
Table 6 - Code chương trình đếm số lần xuất hiện	32
Table 7 - Code chương trình quản lý học sinh	37
Table 8 - Code chương trình quản lý thông tin chuyến bay	44
Table 9 - Code chương trình quản lý sổ tiết kiệm	53

NỘI DUNG BÀI LÀM

Bài tập 1: Viết chương trình nhập vào một phân số, rút gọn phân số và xuất kết quả.

Nội dung flowchart của chương trình rút gọn phân số được trình bày như hình 1.

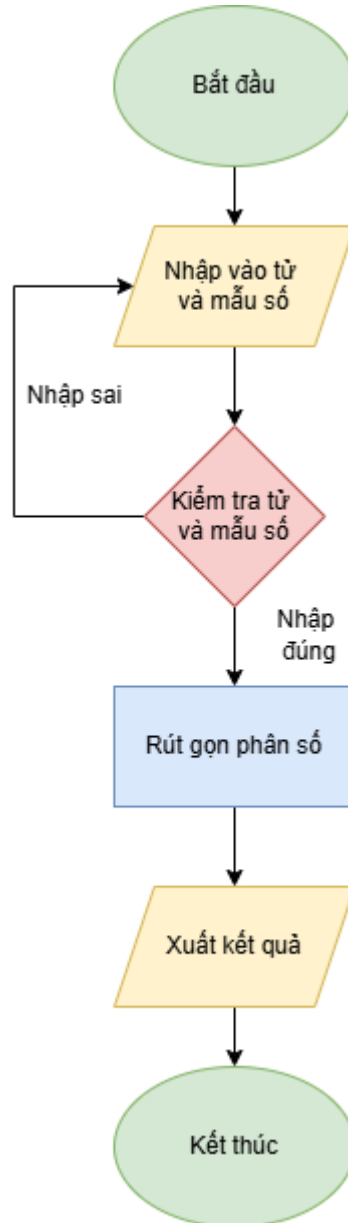


Figure 1 - Flowchart chương trình rút gọn phân số

Mô tả đầu vào, đầu ra và hướng giải quyết của chương trình rút gọn phân số:

- Đầu vào: Tử số và mẫu số
- Đầu ra: Phân số đã được rút gọn
- Hướng giải quyết:

IT002 – Lập trình hướng đối tượng

- + Đối với phân số chứa tử và mẫu đều là số nguyên: Tìm ước chung lớn nhất của tử số và mẫu số sau đó chia tử và mẫu cho ước chung lớn nhất này và xuất phân số đã được rút gọn ra màn hình.
- + Đối với phân số chứa số thực: Chuyển tử số và mẫu số thành số nguyên bằng cách chuyển từng số sang dạng phân số với tử số = tử số * (10 mũ số lượng số sau dấu chấm), mẫu số = 10 mũ số lượng số sau dấu chấm. VD : 3.9 → 39/10. Sau khi đã biến đổi cả tử và mẫu về một phân số nguyên thì thực hiện nhân chéo để tạo một phân số mới. Tiếp tục xử lý phân số mới như một phân số nguyên và xuất phân số đã được rút gọn ra màn hình.

Nội dung code của chương trình rút gọn phân số:

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

using namespace std;

//Định nghĩa cấu trúc phân số
struct phanSo{
    double tuSo, mauSo;
};

// Hàm kiểm tra xem phân số có chứa số thực hay không
int checkDouble(phanSo a){
    string tuSo = to_string(a.tuSo), mauSo =
to_string(a.mauSo);

    for(int i = 0; i < tuSo.size(); ++i){
        if(tuSo[i] == '.') return 1;
    }

    for(int i = 0; i < mauSo.size(); ++i){
        if(mauSo[i] == '.') return 1;
    }

    return 0; //Không phải số thực
}

//Hàm rút gọn phân số nguyên
phanSo rutGonSoNguyen(phanSo a){
    phanSo tmp;

    //Kiểm tra nếu tử số chia hết cho mẫu số (do phanSo a ở
dạng double nên phải ép về kiểu int)
    if((int)a.tuSo % (int)a.mauSo == 0){
        tmp.tuSo = a.tuSo/a.mauSo;
        tmp.mauSo = 1;
        return tmp;
    }
```

```
//Tìm ước chung lớn nhất (UCLN) của tử số và mẫu số
int UCLN = __gcd((int)a.tuSo, (int)a.mauSo);
tmp.tuSo = a.tuSo/UCLN;
tmp.mauSo = a.mauSo/UCLN;
return tmp;
}

//Hàm rút gọn phân số thực
phanSo rutGonSoThuc(phanSo a){
    string tuSo = to_string(a.tuSo), mauSo =
to_string(a.mauSo);
    int checkTuSo = 0, checkMauSo = 0;
    phanSo tmp, tmp1, tmp2;

    //Xác định vị trí dấu chấm trong tử số, từ đó suy ra số
lượng số sau dấu chấm
    for(int i = 0; i < tuSo.size(); ++i){
        if(tuSo[i] == '.'){
            checkTuSo = i+1;
            break;
        }
    }

    //Xác định vị trí dấu chấm trong mẫu số, từ đó suy ra số
lượng số sau dấu chấm
    for(int i = 0; i < mauSo.size(); ++i){
        if(mauSo[i] == '.'){
            checkMauSo = i+1;
            break;
        }
    }

    //Chuyển tử số thành số nguyên bằng cách nhân với 10^ số
chữ số sau dấu chấm
    if(checkTuSo != 0){
        int pos = tuSo.size() - checkTuSo;
        tmp1.tuSo = a.tuSo*pow(10, pos);
        tmp1.mauSo = pow(10, pos);
    } else{
        tmp1.tuSo = a.tuSo;
        tmp1.mauSo = 1;
    }

    //Chuyển mẫu số thành số nguyên bằng cách nhân với 10^ số
chữ số sau dấu chấm
    if(checkMauSo == 1){
        int pos = mauSo.size() - checkMauSo;
        tmp2.tuSo = a.mauSo*pow(10, pos);
        tmp2.mauSo = pow(10, pos);
    } else{
        tmp2.tuSo = a.mauSo;
        tmp2.mauSo = 1;
    }
}
```



```

    }

    //Chia phân số tmp1 cho tmp2 để được phân số mới
    tmp.tuSo = tmp1.tuSo*tmp2.mauSo;
    tmp.mauSo = tmp1.mauSo*tmp2.tuSo;
    //Rút gọn phân số vừa tìm được
    tmp = rutGonSoNguyen(tmp);

    return tmp;
}

int main(){
    phanSo a, ans;

    //Nhập phân số từ người dùng
    cout << "Nhập tu so: "; cin >> a.tuSo;
    cout << "Nhập mau so: "; cin >> a.mauSo;

    //Kiểm tra mẫu số hợp lệ (Khác 0)
    while(a.mauSo == 0){
        cout << "Phan so khong hop le. Vui long nhap lai: ";
        cout << "Nhập tu so: "; cin >> a.tuSo;
        cout << "Nhập mau so: "; cin >> a.mauSo;
    }

    //Kiểm tra và rút gọn theo loại phân số
    if(checkDouble(a) == 0) ans = rutGonSoNguyen(a);
    else ans = rutGonSoThuc(a);

    //Điều chỉnh để dấu trừ nằm ở tử số
    if(ans.mauSo < 0){
        ans.mauSo = -ans.mauSo;
        ans.tuSo = -ans.tuSo;
    }

    //Xuất kết quả
    cout << "Phan so sau khi rut gon la: " << ans.tuSo;
    if(ans.mauSo != 1) cout << "/" << ans.mauSo;

    return 0;
}

```

Table 1 - Code chương trình rút gọn phân số

Nội dung kiểm thử của chương trình rút gọn phân số:

- Kiểm thử 1: Cho tử số = 8, cho mẫu số = 4, kết quả phân số được rút gọn là 2

```

Nhập tu so: 8
Nhập mau so: 4
Phan so sau khi rut gon la: 2

```

IT002 – Lập trình hướng đối tượng

- Kiểm thử 2: Cho tử số = 3, cho mẫu số = 9, kết quả phân số được rút gọn là $1/3$

```
Nhap tu so: 3
Nhap mau so: 9
Phan so sau khi rut gon la: 1/3
```

- Kiểm thử 3: Cho tử số = 5, cho mẫu số = 0, kết quả lỗi, yêu cầu nhập lại

```
Nhap tu so: 5
Nhap mau so: 0
Phan so khong hop le. Vui long nhap lai: Nhap tu so: |
```

- Kiểm thử 4: Cho tử số = 2.75, cho mẫu số = 2, kết quả phân số được rút gọn là $11/8$

```
Nhap tu so: 2.75
Nhap mau so: 2
Phan so sau khi rut gon la: 11/8
```

- Kiểm thử 5: Cho tử số = 3, cho mẫu số = 3.2, kết quả phân số được rút gọn là $15/16$

```
Nhap tu so: 3
Nhap mau so: 3.2
Phan so sau khi rut gon la: 15/16
```

- Kiểm thử 6: Cho tử số = 2.75, cho mẫu số = 3.2, kết quả phân số được rút gọn là $55/64$

```
Nhap tu so: 2.75
Nhap mau so: 3.2
Phan so sau khi rut gon la: 55/64
```

- Kiểm thử 7: Cho tử số = -8, cho mẫu số = 4, kết quả phân số được rút gọn là -2

```
Nhap tu so: -8
Nhap mau so: 4
Phan so sau khi rut gon la: -2
```

- Kiểm thử 8: Cho tử số = 3, cho mẫu số = -9, kết quả phân số được rút gọn là $-1/3$

```
Nhap tu so: 3
Nhap mau so: -9
Phan so sau khi rut gon la: -1/3
```

- Kiểm thử 9: Cho tử số = -2, cho mẫu số = -4, kết quả phân số được rút gọn là $1/2$

```
Nhap tu so: -2
Nhap mau so: -4
Phan so sau khi rut gon la: 1/2
```

- Kiểm thử 10: Cho tử số = -2.75, cho mẫu số = 2, kết quả phân số được rút gọn là $-11/8$

```
Nhap tu so: -2.75
Nhap mau so: 2
Phan so sau khi rut gon la: -11/8
```

- Kiểm thử 11: Cho tử số = 2, cho mẫu số = -2.75, kết quả phân số được rút gọn là $-8/11$

```
Nhap tu so: 2
Nhap mau so: -2.75
Phan so sau khi rut gon la: -8/11
```

IT002 – Lập trình hướng đối tượng

- Kiểm thử 12: Cho tử số = -2.75, cho mẫu số = -3.2, kết quả phân số được rút gọn là 55/64

```
Nhap tu so: -2.75
Nhap mau so: -3.2
Phan so sau khi rut gon la: 55/64
```

Bài tập 2: Viết chương trình nhập vào hai phân số, tìm phân số lớn nhất và xuất kết quả

Nội dung flowchart của chương trình tìm phân số lớn nhất được trình bày như hình 2.

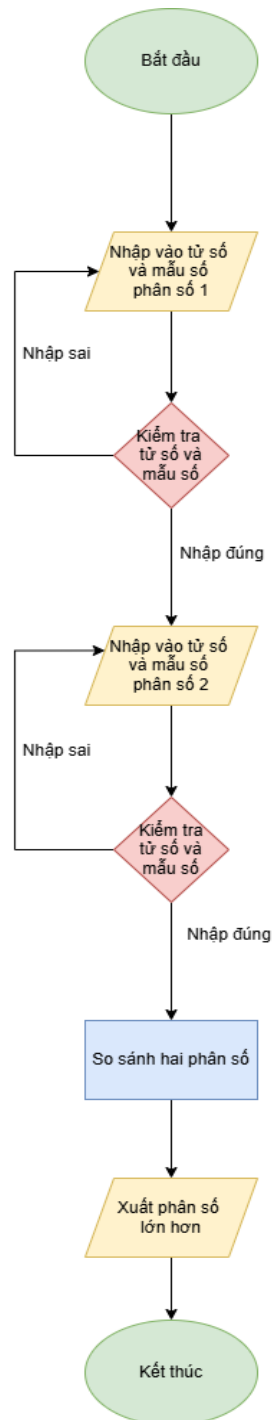


Figure 2 - Flowchart chương trình tìm phân số lớn nhất

IT002 – Lập trình hướng đối tượng

Mô tả đầu vào, đầu ra và hướng giải quyết của chương trình tìm phân số lớn nhất:

- Đầu vào: Tử số và mẫu số của 2 phân số được nhập từ bàn phím.
- Đầu ra: Phân số lớn hơn.
- Hướng giải quyết: Lấy tử số chia cho mẫu số, phân số nào có kết quả thập phân lớn hơn thì xuất phân số đó ra màn hình.

Nội dung code của chương trình tìm phân số lớn nhất:

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

using namespace std;

//Định nghĩa cấu trúc phân số
struct phanSo{
    double tuSo, mauSo;
};

//Hàm nhập hai phân số từ bàn phím
void Nhap(phanSo &a, phanSo &b){
    cout << "Nhap tu so phan so thu 1: ";
    cin >> a.tuSo;
    cout << "Nhap mau so phan so thu 1: ";
    cin >> a.mauSo;

    //Kiểm tra mẫu số khác 0 hay không, nếu không yêu cầu nhập lại
    while(a.mauSo == 0){
        cout << "Phan so khong hop le. Vui long nhap lai" << "\n";
        cout << "Nhap tu so phan so thu 1: ";
        cin >> a.tuSo;
        cout << "Nhap mau so phan so thu 1: ";
        cin >> a.mauSo;
    }

    cout << "Nhap tu so phan so thu 2: ";
    cin >> b.tuSo;
    cout << "Nhap mau so phan so thu 2: ";
    cin >> b.mauSo;

    while(b.mauSo == 0){
        cout << "Phan so khong hop le. Vui long nhap lai" << "\n";
        cout << "Nhap tu so phan so thu 2: ";
        cin >> b.tuSo;
        cout << "Nhap mau so phan so thu 2: ";
        cin >> b.mauSo;
    }
}

//Hàm so sánh hai phân số và trả về phân số lớn hơn
phanSo soSanh(phanSo a, phanSo b){
    double tmp, tmp1;
```

```

    tmp = a.tuSo/a.mauSo; //Giá trị thực của a
    tmp1 = b.tuSo/b.mauSo; //Giá trị thực của b

    if(tmp > tmp1) return a;
    else return b;
}

//Hàm xuất phân số ra màn hình
void Xuat(phanSo a){
    cout << "Phan so lon nhat la: ";
    cout << a.tuSo << "/" << a.mauSo;
}

int main(){
    phanSo a, b, ans;

    Nhap(a, b);
    ans = soSanh(a, b);
    Xuat(ans);

    return 0;
}

```

Table 2 – Code chương trình tìm phân số lớn nhất

Nội dung kiểm thử của chương trình tìm phân số lớn nhất:

- Kiểm thử 1: Cho tử số 1 = 3, cho mẫu số 1 = 4, cho tử số 2 = 5, cho mẫu số 2 = 4, kết quả phân số lớn nhất là 5/4.

```

Nhap tu so phan so thu 1: 3
Nhap mau so phan so thu 1: 4
Nhap tu so phan so thu 2: 5
Nhap mau so phan so thu 2: 4
Phan so lon nhat la: 5/4

```

- Kiểm thử 2: Cho tử số 1 = 3, cho mẫu số 1 = 2, cho tử số 2 = 1, cho mẫu số 2 = 5, kết quả phân số lớn nhất là 3/2.

```

Nhap tu so phan so thu 1: 3
Nhap mau so phan so thu 1: 2
Nhap tu so phan so thu 2: 1
Nhap mau so phan so thu 2: 5
Phan so lon nhat la: 3/2

```

- Kiểm thử 3: Cho tử số 1 = 3.2, cho mẫu số 1 = 2.5, cho tử số 2 = 4.3, cho mẫu số 2 = 3, kết quả phân số lớn nhất là 4.3/3.

```
Nhap tu so phan so thu 1: 3.2
Nhap mau so phan so thu 1: 2.5
Nhap tu so phan so thu 2: 4.3
Nhap mau so phan so thu 2: 3
Phan so lon nhat la: 4.3/3
```

- Kiểm thử 4: Cho tử số 1 = -2, cho mẫu số 1 = 3, cho tử số 2 = -3, cho mẫu số = -2, kết quả phân số lớn nhất là -3/-2.

```
Nhap tu so phan so thu 1: -2
Nhap mau so phan so thu 1: 3
Nhap tu so phan so thu 2: -3
Nhap mau so phan so thu 2: -2
Phan so lon nhat la: -3/-2
```

- Kiểm thử 5: Cho tử số 1 = -3.2, cho mẫu số 1 = 2.5, cho tử số = -3.3, cho mẫu số 2 = 4.3, kết quả phân số lớn nhất là -3.3/4.3.

```
Nhap tu so phan so thu 1: -3.2
Nhap mau so phan so thu 1: 2.5
Nhap tu so phan so thu 2: -3.3
Nhap mau so phan so thu 2: 4.3
Phan so lon nhat la: -3.3/4.3
```

Bài tập 3: Viết chương trình nhập vào hai phân số. Tính tổng, hiệu, tích, thương giữa chúng và xuất kết quả.

Nội dung flowchart chương trình tính toán hai phân số được trình bày như hình 3.

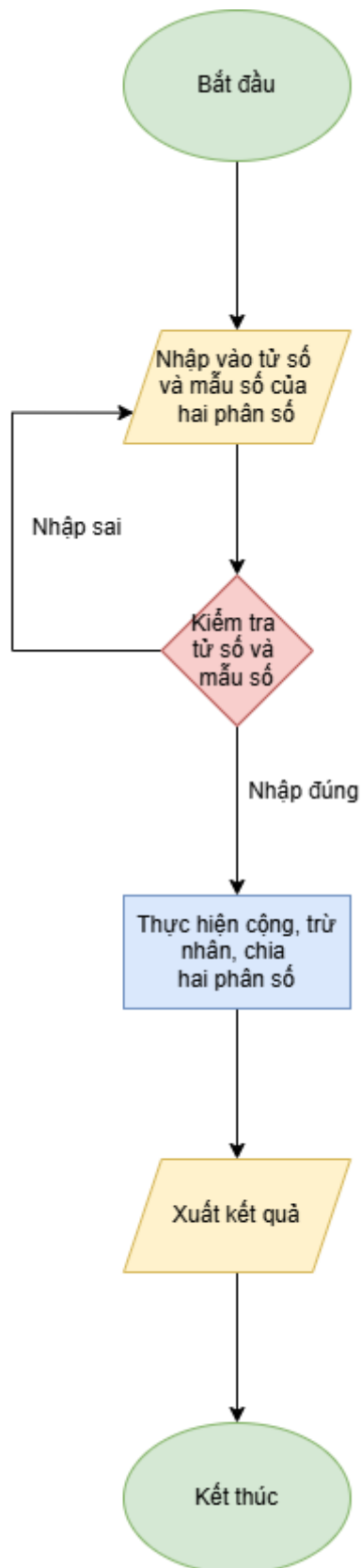


Figure 3 - Flowchart chương trình tính toán hai phân số

Mô tả đầu vào, đầu ra và hướng giải quyết của chương trình tính toán hai phân số:

- Đầu vào: Tử số và mẫu số của 2 phân số được nhập từ bàn phím

IT002 – Lập trình hướng đối tượng

- Đầu ra: Tổng, hiệu, tích, thương của 2 phân số vừa nhập.
- Hướng giải quyết:
 - + Tổng: Quy đồng 2 phân số, cộng lại sau đó rút gọn.
 - + Hiệu: Quy đồng 2 phân số, trừ cho nhau sau đó rút gọn.
 - + Tích: Lấy tử nhân tử, mẫu nhân mẫu sau đó rút gọn.
 - + Thương: Lấy tử của phân số này nhân cho mẫu số của phân số kia và ngược lại, sau đó rút gọn.

Nội dung code của chương trình tính toán hai phân số:

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

using namespace std;

//Định nghĩa cấu trúc của phân số
struct phanSo{
    double tuSo, mauSo;
};

//Hàm nhập 2 phân số
void Nhap(phanSo &a, phanSo &b){
    cout << "Nhap tu so phan so thu 1: "; cin >> a.tuSo;
    cout << "Nhap mau so phan so thu 1: "; cin >> a.mauSo;
    cout << "Nhap tu so phan so thu 2: "; cin >> b.tuSo;
    cout << "Nhap mau so phan so thu 2: "; cin >> b.mauSo;
}

//Định nghĩa toán tử + cho hai phân số
phanSo operator+(phanSo a, phanSo b){
    phanSo tmp;

    tmp.tuSo = a.tuSo*b.mauSo + b.tuSo*a.mauSo;
    tmp.mauSo = a.mauSo*b.mauSo;

    return tmp;
}

//Định nghĩa toán tử - cho hai phân số
phanSo operator-(phanSo a, phanSo b){
    phanSo tmp;

    tmp.tuSo = a.tuSo*b.mauSo - b.tuSo*a.mauSo;
    tmp.mauSo = a.mauSo*b.mauSo;

    return tmp;
}

//Định nghĩa toán tử * cho hai phân số
phanSo operator*(phanSo a, phanSo b){
```



```
        phanSo tmp;

        tmp.tuSo = a.tuSo*b.tuSo;
        tmp.mauSo = a.mauSo*b.mauSo;

        return tmp;
    }

    //Định nghĩa toán tử / cho hai phân số
    phanSo operator/(phanSo a, phanSo b){
        phanSo tmp;

        tmp.tuSo = a.tuSo*b.mauSo;
        tmp.mauSo = a.mauSo*b.tuSo;

        return tmp;
    }

    //Hàm kiểm tra xem phân số có chứa số thực hay không
    //Bằng cách chuyển tử số và mẫu số sang chuỗi
    //và kiểm tra có dấu chấm hay không.
    int checkDouble(phanSo a){
        string tuSo = to_string(a.tuSo), mauSo =
        to_string(a.mauSo);

        for(int i = 0; i < tuSo.size(); ++i){
            if(tuSo[i] == '.') return 1;
        }

        for(int i = 0; i < mauSo.size(); ++i){
            if(mauSo[i] == '.') return 1;
        }

        return 0; //Không phải số thực
    }

    //Hàm rút gọn phân số nguyên
    phanSo rutGonSoNguyen(phanSo a){
        phanSo tmp;
        int tuSo = round(a.tuSo), mauSo = round(a.mauSo);

        //Kiểm tra nếu tử số chia hết cho mẫu số (do phanSo a ở
        dạng double nên phải ép về kiểu int)
        if(tuSo % mauSo == 0){
            tmp.tuSo = tuSo/mauSo;
            tmp.mauSo = 1;
            return tmp;
        }

        //Tìm ước chung lớn nhất (UCLN) của tử số và mẫu số
        int UCLN = __gcd(tuSo, mauSo);
        tmp.tuSo = tuSo/UCLN;
        tmp.mauSo = mauSo/UCLN;
```

```

    return tmp;
}

//Hàm rút gọn phân số thực
phanSo rutGonSoThuc(phanSo a){
    string tuSo = to_string(a.tuSo), mauSo =
to_string(a.mauSo);
    int checkTuSo = 0, checkMauSo = 0;
    phanSo tmp, tmp1, tmp2;

    int flag = 0;
    while(tuSo[flag] != '0') ++flag;
    tuSo.erase(std::remove(tuSo.begin()+flag, tuSo.end(),
'0'), tuSo.end());

    flag = 0;
    while(mauSo[flag] != '0') ++flag;
    mauSo.erase(std::remove(mauSo.begin()+flag, mauSo.end(),
'0'), mauSo.end());

    //Xác định vị trí dấu chấm trong tử số, từ đó suy ra số
lượng số sau dấu chấm
    for(int i = 0; i < tuSo.size(); ++i){
        if(tuSo[i] == '.'){
            checkTuSo = i+1;
            break;
        }
    }

    //Xác định vị trí dấu chấm trong mẫu số, từ đó suy ra số
lượng số sau dấu chấm
    for(int i = 0; i < mauSo.size(); ++i){
        if(mauSo[i] == '.'){
            checkMauSo = i+1;
            break;
        }
    }

    //Chuyển tử số thành số nguyên bằng cách nhân với 10^ số
chữ số sau dấu chấm
    if(checkTuSo != 0){
        int pos = tuSo.size() - checkTuSo;
        tmp1.tuSo = a.tuSo*pow(10, pos);
        tmp1.mauSo = pow(10, pos);
    } else{
        tmp1.tuSo = a.tuSo;
        tmp1.mauSo = 1;
    }

    //Chuyển mẫu số thành số nguyên bằng cách nhân với 10^ số
chữ số sau dấu chấm
    if(checkMauSo != 0){

```

```
        int pos = mauSo.size() - checkMauSo;
        tmp2.tuSo = a.mauSo*pow(10, pos);
        tmp2.mauSo = pow(10, pos);
    } else{
        tmp2.tuSo = a.mauSo;
        tmp2.mauSo = 1;
    }

    //Chia phân số tmp1 cho tmp2 để được phân số mới
    tmp.tuSo = tmp1.tuSo*tmp2.mauSo;
    tmp.mauSo = tmp1.mauSo*tmp2.tuSo;
    //Rút gọn phân số vừa tìm được
    tmp = rutGonSoNguyen(tmp);

    return tmp;
}

//Hàm chuẩn hóa mẫu số luôn dương
phanSo chuanHoa(phanSo ans){
    if(ans.mauSo < 0){
        ans.mauSo = -ans.mauSo;
        ans.tuSo = -ans.tuSo;
    }

    return ans;
}

int main(){
    phanSo a, b, ans;

    Nhap(a, b);

    ans = a+b;
    if(checkDouble(ans) == 1) ans = rutGonSoThuc(ans);
    else ans = rutGonSoNguyen(ans);
    ans = chuanHoa(ans);
    cout << "Tong hai phan so la: " << ans.tuSo << "/" <<
ans.mauSo << "\n";

    ans = a-b;
    if(checkDouble(ans) == 1) ans = rutGonSoThuc(ans);
    else ans = rutGonSoNguyen(ans);
    ans = chuanHoa(ans);
    cout << "Hieu hai phan so la: " << ans.tuSo << "/" <<
ans.mauSo << "\n";

    ans = a*b;
    if(checkDouble(ans) == 1) ans = rutGonSoThuc(ans);
    else ans = rutGonSoNguyen(ans);
    ans = chuanHoa(ans);
    cout << "Tich hai phan so la: " << ans.tuSo << "/" <<
ans.mauSo << "\n";
```

```
ans = a/b;  
if(checkDouble(ans) == 1) ans = rutGonSoThuc(ans);  
else ans = rutGonSoNguyen(ans);  
ans = chuanHoa(ans);  
cout << "Thuong hai phan so la: " << ans.tuSo << "/" <<  
ans.mauSo << "\n";  
  
return 0;  
}
```

Table 3 - Code chương trình tính toán hai phân số

Nội dung kiểm thử của chương trình tính toán hai phân số:

- Kiểm thử 1: Cho tử số 1 = 3, cho mẫu số 1 = 4, cho tử số 2 = 5, cho mẫu số 2 = 4, kết quả: tổng = 2/1, hiệu = -1/2, tích = 15/16, thương = 3/5.

```
Nhap tu so phan so thu 1: 3  
Nhap mau so phan so thu 1: 4  
Nhap tu so phan so thu 2: 5  
Nhap mau so phan so thu 2: 4  
Tong hai phan so la: 2/1  
Hieu hai phan so la: -1/2  
Tich hai phan so la: 15/16  
Thuong hai phan so la: 3/5
```

- Kiểm thử 2: Cho tử số 1 = 3, cho mẫu số 1 = 2, cho tử số 2 = 1, cho mẫu số 2 = 5, kết quả: tổng = 17/10, hiệu = 13/10, tích = 3/10, thương = 15/2.

```
Nhap tu so phan so thu 1: 3  
Nhap mau so phan so thu 1: 2  
Nhap tu so phan so thu 2: 1  
Nhap mau so phan so thu 2: 5  
Tong hai phan so la: 17/10  
Hieu hai phan so la: 13/10  
Tich hai phan so la: 3/10  
Thuong hai phan so la: 15/2
```

IT002 – Lập trình hướng đối tượng

- Kiểm thử 3: Cho tử số 1 = 3.2, cho mẫu số 1 = 2.5, cho tử số 2 = 4.3, cho mẫu số 2 = 3, kết quả: tổng = 407/150, hiệu = -23/150, tích = 688/375, thương = 192/215.

```
Nhap tu so phan so thu 1: 3.2
Nhap mau so phan so thu 1: 2.5
Nhap tu so phan so thu 2: 4.3
Nhap mau so phan so thu 2: 3
Tong hai phan so la: 407/150
Hieu hai phan so la: -23/150
Tich hai phan so la: 688/375
Thuong hai phan so la: 192/215
```

- Kiểm thử 4: Cho tử số 1 = -2, cho mẫu số 1 = 3, cho tử số 2 = -3, cho mẫu số 2 = -2, kết quả: tổng = 5/6, hiệu = -13/6, tích = -1/1, thương = -4/9.

```
Nhap tu so phan so thu 1: -2
Nhap mau so phan so thu 1: 3
Nhap tu so phan so thu 2: -3
Nhap mau so phan so thu 2: -2
Tong hai phan so la: 5/6
Hieu hai phan so la: -13/6
Tich hai phan so la: -1/1
Thuong hai phan so la: -4/9
```

- Kiểm thử 5: Cho tử số 1 = -3.2, cho mẫu số 1 = 2.5, cho tử số 2 = -3.3, cho mẫu số 2 = 4.3, kết quả: tổng = -2201/1075, hiệu = -551/1075, tích = 1056/1075, thương = 1376/825.

```
Nhap tu so phan so thu 1: -3.2
Nhap mau so phan so thu 1: 2.5
Nhap tu so phan so thu 2: -3.3
Nhap mau so phan so thu 2: 4.3
Tong hai phan so la: -2201/1075
Hieu hai phan so la: -551/1075
Tich hai phan so la: 1056/1075
Thuong hai phan so la: 1376/825
```

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

Nội dung flowchart của chương trình tính $\sin(x)$ được trình bày như hình 4.

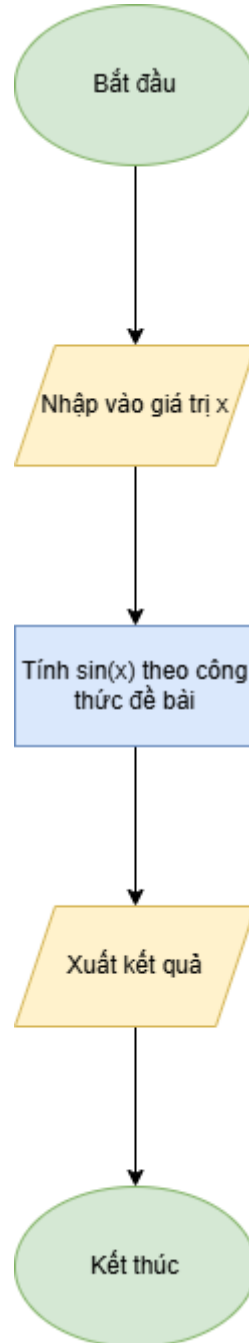


Figure 4 - Flowchart chương trình tính $\sin(x)$

Mô tả đầu vào, đầu ra và hướng giải quyết của chương trình tính toán $\sin(x)$:

- Đầu vào: Giá trị của x được nhập từ bàn phím
- Đầu ra: Kết quả $\sin(x)$ được tính theo công thức đề bài
- Hướng giải quyết: Áp dụng công thức Taylor

IT002 – Lập trình hướng đối tượng

Nội dung code của chương trình tính $\sin(x)$:

```
#include <iostream>
#include <iomanip>
using namespace std;

// Hàm tính giai thừa của một số nguyên dương n
double gthua(int n) {
    double ans = 1.00;

    for (int i = 1; i <= n; ++i)
        ans *= i; // Nhân dồn từ 1 đến n để tính giai thừa

    return ans;
}

// Hàm tính lũy thừa:  $x^n$ 
double mu(double x, int n) {
    double ans = 1.00;

    for (int i = 1; i <= n; ++i)
        ans *= x; // Nhân dồn để tính  $x^n$ 

    return ans;
}

// Hàm tính giá trị theo công thức chuỗi Taylor
double tinh(double x, int n, double s) {
    // Tính toán từng hạng tử theo công thức:  $(-1)^n * (x^{(2n+1)} / (2n+1)!)$ 
    double a = mu(-1, n) * (mu(x, 2 * n + 1) / gthua(2 * n + 1));

    // Cộng dồn vào tổng hiện tại
    double res = s + a;

    // Kiểm tra độ chính xác của kết quả, nếu không thay đổi giá trị thì dừng đệ quy
    double tmp = mu(10, 5); // Tạo hệ số làm tròn để kiểm tra sai số
    if (res * tmp / tmp == s * tmp / tmp)
        return s;

    // Đệ quy tiếp tục tính toán với n tăng lên
    return tinh(x, n + 1, s + a);
}

int main() {
    double x;

    cin >> x; // Nhập giá trị x từ bàn phím

    // Xuất kết quả với 5 chữ số thập phân
```

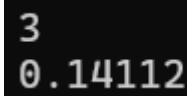
IT002 – Lập trình hướng đối tượng

```
cout << fixed << setprecision(5) << tinh(x, 0, 0);  
  
return 0;  
}
```

Table 4 - Code chương trình tính $\sin(x)$

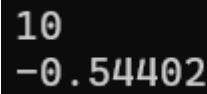
Nội dung kiểm thử của chương trình tính $\sin(x)$:

- Kiểm thử 1: Cho $x = 3$, kết quả = 0.14112.



3
0.14112

- Kiểm thử 2: Cho $x = 10$, kết quả = -0.54402.

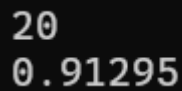


10
-0.54402



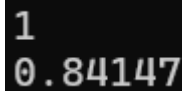
0
0.00000

- Kiểm thử 3: Cho $x = 0$, kết quả = 0.00000.



20
0.91295

- Kiểm thử 4: Cho $x = 20$, kết quả = 0.91295.



1
0.84147

- Kiểm thử 5: Cho $x = 1$, kết quả = 0.84147.

Bài tập 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.
- Tìm xem ngày đó là ngày thứ bao nhiêu trong năm và xuất kết quả.

Nội dung flowchart của chương trình tính ngày được trình bày như hình 5.

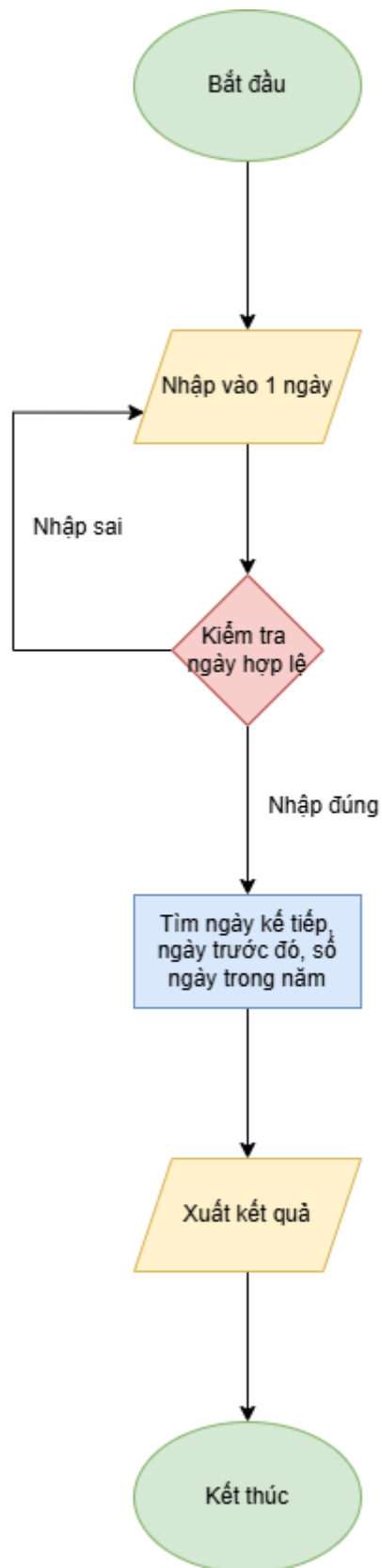


Figure 5 - Flowchart chương trình tính ngày

Mô tả đầu vào, đầu ra và hướng giải quyết của chương trình tính ngày:

- Đầu vào: Một ngày được nhập từ bàn phím theo định dạng ddmmyyyy.
- Đầu ra: Ngày kế tiếp, ngày trước đó, ngày thứ bao nhiêu trong năm.

IT002 – Lập trình hướng đối tượng

- Hướng giải quyết:
 - + Trong năm nhuận, tháng 2 có 29 ngày. Ngược lại, nếu không phải năm nhuận, tháng 2 có 28 ngày.
 - + Các tháng 4, tháng 6, tháng 9 và tháng 11 có 30 ngày.
 - + Các tháng 1, tháng 3, tháng 5, tháng 7, tháng 8, tháng 10, tháng 12 có 31 ngày.
 - + Để tìm ngày kế tiếp chỉ cần cộng 1 ngày vào ngày hiện tại. Nếu ngày đó quá số ngày của tháng hiện tại thì tăng tháng và cập nhật ngày về 1. Nếu tháng quá 12 thì tăng năm và cập nhật tháng về 1.
 - + Để tìm ngày trước đó chỉ cần trừ xuống 1 ngày. Nếu ngày đó = 0 thì giảm tháng và cập nhật ngày về số ngày của tháng sau khi giảm. Nếu tháng đó = 0 thì giảm năm và cập nhật tháng về 12.
 - + Để tính ngày là ngày thứ bao nhiêu trong năm chỉ cần cộng số ngày của các tháng trước đó và số ngày của tháng hiện tại.

Nội dung code của chương trình tính ngày:

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

using namespace std;

//Hàm kiểm tra năm nhuận
int kiemTraNamNhuan(int a){
    if(a%4 == 0 && a%100 != 0 || a%400 == 0) return 1;
    return 0;
}

//Hàm tìm ngày tiếp theo
void findNextDate(int date){
    //Tách ngày, tháng, năm từ giá trị số nguyên date
    (ddmmyyyy)
    int year = date%10000;
    date = date/10000;
    int month = date%100, day = date/100;

    int check = kiemTraNamNhuan(year);

    day = day+1;
    //Xử lý trường hợp ngày vượt quá số ngày của tháng
    if(month == 2 && check == 1){
        if(day > 29){
            day = 1;
            ++month;
        }
    }
    if(month == 2 && check == 0){
        if(day > 28){
            day = 1;
            ++month;
        }
    }
    if(month == 4 || month == 6 || month == 9 || month ==
11){
```

```
        if(day > 30){
            day = 1;
            ++month;
        }
    }
    if(month == 1 || month == 3 || month == 5 || month == 7
|| month == 8 || month == 10 || month == 12){
        if(day > 31){
            day = 1;
            ++month;
        }
    }
    if(month > 12){
        month -= 12;
        ++year;
    }

    cout << "Ngày kế tiếp là: ";
    if(day < 10) cout << "0";
    cout << day << "/";
    if(month < 10) cout << "0";
    cout << month << "/";
    cout << year << "\n";
}

//Hàm tìm ngày trước đó
void findPreviousDate(int date){
    //Tách ngày, tháng, năm từ giá trị số nguyên
    date(ddmmyyyy)
    int year = date%10000;
    date = date/10000;
    int month = date%100, day = date/100;

    int check = kiemTraNamNhuan(year);
    day = day - 1;
    //Nếu ngày giảm về 0, cần chuyển sang tháng trước
    if(day == 0){
        if(month == 2 && check == 1) day = 29;
        if(month == 2 && check == 0) day = 28;
        if(month == 4 || month == 6 || month == 9 || month ==
11) day = 30;
        if(month == 1 || month == 3 || month == 5 || month ==
7 || month == 8 || month == 10 || month == 12) day = 31;
        --month;
        if(month == 0){
            month = 12;
            --year;
        }
    }

    cout << "Ngày trước đó là: ";
    if(day < 10) cout << "0";
    cout << day << "/";
```

```
    if(month < 10) cout << "0";
    cout << month << "/";
    cout << year << "\n";
}

//Hàm đếm ngày của năm (tính xem ngày nhập vào là ngày thứ
mấy trong năm)
void countDateOfYear(int date){
    //Tách ngày, tháng, năm từ giá trị số nguyên
    date(ddmmyyyy)
    int year = date%10000;
    date = date/10000;
    int month = date%100, day = date/100;
    int cntDate = 0, check = kiemTraNamNhuan(year);

    for(int i = 1; i < month; ++i){
        if(i == 2){
            if(check == 1) cntDate += 29;
            else cntDate += 28;
        }
        if(i == 4 || i == 6 || i == 9 || i == 11) cntDate +=
30;
        if(i == 1 || i == 3 || i == 5 || i == 7 || i == 8 ||
i == 10 || i == 12) cntDate += 31;
    }

    cntDate += day;
    cout << "Ngày bạn nhập là ngày thu " << cntDate << "
trong nam." << "\n";
}

//Menu nhập ngày và chọn chương trình mong muốn
void chuongTrinh(){
    int date, n;

    cout << "Nhap vao mot ngay theo dinh dang ddmmyyyy: ";
    cin >> date;

    //Kiểm tra xem ngày nhập vào đã hợp lệ hay chưa
    while(1){
        int tmp = date;
        int year = tmp%10000;
        tmp = tmp/10000;
        int month = tmp%100, day = tmp/100, check =
kiemTraNamNhuan(year);

        if(month == 2){
            if(check == 1 && day >= 1 && day <= 29) break;
            if(check == 0 && day >= 1 && day <= 28) break;
        }
        if(month == 4 || month == 6 || month == 9 || month ==
11){
            if(day >= 1 && day <= 30) break;
        }
    }
}
```

```

        }
        if(month == 1 || month == 3 || month == 5 || month ==
7 || month == 8 || month == 10 || month == 12){
            if(day >= 1 && day <= 31) break;
        }
        cout << "Ngày không hợp lệ. Vui lòng nhập lại: ";
        cin >> date;
    }

    findNextDate(date);
    findPreviousDate(date);
    countDateOfYear(date);
}

int main() {
    chuongTrinh();

    return 0;
}

```

Table 5 - Code chương trình tính ngày

Nội dung kiểm thử cho chương trình tính ngày:

- Kiểm thử 1: Cho ngày = 01012025, kết quả:
 - + Ngày kế tiếp: 02/01/2025
 - + Ngày trước đó: 31/12/2024
 - + Là ngày thứ 1 trong năm

```

Nhập vào một ngày theo định dạng ddmmyyyy: 01012025
Ngày kế tiếp là: 02/01/2025
Ngày trước đó là: 31/12/2024
Ngày bạn nhập là ngày thứ 1 trong năm.

```

- Kiểm thử 2: Cho ngày = 29022025, kết quả: Ngày không hợp lệ.

```

Nhập vào một ngày theo định dạng ddmmyyyy: 29022025
Ngày không hợp lệ. Vui lòng nhập lại: |

```

- Kiểm thử 3: Cho ngày = 31122024, kết quả:
 - + Ngày kế tiếp: 01/01/2025
 - + Ngày trước đó: 30/12/2024
 - + Là ngày thứ 366 trong năm

```

Nhập vào một ngày theo định dạng ddmmyyyy: 31122024
Ngày kế tiếp là: 01/01/2025
Ngày trước đó là: 30/12/2024
Ngày bạn nhập là ngày thứ 366 trong năm.

```

IT002 – Lập trình hướng đối tượng

- Kiểm thử 4: Cho ngày = 04092006, kết quả:
 - + Ngày kế tiếp: 05092006
 - + Ngày trước đó: 03092006

```
Nhap vao mot ngay theo dinh dang ddmmyyyy: 04092006
Ngay ke tiep la: 05/09/2006
Ngay truoc do la: 03/09/2006
Ngay ban nhap la ngay thu 247 trong nam.
```

Bài tập 6: 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.

Nội dung flowchart của chương trình đếm số lần xuất hiện được trình bày như hình 6.

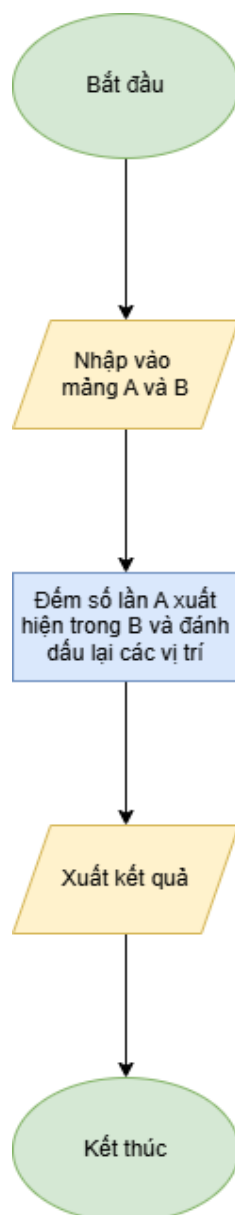


Figure 6 - Flowchart chương trình đếm số lần xuất hiện

Mô tả đầu vào, đầu ra và hướng giải quyết của chương trình đếm số lần xuất hiện:

IT002 – Lập trình hướng đối tượng

- Đầu vào: Số phần tử của mảng A và B, các phần tử của mảng A và B
- Đầu ra: Số lần xuất hiện của A trong B, các vị trí của A bắt đầu xuất hiện trong B.
- Hướng giải quyết:
 - + Duyệt từng phần tử của mảng B từ đầu đến cuối.
 - + Khi gặp một phần tử khớp với phần tử đầu tiên của dãy A, bắt đầu kiểm tra các phần tử tiếp theo để xem có khớp toàn bộ hay không.
 - + Nếu khớp toàn bộ, ghi nhận vị trí bắt đầu của dãy A trong B.
 - + Nếu không khớp, tiếp tục tìm từ vị trí tiếp theo.

Nội dung code của chương trình đếm số lần xuất hiện:

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

using namespace std;

//Hàm đếm số lần xuất hiện của a trong b
int demXuatHien(int a[], int b[], int pos[], int n, int m){
    int cnt, ans, bg;

    bg = -1; //Đánh dấu lại vị trí bắt đầu
    cnt = ans = 0; //cnt: số phần tử khớp, ans: số lần tìm thấy
    for(int i = 0; i < m; ++i){
        if(b[i] == a[cnt]){ //Nếu phần tử hiện tại của b
            //trùng với phần tử cnt của a
            if(bg == -1) bg = i; //Đánh dấu vị trí bắt đầu
            //của dãy a trong b
            ++cnt;
        } else { //Nếu không khớp
            cnt = 0; //Reset biến đếm số phần tử khớp
            if(b[i] == a[cnt]){ //Kiểm tra lại từ đầu
                if(bg == -1) bg = i;
                ++cnt;
            } else bg = -1; //Nếu không khớp thì đặt lại vị
            //trí bắt đầu
        }
        if(cnt == n){ //Nếu đã tìm thấy đầy đủ dãy a trong b
            pos[ans] = bg; //Lưu lại vị trí bắt đầu
            ++ans; //Tăng số lần tìm thấy
            cnt = 0; //Reset biến đếm
            bg = -1; //Reset vị trí bắt đầu
        }
    }

    return ans;
}

int main(){
    int n, m, a[100005], b[100005], pos[100005], cntEqualA =
    1, cntEqualB = 1;

    cin >> n >> m;
    for(int i = 0; i < n; ++i){
```

```

        cin >> a[i];
        if(i != 0 && a[i] == a[0]) ++cntEqualA;
    }
    for(int j = 0; j < m; ++j){
        cin >> b[j];
        if(j != 0 && b[j] == b[0]) ++cntEqualB;
    }

    int ans;
    if(a[0] == b[0] && cntEqualA == n && cntEqualB == m){
        ans = m-n+1;
        for(int i = 0; i < ans; ++i) pos[i] = i;
    } else ans = demXuatHien(a, b, pos, n, m);

    cout << "So lan xuat hien cua A trong B la: " << ans <<
    "\n";
    if(ans != 0){
        cout << "A xuat hien tai cac vi tri: ";
        for(int i = 0; i < ans; ++i) cout << pos[i] << " ";
    }

    return 0;
}

```

Table 6 - Code chương trình đếm số lần xuất hiện

Nội dung kiểm thử cho chương trình đếm số lần xuất hiện:

- Kiểm thử 1: Cho A = [1, 2], B = [1, 2, 1, 2, 3, 1, 2], kết quả: mảng A xuất hiện 3 lần tại các chỉ số bắt đầu là 0, 2 và 5.

```

2 7
1 2
1 2 1 2 3 1 2
So lan xuat hien cua A trong B la: 3
A xuat hien tai cac vi tri: 0 2 5

```

- Kiểm thử 2: Cho A = [2, 2, 2], B = [2, 2, 2, 2, 2, 2, 2], kết quả: mảng A xuất hiện 5 lần tại các chỉ số bắt đầu là 0, 1, 2, 3, 4.

```

3 7
2 2 2
2 2 2 2 2 2 2
So lan xuat hien cua A trong B la: 5
A xuat hien tai cac vi tri: 0 1 2 3 4

```

- Kiểm thử 3: Cho A = [1, 2, 3], B = [1, 2, 1, 2, 1, 2, 4], kết quả: mảng A xuất hiện 0 lần.

```

3 7
1 2 3
1 2 1 2 1 2 4
So lan xuat hien cua A trong B la: 0

```


IT002 – Lập trình hướng đối tượng

Bài tập 7: 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:

- Nhập thông tin cho n học sinh, bao gồm: Họ tên, điểm toán, văn và ngoại ngữ.
- Tính điểm trung bình của mỗi học sinh theo công thức:

$$\text{Điểm trung bình} = \frac{2 \cdot \text{Điểm Toán} + \text{Điểm Văn} + \text{Điểm Anh}}{4}$$

- Phân loại học sinh dựa trên điểm trung bình
- Tìm kiếm và xuất thông tin của học sinh có điểm trung bình cao nhất
- Tìm kiếm học sinh theo tên
- Xuất danh sách học sinh có điểm toán thấp nhất.

Nội dung flowchart của chương trình quản lý học sinh được trình bày như hình 7.

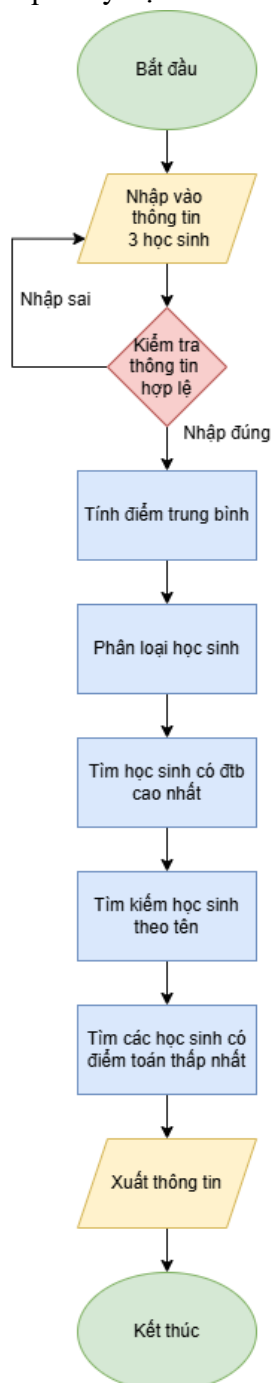


Figure 7 - Flowchart chương trình quản lý học sinh

IT002 – Lập trình hướng đối tượng

Mô tả đầu vào, đầu ra và hướng giải quyết của chương trình quản lý học sinh:

- Đầu vào: Thông tin của 3 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.

Nội dung code của chương trình quản lý học sinh:

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

using namespace std;

//Khai báo cấu trúc lưu thông tin học sinh
struct thôngTinHocSinh{
    string hoTen;
    double diemToan, diemVan, diemNN;
};

//Hàm kiểm tra tên có hợp lệ không (không chứa số)
int kiemTraHopLeTen(string s){
    for(int i = 0; i < s.size(); ++i){
        if('0' <= s[i] && s[i] <= '9') return 0;
    }
    return 1;
}

//Hàm nhập thông tin học sinh
void Nhap(thôngTinHocSinh a[]){
    for(int i = 0; i < 3; ++i){
        cout << "Nhap ho ten cua hoc sinh thu " << i+1 << ": ";

        getline(cin, a[i].hoTen);

        //Kiểm tra tính hợp lệ của tên
        int checkTen = kiemTraHopLeTen(a[i].hoTen);
        while(checkTen == 0){
            cout << "Ten khong hop le. Vui long nhap lai: ";
            getline(cin, a[i].hoTen);
            checkTen = kiemTraHopLeTen(a[i].hoTen);
        }

        //Nhập điểm Toán và kiểm tra hợp lệ
        cout << "Diem toan cua hoc sinh thu " << i+1 << ": ";
        cin >> a[i].diemToan;
        while(a[i].diemToan < 0 || a[i].diemToan > 10){
            cout << "Diem toan khong hop le. Vui long nhap lai: ";
            cin >> a[i].diemToan;
        }
    }
}
```

```
//Nhập điểm Văn và kiểm tra hợp lệ
cout << "Diem van cua hoc sinh thu " << i+1 << ": ";
cin >> a[i].diemVan;
while(a[i].diemVan < 0 || a[i].diemVan > 10){
    cout << "Diem van khong hop le. Vui long nhap
lai: ";
    cin >> a[i].diemVan;
}

//Nhập điểm Ngoại ngữ và kiểm tra hợp lệ
cout << "Diem ngoai ngu cua hoc sinh thu " << i+1 <<
": "; cin >> a[i].diemNN;
while(a[i].diemNN < 0 || a[i].diemNN > 10){
    cout << "Diem ngoai ngu khong hop le. Vui long
nhap lai: ";
    cin >> a[i].diemNN;
}

cin.ignore(); //Xóa bộ đệm trước khi nhập chuỗi tiếp
theo
}
}

//Hàm tìm kiếm học sinh theo tên
void timKiemTen(thongTinHocSinh a[], string s){
    int cnt = 0;

    //Chuyển từ khóa tìm kiếm sang chữ thường để tìm kiếm
    không phân biệt hoa thường
    transform(s.begin(), s.end(), s.begin(), ::tolower);
    for(int i = 0; i < 3; ++i){
        string tmp = a[i].hoTen;

        //Chuyển họ tên sang chữ thường để tìm kiếm không
        phân biệt hoa thường
        transform(tmp.begin(), tmp.end(), tmp.begin(),
        ::tolower);
        if(tmp.find(s) != string::npos){
            ++cnt;
            cout << a[i].hoTen << ": Diem toan: " <<
a[i].diemToan << " Diem van: " << a[i].diemVan << " Diem
NN: " << a[i].diemNN << "\n";
        }
    }

    if(cnt == 0) cout << "Ten khong co trong danh sach" <<
"\n";
}

//Hàm liệt kê danh sách học sinh có điểm Toán thấp nhất trong
3 môn
```

```

void locDanhSachToanMin(thongTinHocSinh a[]){
    int cnt = 0;
    for(int i = 0; i < 3; ++i){
        if(a[i].diemToan < a[i].diemVan && a[i].diemToan <
a[i].diemNN){
            cout << a[i].hoTen << "\n";
            ++cnt;
        }
    }

    if(cnt == 0) cout << "Khong co hoc sinh nao co diem toan
thap nhat" << "\n";
}

//Hàm xuất thông tin học sinh
void Xuat(thongTinHocSinh a[]){
    string nameNeedFind;
    double mxDiemTrungBinh = 0;
    int posMaxDiem = 0;

    //Xuất điểm trung bình và phân loại học sinh
    for(int i = 0; i < 3; ++i){
        double diemTrungBinh =
((2*a[i].diemToan)+a[i].diemVan+a[i].diemNN)/4;
        cout << "Diem trung binh cua hoc sinh thu " << i+1 <<
": " << diemTrungBinh << "\n";

        cout << "Phan loai cua hoc sinh thu " << i+1 << ": ";
        if(diemTrungBinh < 5) cout << "Yeu" << "\n";
        else if(diemTrungBinh < 6.5) cout << "Trung binh" <<
"\n";

        else if(diemTrungBinh < 8) cout << "Kha" << "\n";
        else if(diemTrungBinh < 9) cout << "Gioi" << "\n";
        else cout << "Xuat sac" << "\n";

        if(diemTrungBinh > mxDiemTrungBinh){
            mxDiemTrungBinh = diemTrungBinh;
            posMaxDiem = i;
        }
    }

    //Xuất học sinh có điểm trung bình cao nhất
    cout << "Hoc sinh co diem trung binh cao nhat la: " <<
a[posMaxDiem].hoTen << "\n";
    //Tìm kiếm học sinh theo tên
    cout << "Nhap ten hoc sinh ban muon tim kiem: ";
    getline(cin, nameNeedFind);
    timKiemTen(a, nameNeedFind);
    while(1){
        cout << "Ban muon tim kiem hoc sinh nao khac khong.
Neu co hay nhap ten, nguoc lai nhap 0 de ket thuc: ";
        getline(cin, nameNeedFind);
        if(nameNeedFind == "0") break;
    }
}

```

```

        timKiemTen(a, nameNeedFind);
    }

    //Xuất danh sách học sinh có điểm toán thấp nhất
    cout << "Danh sach hoc sinh co diem toan thap nhat: ";
    locDanhSachToanMin(a);
}

int main() {
    thôngTinHocSinh a[5];
    Nhập(a);
    Xuất(a);

    return 0;
}

```

*Table 7 - Code chương trình quản lý học sinh***Nội dung kiểm thử cho chương trình quản lý học sinh**

- Kiểm thử 1:
 - + Học sinh 1: Đoàn Nguyễn Minh Thư, điểm toán = 8.9, điểm văn = 9.5, điểm ngoại ngữ = 9.3.
 - + Học sinh 2: Phan Thị Như Huỳnh, điểm toán = 9.3, điểm văn = 7.6, điểm ngoại ngữ = 6.7.
 - + Học sinh 3: Huỳnh Phạm Bảo Trúc, điểm toán = 7.6, điểm văn = 5.5, điểm ngoại ngữ = 9.0.
 - + Học sinh cần tìm kiếm: Trúc
 - + Kết quả:
 - Điểm tb học sinh 1: 9.15, xếp loại xuất sắc
 - Điểm tb học sinh 2: 8.225, xếp loại giỏi
 - Điểm tb học sinh 3: 7.425, xếp loại khá
 - Học sinh có điểm tb cao nhất là: Đoàn Nguyễn Minh Thư
 - Học sinh tìm kiếm: Huỳnh Phạm Bảo Trúc
 - Danh sách học sinh có điểm toán thấp nhất: Đoàn Nguyễn Minh Thư

```

Nhập họ tên của học sinh thu 1: Doan Nguyen Minh Thu
Diem toan của học sinh thu 1: 8.9
Diem van của học sinh thu 1: 9.5
Diem ngoai ngu của học sinh thu 1: 9.3
Nhập họ tên của học sinh thu 2: Phan Thi Nhu Huynh
Diem toan của học sinh thu 2: 9.3
Diem van của học sinh thu 2: 7.6
Diem ngoai ngu của học sinh thu 2: 6.7
Nhập họ tên của học sinh thu 3: Huynh Pham Bao Truc
Diem toan của học sinh thu 3: 7.6
Diem van của học sinh thu 3: 5.5
Diem ngoai ngu của học sinh thu 3: 9.0
Diem trung binh của học sinh thu 1: 9.15
Phan loai của học sinh thu 1: Xuat sac
Diem trung binh của học sinh thu 2: 8.225
Phan loai của học sinh thu 2: Gioi
Diem trung binh của học sinh thu 3: 7.425
Phan loai của học sinh thu 3: Kha
Hoc sinh co diem trung binh cao nhat la: Doan Nguyen Minh Thu
Nhập tên học sinh bạn muốn tìm kiếm: Truc
Huynh Pham Bao Truc: Diem toan: 7.6 Diem van: 5.5 Diem NN: 9
Bạn muốn tìm kiếm học sinh nào khác không. Nếu có hãy nhập tên, ngược lại nhập 0 để kết thúc: 0
Danh sach hoc sinh co diem toan thap nhat: Doan Nguyen Minh Thu

```

IT002 – Lập trình hướng đối tượng

Bài tập 8: Xây dựng 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 chứa khoảng trắng hay ký tự đặc biệt.
- Ngày bay: Ngày phải tồn tại và hợp lệ theo lịch.
- Giờ bay: Dữ liệu giờ khởi hành được nhập theo định dạng 24 giờ (định dạng hhmm từ 0000 → 2359)
- Nơi đi: Chuỗi ký tự dài tối đa 20 ký tự, không được chứa số hoặc ký tự đặc biệt.
- Nơi đến: Chuỗi ký tự dài tối đa 20 ký tự, không được chứa số hoặc ký tự đặc biệt.

Nội dung flowchart của chương trình quản lý thông tin chuyến bay được trình bày như hình 8.

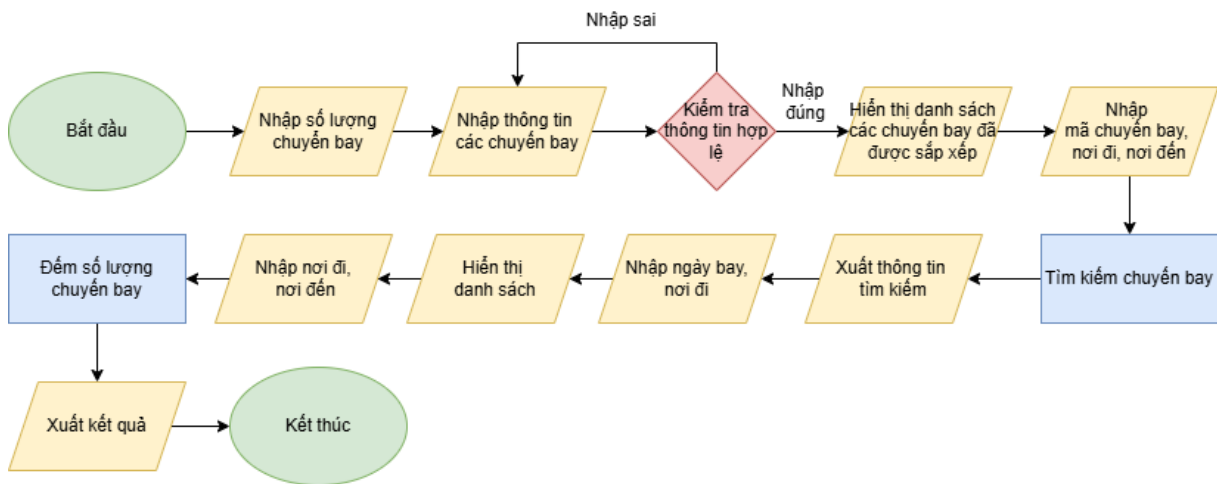


Figure 8 - Flowchart chương trình quản lý thông tin chuyến bay

Mô tả đầu vào, đầu ra và hướng giải quyết của chương trình quản lý thông tin chuyến bay:

- 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 kết quả theo yêu cầu.
- Hướng giải quyết:
 - + Lưu trữ thông tin chuyến bay: Sử dụng mảng để lưu danh sách chuyến bay, mỗi chuyến bay được biểu diễn bằng cấu trúc chuyếnBay chứa mã chuyến, ngày bay, giờ bay, nơi đi và nơi đến.
 - + Kiểm tra tính hợp lệ: Sử dụng các hàm isalpha(), isdigit(), isspace() để kiểm tra tính hợp lệ của các thông tin.
 - + Xử lý nhập liệu và tìm kiếm: Sử dụng vòng lặp và điều kiện để nhập dữ liệu, kiểm tra tính hợp lệ và tìm kiếm chuyến bay theo yêu cầu của người dùng.

IT002 – Lập trình hướng đối tượng

Nội dung code của chương trình quản lý thông tin chuyến bay:

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

using namespace std;

//Định nghĩa cấu trúc cho chuyến bay
struct chuyenBay{
    string maChuyen, noiDi, noiDen;
    int ngayBay, gioBay;
};

//Định nghĩa cấu trúc để so sánh thời gian các chuyến bay
struct soSanh{
    int year, month, day, hour, minute;
};

//Hàm kiểm tra tính hợp lệ của mã chuyến bay
int kiemTraHopLeMaChuyen(string s){
    for(int i = 0; i < s.size(); ++i){
        if(!isalpha(s[i]) && !isdigit(s[i])) return 0;
        if(isspace(s[i])) return 0;
    }
    return 1;
}

//Hàm kiểm tra năm nhuận
int kiemTraNamNhuan(int a){
    if(a%4 == 0 && a%100 != 0 || a%400 == 0) return 1;
    return 0;
}

//Hàm kiểm tra tính hợp lệ của ngày bay (định dạng ddmmyyyy)
int kiemTraHopLeNgay(int a){
    int year = a%10000;
    a = a/10000;
    int month = a%100, day = a/100;
    int check = kiemTraNamNhuan(year);

    if(day <= 0) return 0;
    if(month == 2){
        if(check == 1 && day <= 29) return 1;
        if(check == 0 && day <= 28) return 1;
    }
    if(month == 4 || month == 6 || month == 9 || month == 11){
        if(day <= 30) return 1;
    }
    if(month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month == 12){
        if(day <= 31) return 1;
    }
    return 0;
}
```

```

}

//Hàm kiểm tra tính hợp lệ của địa danh
int kiemTraHopLeDiaDanh(string s){
    for(int i = 0; i < s.size(); ++i){
        if(isdigit(s[i])) return 0;
        if(!isdigit(s[i]) && !isalpha(s[i]) &&
!isspace(s[i])) return 0;
    }
    return 1;
}

//Hàm nhập thông tin các chuyến bay
void Nhap(chuyenBay a[], int &n){
    cout << "So chuyen bay: "; cin >> n;

    cin.ignore();
    for(int i = 0; i < n; ++i){
        cout << "Ma chuyen bay: ";
        getline(cin, a[i].maChuyen);
        while(a[i].maChuyen.size() > 5 ||
kiemTraHopLeMaChuyen(a[i].maChuyen) == 0){
            cout << "Ma chuyen khong hop le. Vui long nhap
lai: ";
            getline(cin, a[i].maChuyen);
        }

        cout << "Ngay bay (nhap theo dinh dang ddmmyyyy): ";
cin >> a[i].ngayBay;
        while(kiemTraHopLeNgay(a[i].ngayBay) == 0){
            cout << "Ngay bay khong hop le. Vui long nhap
lai: ";
            cin >> a[i].ngayBay;
        }

        cout << "Gio bay (nhap theo dinh dang hhmm tu 0000
den 2359): "; cin >> a[i].gioBay;
        while(a[i].gioBay < 0 || a[i].gioBay%100 >= 60 ||
a[i].gioBay > 2359){
            cout << "Gio bay khong hop le. Vui long nhap lai:
";
            cin >> a[i].gioBay;
        }

        cin.ignore();
        cout << "Noi di: "; getline(cin, a[i].noiDi);
        while(a[i].noiDi.size() > 20 ||
kiemTraHopLeDiaDanh(a[i].noiDi) == 0){
            cout << "Noi di khong hop le. Vui long nhap lai:
";
            getline(cin, a[i].noiDi);
        }
    }
}

```



```
        cout << "Noi den: "; getline(cin, a[i].noiDen);
        while(a[i].noiDen.size() > 20 ||
kiemTraHopLeDiaDanh(a[i].noiDen) == 0){
            cout << "Noi den khong hop le. Vui long nhap lai:
";
            getline(cin, a[i].noiDen);
        }
    }
}

//Hàm xuất thông tin một chuyến bay
void Xuat(chuyenBay a){
    cout << a.maChuyen << " " << a.ngayBay << " " << a.gioBay
<< " " << a.noiDi << "-->" << a.noiDen << "\n";
}

//Hàm so sánh hai chuyến bay để sắp xếp theo thời gian
bool cmp(chuyenBay x, chuyenBay y){
    soSanh tmp, tmp1;

    //Trích xuất thông tin từ x
    tmp.year = x.ngayBay%10000;
    x.ngayBay /= 10000;
    tmp.month = x.ngayBay%100;
    tmp.day = x.ngayBay/100;
    tmp.hour = x.gioBay/100;
    tmp.minute = x.gioBay%100;

    //Trích xuất thông tin từ y
    tmp1.year = y.ngayBay%10000;
    y.ngayBay /= 10000;
    tmp1.month = y.ngayBay%100;
    tmp1.day = y.ngayBay/100;
    tmp1.hour = y.gioBay/100;
    tmp1.minute = y.gioBay%100;

    //So sánh theo thứ tự: năm, tháng, ngày, giờ, phút
    if(tmp.year < tmp1.year) return true;
    if(tmp.year == tmp1.year && tmp.month < tmp1.month)
return true;
    if(tmp.year == tmp1.year && tmp.month == tmp1.month &&
tmp.day < tmp1.day) return true;
    if(tmp.year == tmp1.year && tmp.month == tmp1.month &&
tmp.day == tmp1.day && tmp.hour < tmp1.hour) return true;
    if(tmp.year == tmp1.year && tmp.month == tmp1.month &&
tmp.day == tmp1.day && tmp.hour == tmp1.hour && tmp.minute <
tmp1.minute)return true;

    return false;
}

//Hàm sắp xếp và hiển thị danh sách sau sắp xếp
void sapXepChuyenBay(chuyenBay a[], int n){
```

```

        sort(a, a+n, cmp);

        cout << "Danh sach chuyen bay da sap xep la: " << "\n";
        for(int i = 0; i < n; ++i) Xuat(a[i]);
    }

//Hàm tìm kiếm chuyến bay theo mã, nơi đi hoặc nơi đến
void timKiemChuyenBay(chuyenBay a[], int n){
    int cnt = 0;
    string nameFind, tmpChuyen, tmpDi, tmpDen;

    cin.ignore();
    cout << "Nhap thong tin chuyen bay ban muon timkiem (ma
chuyen bay, noi di hoac noi den): ";
    getline(cin, nameFind);
    transform(nameFind.begin(), nameFind.end(),
nameFind.begin(), ::tolower);

    cout << "Chuyen bay ma ban timkiem la: ";
    for(int i = 0; i < n; ++i){
        tmpChuyen = a[i].maChuyen;
        tmpDi = a[i].noiDi;
        tmpDen = a[i].noiDen;
        transform(tmpChuyen.begin(), tmpChuyen.end(),
tmpChuyen.begin(), ::tolower);
        transform(tmpDi.begin(), tmpDi.end(), tmpDi.begin(),
::tolower);
        transform(tmpDen.begin(), tmpDen.end(),
tmpDen.begin(), ::tolower);

        if(tmpChuyen.find(nameFind) != string::npos ||
tmpDi.find(nameFind) != string::npos || tmpDen.find(nameFind)
!= string::npos){
            ++cnt;
            Xuat(a[i]);
        }
    }
    if(cnt == 0) cout << "Khong co chuyen bay phu hop" <<
"\n";
}

//Hàm hiển thị danh sách chuyến bay theo tìm kiếm ngay và nơi
đi
void hienThiDanhSach(chuyenBay a[], int n){
    int dateSelect;
    string placeSelect;

    cout << "Nhap ngay ban muon timkiem (theo dinh dang
ddmmyyyy): ";
    cin >> dateSelect;
    while(kiemTraHopLeNgay(dateSelect) == 0){
        cout << "Ngay ban timkiem khong hop le. Vui long
nhap lai: ";

```

```

        cin >> dateSelect;
    }

    cin.ignore();
    cout << "Nhap diem di cua chuyen bay ban muon tim kiem: ";
    getline(cin, placeSelect);
    while(kiemTraHopLeDiaDanh(placeSelect) == 0){
        cout << "Diem di ban tim kiem khong hop le. Vui long
nhap lai: ";
        getline(cin, placeSelect);
    }

    transform(placeSelect.begin(), placeSelect.end(),
placeSelect.begin(), ::tolower);

    int cnt = 0;
    string tmp;
    cout << "Cac chuyen bay theo tim kiem la: ";
    for(int i = 0; i < n; ++i){
        tmp = a[i].noiDi;
        transform(tmp.begin(), tmp.end(), tmp.begin(),
::tolower);
        if(a[i].ngayBay == dateSelect &&
tmp.find(placeSelect) != string::npos){
            Xuat(a[i]);
            ++cnt;
        }
    }
    if(cnt == 0) cout << "Khong co" << "\n";
}

//Hàm đếm số lượng chuyến bay từ nơi đi đến nơi đến theo tìm
kiếm
void demSoLuongChuyenBay(chuyenBay a[], int n){
    int ans = 0;
    chuyenBay x;

    cin.ignore();
    cout << "Ban muon dem so chuyen bay tu dau den dau: " <<
"\n";
    cout << "Noi di: "; getline(cin, x.noiDi);
    while(kiemTraHopLeDiaDanh(x.noiDi) == 0){
        cout << "Noi di ban tim kiem khong hop le. Vui long
nhap lai: ";
        getline(cin, x.noiDi);
    }

    cout << "Noi den: "; getline(cin, x.noiDen);
    while(kiemTraHopLeDiaDanh(x.noiDen) == 0){
        cout << "Noi den ban tim kiem khong hop le. Vui long
nhap lai: ";
        getline(cin, x.noiDen);
    }
}

```

```

    }

    string tmpXDi = x.noiDi, tmpXDen = x.noiDen, tmpADi,
tmpADen;
    transform(tmpXDi.begin(), tmpXDi.end(), tmpXDi.begin(),
::tolower);
    transform(tmpXDen.begin(), tmpXDen.end(),
tmpXDen.begin(), ::tolower);

    for(int i = 0; i < n; ++i){
        tmpADi = a[i].noiDi;
        tmpADen = a[i].noiDen;
        transform(tmpADi.begin(), tmpADi.end(),
tmpADi.begin(), ::tolower);
        transform(tmpADen.begin(), tmpADen.end(),
tmpADen.begin(), ::tolower);
        if(tmpADi.find(tmpXDi) != string::npos &&
tmpADen.find(tmpXDen) != string::npos) ++ans;
    }

    cout << "So luong chuyen bay tu " << x.noiDi << "-->" <<
x.noiDen << " la: " << ans << "\n";
}

//Hàm menu để người dùng chọn chức năng
void Xuat(chuyenBay a[], int n){
    timKiemChuyenBay(a, n);
    hienThiDanhSach(a, n);
    demSoLuongChuyenBay(a, n);
}

int main() {
    int n;
    chuyenBay a[105];

    Nhap(a, n);
    sapXepChuyenBay(a, n);
    Xuat(a, n);

    return 0;
}

```

Table 8 - Code chương trình quản lý thông tin chuyến bay

IT002 – Lập trình hướng đối tượng

Nội dung kiểm thử cho chương trình quản lý thông tin chuyến bay:

- Kiểm thử 1:

```
So chuyen bay: 2
Ma chuyen bay: VNA52
Ngay bay (nhap theo dinh dang ddmmyyyy): 25032025
Gio bay (nhap theo dinh dang hhmm tu 0000 den 2359): 2250
Noi di: Can Tho
Noi den: Ho Chi Minh
Ma chuyen bay: VNF52
Ngay bay (nhap theo dinh dang ddmmyyyy): 04092024
Gio bay (nhap theo dinh dang hhmm tu 0000 den 2359): 2240
Noi di: Ho Chi Minh
Noi den: Campuchia
Danh sach chuyen bay da sap xep la:
VNF52 4092024 2240 Ho Chi Minh-->Campuchia
VNA52 25032025 2250 Can Tho-->Ho Chi Minh

Nhap thong tin chuyen bay ban muon tim kiem (ma chuyen bay, noi di hoac noi den): VNA52
Chuyen bay ma ban tim kiem la: VNA52 25032025 2250 Can Tho-->Ho Chi Minh
Nhap ngay ban muon tim kiem (theo dinh dang ddmmyyyy): 04092024
Nhap diem di cua chuyen bay ban muon tim kiem: Can Tho
Cac chuyen bay theo tim kiem la: Khong co

Ban muon dem so chuyen bay tu dau den dau:
Noi di: Ho chi minh
Noi den: cam
So luong chuyen bay tu Ho chi minh-->cam la: 1
```

Bài tập 9: 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
- 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 có đúng 9 hoặc 12 chữ số
- Ngày mở sổ: Ngày phải tồn tại theo lịch
- Số tiền gửi: Số dương

Nội dung flowchart của chương trình quản lý sổ tiết kiệm được trình bày như hình 9.

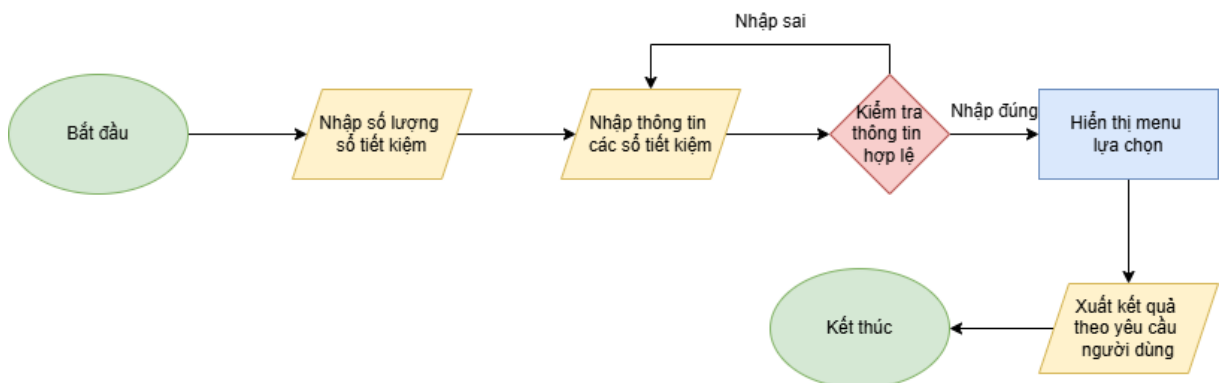


Figure 9 - Flowchart chương trình quản lý sổ tiết kiệm

IT002 – Lập trình hướng đối tượng

Mô tả đầu vào, đầu ra và hướng giải quyết của chương trình quản lý sổ tiết kiệm:

- Đầu vào: Thông tin sổ tiết kiệm bao gồm: Mã sổ, loại tiết kiệm, họ tên khách hàng, chứng minh nhân dân, ngày mở sổ, số tiền gửi.
- Đầu ra:
 - + Thông tin sổ tiết kiệm
 - + Thông tin dựa vào yêu cầu của người dùng
- Hướng giải quyết:
 - + Lưu trữ thông tin sổ tiết kiệm: Sử dụng mảng để lưu thông tin sổ tiết kiệm được biểu diễn cấu trúc dữ liệu struct.
 - + Kiểm tra tính hợp lệ: Sử dụng các hàm isalpha(), isdigit(), isspace(), chiều dài của chuỗi, để kiểm tra tính hợp lệ của các thông tin.
 - + Xử lý nhập liệu và yêu cầu của người dùng: Sử dụng vòng lặp và điều kiện để nhập dữ liệu, kiểm tra tính hợp lệ và tìm kiếm, sắp xếp, cập nhật theo yêu cầu.

Nội dung code của chương trình quản lý sổ tiết kiệm:

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

using namespace std;

//Cấu trúc lưu trữ thông tin sổ tiết kiệm
struct soTietKiem{
    string maSo, loaiTietKiem, hoTenKH, CMND;
    int ngayMoSo;
    double soTien;
    double laiSuat;
};

//Hàm xuất thông tin một sổ tiết kiệm
void Xuat(soTietKiem a){
    cout << a.maSo << " " << a.loaiTietKiem << " " <<
a.hoTenKH << " ";
    cout << a.CMND << " " << a.ngayMoSo << " " << a.soTien <<
"\n";
}

//Hàm kiểm tra xem chuỗi có chứa ký tự đặc biệt không
int kiemTraKyTuDacBiet(string s){
    for(int i = 0; i < s.size(); ++i){
        if(!isdigit(s[i]) && !isalpha(s[i]) &&
!isspace(s[i])) return 1;
    }
    return 0;
}

//Hàm kiểm tra chuỗi có chứa số không
int kiemTraSo(string s){
    for(int i = 0; i < s.size(); ++i){
        if(isdigit(s[i])) return 1;
    }
    return 0;
}
```

```
//Hàm kiểm tra chuỗi có khoảng trắng không
int kiemTraKhoangTrang(string s){
    for(int i = 0; i < s.size(); ++i){
        if(isspace(s[i])) return i;
    }
    return 0;
}

//Hàm kiểm tra năm nhuận
int kiemTraNamNhuan(int a){
    if(a%4 == 0 && a%100 != 0 || a%400 == 0) return 1;
    return 0;
}

//Hàm kiểm tra tính hợp lệ của ngày
int kiemTraNgay(int date){
    int tmp = date;
    int year = tmp%10000;
    tmp = tmp/10000;
    int month = tmp%100, day = tmp/100, check =
    kiemTraNamNhuan(year);

    //Kiểm tra số ngày trong từng tháng, có tính đến năm
    nhuận
    if(month == 2){
        if(check == 1 && day >= 1 && day <= 29) return 1;
        if(check == 0 && day >= 1 && day <= 28) return 1;
    }
    if(month == 4 || month == 6 || month == 9 || month ==
    11){
        if(day >= 1 && day <= 30) return 1;
    }
    if(month == 1 || month == 3 || month == 5 || month == 7
    || month == 8 || month == 10 || month == 12){
        if(day >= 1 && day <= 31) return 1;
    }

    return 0;
}

//Hàm nhập thông tin cho các sổ tiết kiệm
void Nhap(soTietKiem a[], int &n){
    cout << "So luong so tietkiem: "; cin >> n;

    cin.ignore();
    for(int i = 0; i < n; ++i){
        cout << "Ma so: ";
        getline(cin, a[i].maSo);
        while(a[i].maSo.size() > 5 || kiemTraKyTuDacBiet
        (a[i].maSo) == 1 || kiemTraKhoangTrang(a[i].maSo) == 1){
            cout << "Ma so vua nhap khong hop le. Vui long
            nhap lai: ";
        }
    }
}
```

```

        getline(cin, a[i].maSo);
    }

    cout << "Loai tiet kiem: ";
    getline(cin, a[i].loaiTietKiem);
    while(a[i].loaiTietKiem.size() > 10){
        cout << "Loai tiet kiem vua nhap khong hop le.
Vui long nhap lai: ";
        getline(cin, a[i].loaiTietKiem);
    }

    cout << "Ho ten KH: ";
    getline(cin, a[i].hoTenKH);
    while(a[i].hoTenKH.size() > 30 ||
kiemTraSo(a[i].hoTenKH) == 1 || kiemTraKyTuDacBiet
(a[i].hoTenKH) == 1){
        cout << "Ho ten vua nhap khong hop le. Vui long
nhap lai: ";
        getline(cin, a[i].hoTenKH);
    }

    cout << "Chung minh nhan dan: ";
    getline(cin, a[i].CMND);
    while(a[i].CMND.size() != 9 && a[i].CMND.size() !=
12){
        cout << "CMND khong hop le. Vui long nhap lai: ";
        getline(cin, a[i].CMND);
    }

    cout << "Ngay mo so (theo dinh dang ddmmyyyy): ";
    cin >> a[i].ngayMoSo;
    while(kiemTraNgay(a[i].ngayMoSo) == 0){
        cout << "Ngay vua nhap khong hop le. Vui long
nhap lai: ";
        cin >> a[i].ngayMoSo;
    }

    cout << "So tien gui: ";
    cin >> a[i].soTien;
    while(a[i].soTien < 0){
        cout << "So tien khong hop le. Vui long nhap lai:
";
        cin >> a[i].soTien;
    }

    a[i].laiSuat = 0;

    cin.ignore();
}
}

//Hàm kiểm tra mã số có tồn tại trong mảng hay không
int isValid(string maSo, soTietKiem a[], int n){

```



```
for(int i = 0; i < n; ++i){
    if(a[i].maSo == maSo) return i;
}
return -1;
}
//Hàm cập nhật lãi suất cho một sổ tiết kiệm
void capNhatLaiSuat(soTietKiem a[], int n){
    string maSo;

    cin.ignore();
    cout << "Nhap ma so tai khoan: ";
    getline(cin, maSo);

    while(isValid(maSo, a, n) == -1){
        cout << "Tai khoan khong ton tai. Vui long nhap lai
hoac go 'exit' de ket thuc chuong trinh: ";
        cin >> maSo;
        if(maSo == "exit") return;
    }

    int pos = isValid(maSo, a, n);

    double laiSuat;
    cout << "Nhap lai suat ban muon cap nhat: "; cin >>
laiSuat;
    while(laiSuat < 0){
        cout << "Lai suat khong hop le. Vui long nhap lai: ";
        cin >> laiSuat;
    }

    a[pos].laiSuat = laiSuat;

    cout << "Lai suat duoc cap nhat thanh cong cho tai khoan:
" << maSo << "\n";
}

//Hàm tính toán tiền lãi cho một sổ tiết kiệm
void tinhToanTienLai(soTietKiem a[], int n){
    string maSo;

    cin.ignore();
    cout << "Nhap ma so tai khoan: ";
    getline(cin, maSo);

    while(isValid(maSo, a, n) == -1){
        cout << "Tai khoan khong ton tai. Vui long nhap lai
hoac go 'exit' de ket thuc chuong trinh: ";
        cin >> maSo;
        if(maSo == "exit") return;
    }

    int pos = isValid(maSo, a, n);
```

```

    time_t now = time(0);
    tm *ltm = localtime(&now);
    int nowYear = 1900 + ltm->tm_year;
    int nowMonth = 1 + ltm->tm_mon;
    int nowNgay = ltm->tm_mday;

    int tmp = a[pos].ngayMoSo;
    int moSoYear = tmp%10000;
    tmp /= 10000;
    int moSoMonth = tmp%100;
    int moSoDay = tmp/100;

    int soNgay = (nowYear - moSoYear) * 365 + (nowMonth -
moSoMonth) * 30 + (nowNgay - moSoDay);
    double tienLai = (a[pos].soTien * soNgay *
a[pos].laiSuat) / 100;

    cout << "So tien lai cua tai khoan " << maSo << " hien
tai la: " << tienLai << "\n";
}

//Hàm rút tiền từ sổ tiết kiệm
void rutTien(soTietKiem a[], int n){
    string maSo;

    cin.ignore();
    cout << "Nhap ma so tai khoan: ";
    getline(cin, maSo);

    while(isValid(maSo, a, n) == -1){
        cout << "Tai khoan khong ton tai. Vui long nhap lai
hoac go 'exit' de ket thuc chuong trinh: ";
        cin >> maSo;
        if(maSo == "exit") return;
    }

    int pos = isValid(maSo, a, n);

    double tienRut;
    cout << "So tien ban muon rut la: "; cin >> tienRut;
    while(tienRut < 0 || a[pos].soTien < tienRut){
        cout << "So tien vua nhap khong hop lai. Vui long
nhap lai hoac go 0 de ket thuc chuong trinh: ";
        cin >> tienRut;
        if(tienRut == 0) return;
    }

    a[pos].soTien -= tienRut;
    cout << "Rut tien thanh cong. Cam on quy khach" << "\n";
}

//Hàm tìm kiếm sổ tiết kiệm theo CMND hoặc mã số
void timKiem(soTietKiem a[], int n){

```

```

int option;

cout << "Ban muon tim kiem theo CMND hay ma so" << "\n";
cout << "Nhan 1 de tim kiem theo CMND" << "\n";
cout << "Nhan 2 de tim kiem theo ma so" << "\n";
cout << "Su lua chon cua ban la: ";
cin >> option;

cin.ignore();
if(option == 1){
    string CMND;

    cout << "Nhap CMND ban muon tim kiem: ";
    getline(cin, CMND);
    while(CMND.size() != 9 && CMND.size() != 12){
        cout << "CMND ban vua nhap khong hop le. Vui long
nhap lai hoac go 'exit' de ket thuc chuong trinh: ";
        cin >> CMND;
        if(CMND == "exit") return;
    }

    int cnt = 0;
    cout << "So ban tim kiem la: ";
    for(int i = 0; i < n; ++i){
        if(a[i].CMND == CMND){
            ++cnt;
            Xuat(a[i]);
        }
    }
    if(cnt == 0) cout << "Khong co so phu hop" << "\n";
} else {
    string maSo;

    cout << "Nhap ma so ban muon tim kiem: ";
    getline(cin, maSo);
    while(maSo.size() > 5 || kiemTraKyTuDacBiet (maSo) ==
1 || kiemTraKhoangTrang(maSo) == 1){
        cout << "Ma so vua nhap khong hop le. Vui long
nhap lai hoac go 'exit' de ket thuc chuong trinh: ";
        getline(cin, maSo);
        if(maSo == "exit") return;
    }

    int cnt = 0;
    cout << "So ban tim kiem la: ";
    for(int i = 0; i < n; ++i){
        if(a[i].maSo == maSo){
            ++cnt;
            Xuat(a[i]);
        }
    }
    if(cnt == 0) cout << "Khong co so phu hop" << "\n";
}
}

```

```

}

//Hàm liệt kê các sổ tiết kiệm được mở vào một ngày cụ thể
void lietKe(soTietKiem a[], int n){
    int date;

    cout << "Ban muon tim so mo trong thoi gian nao (nhap
theo dinh dang ddmmyyyy): ";
    cin >> date;
    while(kiemTraNgay(date) == 0){
        cout << "Ngay ban vua nhap khong hop le. Vui long
nhap lai hoac nhap 0 de ket thuc chuong trinh: ";
        cin >> date;
        if(date == 0) return;
    }

    cout << "So theo ban timkiem la: ";
    int cnt = 0;
    for(int i = 0; i < n; ++i){
        if(a[i].ngayMoSo == date){
            ++cnt;
            Xuat(a[i]);
        }
    }

    if(cnt == 0) cout << "Khong co so phu hop" << "\n";
}

//So sánh để sắp xếp theo số tiền tăng dần
int cmp(soTietKiem x, soTietKiem y){
    return x.soTien < y.soTien;
}

//So sánh để sắp xếp theo ngày mở sổ giảm dần
int cmp1(soTietKiem x, soTietKiem y){
    return x.ngayMoSo > y.ngayMoSo;
}

//Hàm sắp xếp danh sách sổ tiết kiệm
void sapXep(soTietKiem a[], int n){
    int option;

    cout << "Nhap 1 neu ban muon sap xep theo so tien gui
giam dan" << "\n";
    cout << "Nhap 2 neu ban muon sap xep theo ngay mo so tang
dan" << "\n";

    cin >> option;
    if(option == 1) sort(a, a+n, cmp);
    else sort(a, a+n, cmp1);

    cout << "Danh sach sau khi sap xep la: " << "\n";
    for(int i = 0; i < n; ++i) Xuat(a[i]);
}

```

```

}
//Chương trình điều khiển các chức năng
void chuongTrinh(soTietKiem a[], int n){
    int option;

    cout << "Moi ban chon chuong trinh: " << "\n";
    cout << "1. Cap nhat lai suat" << "\n";
    cout << "2. Tinh toan tien lai" << "\n";
    cout << "3. Rut tien" << "\n";
    cout << "4. Tim kiem va liet ke" << "\n";
    cout << "5. Sap xep giam dan/tang dan" << "\n";

    cin >> option;
    while(1){
        if(option == 0){
            cout << "Cam on ban" << "\n";
            break;
        }
        if(option == 1) capNhatLaiSuat(a, n);
        if(option == 2) tinhToanTienLai(a, n);
        if(option == 3) rutTien(a, n);
        if(option == 4){
            cout << "Nhap 0 neu ban muon ket thuc chuong
trinh" << "\n";
            cout << "Nhap 1 neu ban muon tim kiem" << "\n";
            cout << "Nhap 2 neu ban muon liet ke" << "\n";
            cout << "Lua chon cua ban la: "; cin >> option;
            while(option != 0){
                if(option == 0) break;
                if(option == 1) timKiem(a, n);
                if(option == 2) lietKe(a, n);
                cout << "Ban muon tiep tục tìm kiếm hay liet
ke khong: "; cin >> option;
            }
        }
        if(option == 5) sapXep(a, n);
        cout << "Ban muon thuc hien chuong trinh nao nua
khong: ";
        cin >> option;
    }
}

int main() {
    int n;
    soTietKiem a[105];

    Nhap(a, n);
    chuongTrinh(a, n);

    return 0;
}

```

Table 9 - Code chương trình quản lý sổ tiết kiệm

IT002 – Lập trình hướng đối tượng

Nội dung kiểm thử cho chương trình quản lý sổ tiết kiệm:

- Kiểm thử 1:

```
So luong so tiet kiem: 1
Ma so: ABCD
Loai tiet kiem: dai han
Ho ten KH: Doan Nguyen Minh Thu
Chung minh nhan dan: 092304557728
Ngay mo so (theo dinh dang ddmmyyyy): 04092024
So tien gui: 1000000
Moi ban chon chuong trinh:
1. Cap nhat lai suat
2. Tinh toan tien lai
3. Rut tien
4. Tim kiem va liet ke
5. Sap xep giam dan/tang dan
1
Nhap ma so tai khoan: ABCD
Nhap lai suat ban muon cap nhat: 5
Lai suat duoc cap nhat thanh cong cho tai khoan: ABCD
Ban muon thuc hien chuong trinh nao nua khong: 2
Nhap ma so tai khoan: ABCD
So tien lai cua tai khoan ABCD hien tai la: 1.03e+07
Ban muon thuc hien chuong trinh nao nua khong: 3
Nhap ma so tai khoan: ABCD
So tien ban muon rut la: 500000
Rut tien thanh cong. Cam on quy khach
Ban muon thuc hien chuong trinh nao nua khong: 4
Nhap 0 neu ban muon ket thuc chuong trinh
Nhap 1 neu ban muon tim kiem
Nhap 2 neu ban muon liet ke
Lua chon cua ban la: 1
Ban muon tim kiem theo CMND hay ma so
Nhan 1 de tim kiem theo CMND
Nhan 2 de tim kiem theo ma so
Su lua chon cua ban la: 1
Nhap CMND ban muon tim kiem: 092304557728
So ban tim kiem la: ABCD dai han Doan Nguyen Minh Thu 092304557728 4092024 500000
Ban muon tiep tục tim kiem hay liet ke khong: 2
Ban muon tim so mo trong thoi gian nao (nhap theo dinh dang ddmmyyyy): 04092024
So theo ban tim kiem la: ABCD dai han Doan Nguyen Minh Thu 092304557728 4092024 500000
Ban muon tiep tục tim kiem hay liet ke khong: 0
Ban muon thuc hien chuong trinh nao nua khong: 5
Nhap 1 neu ban muon sap xep theo so tien gui giam dan
Nhap 2 neu ban muon sap xep theo ngay mo so tang dan
1
Danh sach sau khi sap xep la:
ABCD dai han Doan Nguyen Minh Thu 092304557728 4092024 500000
Ban muon thuc hien chuong trinh nao nua khong: 2
```