

**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: Trần Phước Thanh Nhân - 24521241

### NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

This image shows a full page of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page, providing a template for handwriting practice or general writing. There are no margins, text, or other markings on the page.

....., ngày.....tháng.....năm 2025

**Người nhận xét**

(Ký tên và ghi rõ họ tên)

## MỤC LỤC

<b>Bài tập 1:</b> Viết chương trình nhập vào một phân số, rút gọn phân số và xuất kết quả ..7	
1.1 Sơ đồ khối (Flow Chart).....	7
1.2 Đoạn mã chương trình.....	8
1.3 Kiểm thử chương trình .....	9
<b>Bài tập 2:</b> Viết chương trình nhập vào hai phân số, tìm phân số lớn nhất và xuất kết quả .....	10
2.1 Sơ đồ khối (Flow Chart).....	10
2.2 Đoạn mã chương trình.....	10
2.3 Kiểm thử chương trình .....	10
<b>Bài tập 3:</b> Viết chương trình nhập vào hai phân số. Tính tổng, hiệu, tích, thương giữa chúng và xuất kết quả. ....	11
3.1 Sơ đồ khối (Flow Chart).....	11
3.2 Đoạn mã chương trình.....	11
3.3 Kiểm thử chương trình .....	13
<b>Bài tập 4:</b> Lập chương trình tính $\sin(x)$ với độ chính xác 0.00001 theo công thức.....	13
4.1 Sơ đồ khối (Flow Chart).....	13
4.2 Đoạn mã chương trình.....	14
4.3 Kiểm thử chương trình .....	15
<b>Bài tập 5:</b> 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.....	16
5.1 Sơ đồ khối (Flow Chart).....	16
5.2 Đoạn mã chương trình.....	16
5.3 Kiểm thử chương trình .....	17
<b>Bài tập 6:</b> 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. ....	18
6.1 Sơ đồ khối (Flow Chart).....	18
6.2 Đoạn mã chương trình.....	19
6.3 Kiểm thử chương trình .....	19
<b>Bài tập 7:</b> Viết chương trình quản lý thông tin của nhiều học sinh, bao gồm họ tên, điểm toán, điểm văn, và thêm điểm ngoại ngữ.....	20
7.1 Sơ đồ khối (Flow Chart).....	20
7.2 Đoạn mã chương trình.....	21
7.3 Kiểm thử chương trình .....	21

<b>Bài tập 8:</b> Xây dựng một chương trình quản lý thông tin chuyến bay .....	22
8.1 Sơ đồ khối (Flow Chart).....	22
8.2 Đoạn mã chương trình.....	23
8.3 Kiểm thử chương trình .....	24
<b>Bài tập 9:</b> Xây dựng chương trình quản lý sổ tiết kiệm với các yêu cầu mở rộng.....	25
9.1 Sơ đồ khối (Flow Chart).....	25
9.2 Đoạn mã chương trình.....	26
9.3 Kiểm thử chương trình .....	27

## DANH MỤC BẢNG

Bảng 1: Mô tả Testcase bài tập 1.....	9
Bảng 2: Mô tả Testcase bài tập 2.....	11
Bảng 3: Mô tả Testcase bài tập 3.....	13
Bảng 4: Mô tả Testcase bài tập 4.....	15
Bảng 5: Mô tả Testcase bài tập 5.....	18
Bảng 6: Mô tả Testcase bài tập 6.....	20
Bảng 7: Mô tả Testcase bài tập 7.....	22
Bảng 8: Mô tả Testcase bài tập 8.....	24
Bảng 9: Mô tả Testcase bài tập 9.....	28

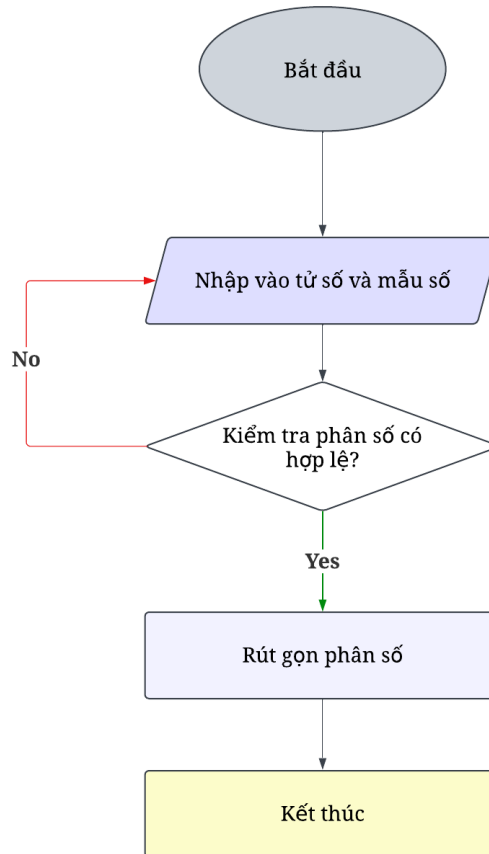
## DANH MỤC HÌNH ẢNH

Hình 1: Sơ đồ khối bài tập 1 .....	7
Hình 2: Đoạn mã chương trình bài tập 1 .....	8
Hình 3: Sơ đồ khối bài tập 2 .....	10
Hình 4: Đoạn mã chương trình bài tập 2 .....	10
Hình 5: Đoạn mã chương trình bài tập 3 .....	12
Hình 6: Sơ đồ khối bài tập 4 .....	14
Hình 7: Đoạn mã chương trình bài tập 4 .....	14
Hình 8: Sơ đồ khối bài tập 5 .....	16
Hình 9: Đoạn mã chương trình bài tập 5 .....	17
Hình 10: Sơ đồ khối bài tập 6 .....	18
Hình 11: Đoạn mã chương trình bài tập 6 .....	19
Hình 12: Sơ đồ khối bài tập 7 .....	20
Hình 13: Đoạn mã chương trình bài tập 7 .....	21
Hình 14: Sơ đồ khối bài tập 8 .....	22
Hình 15: Đoạn mã chương trình bài tập 8 .....	23
Hình 16: Sơ đồ khối bài tập 9 .....	25
Hình 17: Đoạn mã chương trình bài tập 9 .....	26
Hình 18: Đoạn mã các hàm kiểm tra chương trình bài tập 9 .....	27

## NỘI DUNG BÀI LÀM

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

### 1.1 Sơ đồ khối (Flow Chart)



Hình 1: Sơ đồ khối bài tập 1

## 1.2 Đoạn mã chương trình

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  struct PhanSo{
6      int ts, ms;
7      PhanSo() : ts(0), ms(1) {}
8  };
9
10 void input(PhanSo &A){
11     do{
12         cout << "Hay nhap vao tu so: "; cin >> A.ts;
13         cout << "Hay nhap vao mau so: "; cin >> A.ms;
14         if (A.ms == 0) cout << "Mau so khong duoc bang 0, vui long nhap lai!\n";
15     }while(A.ms == 0);
16 }
17
18 int getUCLN(int a, int b){
19     return a == 0 ? b : getUCLN(b % a, a);
20 }
21
22 void rutGon(PhanSo &A){
23     int UCLN = getUCLN(A.ts, A.ms);
24     A.ts /= UCLN, A.ms /= UCLN;
25 }
26
27 void output(const PhanSo &A){
28     cout << "Phan so sau khi duoc rut gon: " << A.ts << "/" << A.ms;
29 }
30
31 int main(){
32     PhanSo A;
33     input(A);
34     rutGon(A);
35     output(A);
36
37     return 0;
38 }

```

Hình 2: Đoạn mã chương trình bài tập 1

### Mô tả đầu vào (Input) và đầu ra (Output) của thuật toán:

- Đầu vào (Input): Nhập vào hai số nguyên a,b lần lượt là tử số và mẫu số của phân số cần rút gọn. ( $-10^9 \leq a, b \leq 10^9$  và  $b \neq 0$ )
- Đầu ra (Output): Hai số nguyên a,b lần lượt là tử số và mẫu số của phân số sau khi đã rút gọn.

### Hướng giải quyết:

- Hướng giải quyết: Tìm ước chung lớn nhất của tử số và mẫu số, chia tử số và mẫu số đồng thời cho ước chung lớn nhất.

### Mô tả chương trình:

- Đối tượng **struct PhanSo**: Định nghĩa cấu trúc **PhanSo** lưu trữ hai biến có kiểu dữ liệu int **ts** là tử số và **ms** là mẫu số. Đối tượng **PhanSo** khi được tạo sẽ tự động có giá trị **ts** bằng 0 và **ms** bằng 1.
- Hàm **void input(PhanSo &A)**: Hàm **input** có nhiệm vụ nhận giá trị nhập vào cho biến tham chiếu **Phanso &A**. Bên trong hàm sử dụng vòng lặp do-while



## IT002 – Lập trình hướng đối tượng

nhằm yêu cầu người dùng nhập giá trị đến khi giá trị nhập là hợp lệ. Hàm đảm bảo giá trị của mẫu số (**A.ms**) sau khi nhập sẽ khác 0.

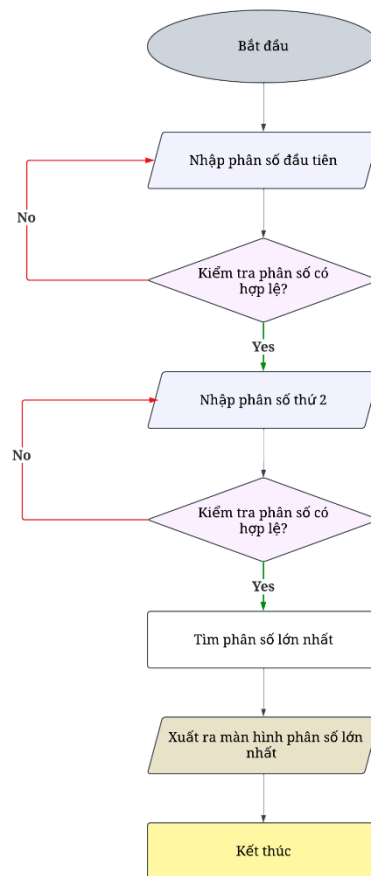
- Hàm **int getUCLN(int a, int b)**: Hàm **getUCLN** có nhiệm vụ trả về ước chung lớn nhất của hai số a và b bằng thuật toán Euclide.
- Hàm **output(const PhanSo &A)**: Hàm **output** có nhiệm vụ xuất kết quả là hai số nguyên a và b sau khi được rút gọn ra màn hình.

### 1.3 Kiểm thử chương trình

Chương trình được kiểm thử với các bộ giá trị sau.

Mô tả trường hợp	Giá trị tử số	Giá trị mẫu số	Kết quả
Người dùng nhập giá trị không hợp lệ	5	0	<pre>Hay nhap vao tu so: 5 Hay nhap vao mau so: 0 Mau so khong duoc bang 0, vui long nhap lai! Hay nhap vao tu so: </pre>
Người dùng nhập giá trị hợp lệ	5	10	<pre>● Hay nhap vao tu so: 5 Hay nhap vao mau so: 10 Phan so sau khi duoc rut gon: 1/2</pre>
	-8	-64	<pre>Hay nhap vao tu so: -8 Hay nhap vao mau so: -64 Phan so sau khi duoc rut gon: 1/8</pre>

Bảng 1: Mô tả Testcase bài tập 1

**Bài tập 2:** Viết chương trình nhập vào hai phân số, tìm phân số lớn nhất và xuất kết quả**2.1 Sơ đồ khối (Flow Chart)**

Hình 3: Sơ đồ khối bài tập 2

**2.2 Đoạn mã chương trình****2.3 Kiểm thử chương trình**

Chương trình được kiểm thử với các bộ giá trị sau.

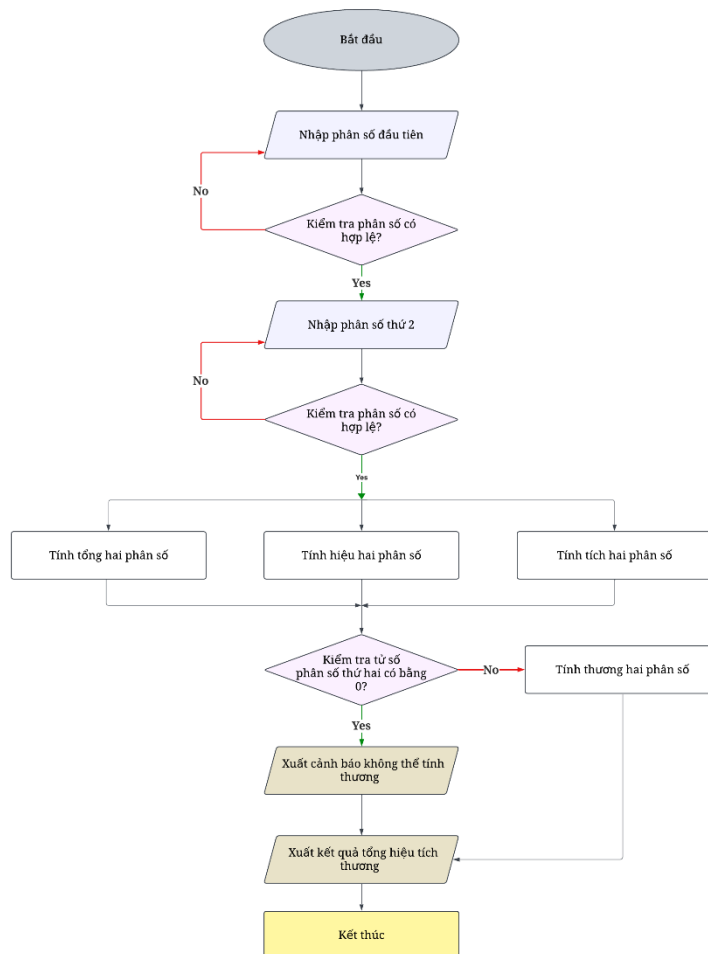
Mô tả trường hợp	Giá trị phân số đầu tiên	Giá trị phân số thứ hai	Kết quả
Người dùng nhập giá trị không hợp lệ	2/0	5/4	Hay nhập vào tu so thu nhat: 2 Hay nhập vào mau so thu nhat: 0 Mau so khong duoc bang 0, vui long nhập lại! Hay nhập vào tu so thu nhat:
	2/3	6/0	Hay nhập vào tu so thu nhat: 2 Hay nhập vào mau so thu nhat: 3 Hay nhập vào tu so thu hai: 6 Hay nhập vào mau so thu hai: 0 Mau so khong duoc bang 0, vui long nhập lại! Hay nhập vào tu so thu hai:
Người dùng nhập giá trị hợp lệ	2/3	9/5	Hay nhập vào tu so thu nhat: 2 Hay nhập vào mau so thu nhat: 3 Hay nhập vào tu so thu hai: 9 Hay nhập vào mau so thu hai: 5 Phan so lon nhat trong hai phân số là: 9/5

	5/4	-9/5	<pre> Hay nhap vao tu so thu nhat: 5 Hay nhap vao mau so thu nhat: 4 Hay nhap vao tu so thu hai: -9 Hay nhap vao mau so thu hai: 5 Phan so lon nhat trong hai phan so la: 5/4 </pre>
--	-----	------	--

Bảng 2: Mô tả Testcase bài tập 2

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

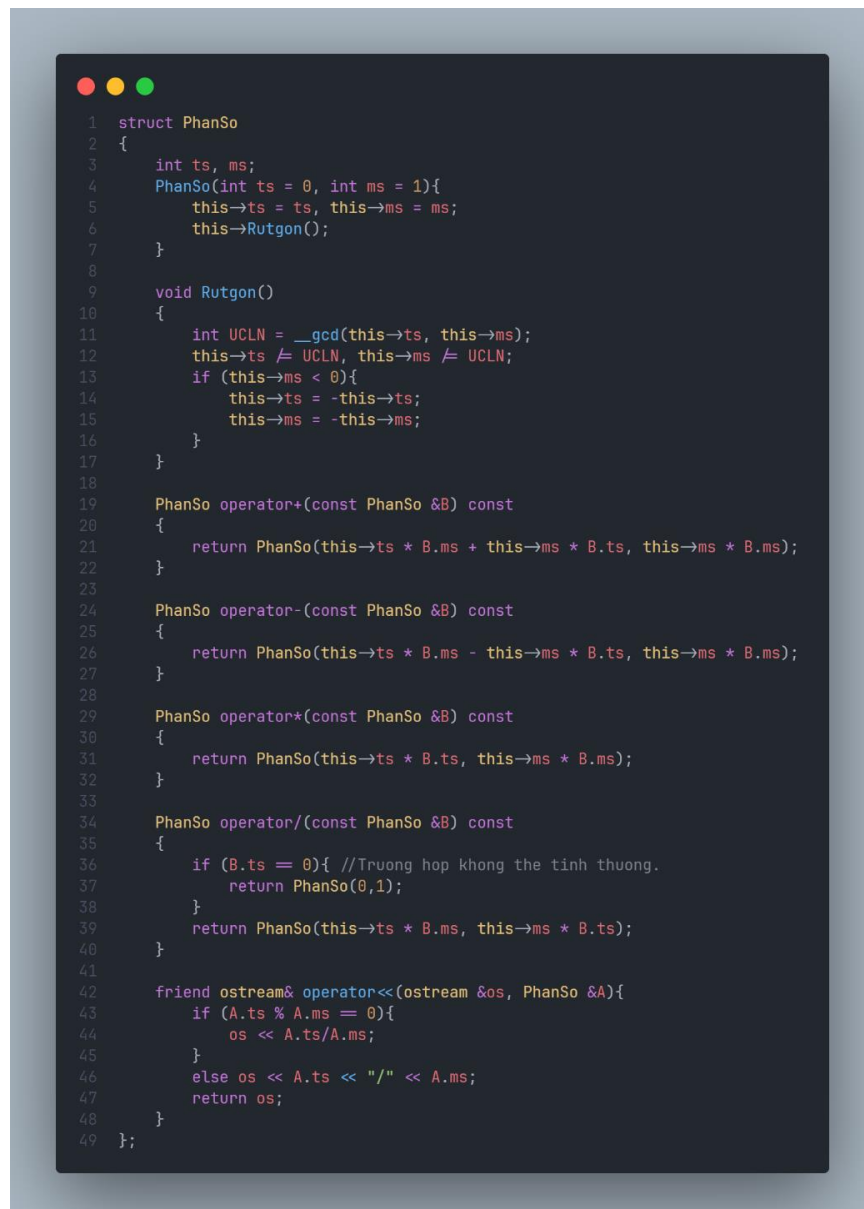
### 3.1 Sơ đồ khối (Flow Chart)



Hình 3.1: Sơ đồ khối bài tập 3

### 3.2 Đoạn mã chương trình

Chương trình sử dụng cấu trúc **struct PhanSo** định nghĩa cấu trúc phân số và các phép toán cộng, trừ, nhân và chia hai phân số.



```

1  struct PhanSo
2  {
3      int ts, ms;
4      PhanSo(int ts = 0, int ms = 1){
5          this->ts = ts, this->ms = ms;
6          this->Rutgon();
7      }
8
9      void Rutgon()
10     {
11         int UCLN = __gcd(this->ts, this->ms);
12         this->ts /= UCLN, this->ms /= UCLN;
13         if (this->ms < 0){
14             this->ts = -this->ts;
15             this->ms = -this->ms;
16         }
17     }
18
19     PhanSo operator+(const PhanSo &B) const
20     {
21         return PhanSo(this->ts * B.ms + this->ms * B.ts, this->ms * B.ms);
22     }
23
24     PhanSo operator-(const PhanSo &B) const
25     {
26         return PhanSo(this->ts * B.ms - this->ms * B.ts, this->ms * B.ms);
27     }
28
29     PhanSo operator*(const PhanSo &B) const
30     {
31         return PhanSo(this->ts * B.ts, this->ms * B.ms);
32     }
33
34     PhanSo operator/(const PhanSo &B) const
35     {
36         if (B.ts == 0){ //Trường hợp không thể tính thương.
37             return PhanSo(0,1);
38         }
39         return PhanSo(this->ts * B.ms, this->ms * B.ts);
40     }
41
42     friend ostream& operator<<(ostream &os, PhanSo &A){
43         if (A.ts % A.ms == 0){
44             os << A.ts/A.ms;
45         }
46         else os << A.ts << "/" << A.ms;
47         return os;
48     }
49 };

```

Hình 5: Đoạn mã chương trình bài tập 3

**Mô tả đầu vào (Input) và đầu ra (Output) của thuật toán:**

- Đầu vào (Input): Bốn số nguyên  $a, b, c, d$  lần lượt là tử số và mẫu số của phân số thứ nhất và phân số thứ hai. ( $-10^9 \leq a, b, c, d \leq 10^9$  và  $b * d \neq 0$ )
- Đầu ra (Output): Bốn phân số **tong, hieu, tich, thuong** lần lượt là kết quả tổng, hiệu, tích, thương của hai phân số đầu vào.

**Hướng giải quyết:**

- Thực hiện quy đồng mẫu số và cộng tử số sau quy đồng, rút gọn phân số nhận được để nhận phân số tối giản là kết quả phép toán tổng, hiệu, tích, thương của hai phân số.

**Mô tả chương trình:**

Trong cấu trúc **PhanSo** được định nghĩa phép khởi tạo hai giá trị **ts** và **ms** với giá trị mặc định là 0 và 1. Cấu trúc đồng thời định nghĩa hàm **Rutgon()** có nhiệm vụ rút gọn phân số bằng cách chia tử số và mẫu số cho ước chung lớn nhất (được xác định bằng hàm **\_\_gcd** từ thư viện **algorithm**). Cấu trúc đồng thời định nghĩa các phép toán cộng,

## IT002 – Lập trình hướng đối tượng

trừ, nhân và chia cho hai phân số; chồng lặp (tái định nghĩa) phép xuất giá trị << cho phép xuất dưới dạng phân số.

### 3.3 Kiểm thử chương trình

Chương trình được kiểm thử với các bộ giá trị sau.

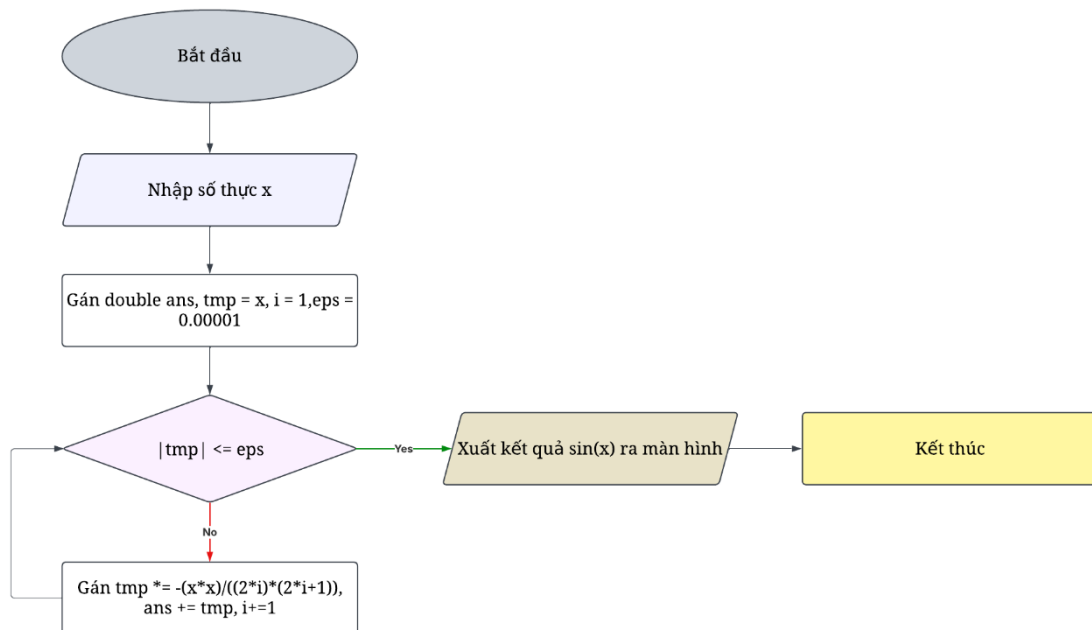
Mô tả trường hợp	Giá trị phân số đầu tiên	Giá trị phân số thứ hai	Kết quả
Người dùng nhập giá trị không hợp lệ	5/6	7/0	Hay nhap vao tu so thu nhat: 5 Hay nhap vao mau so thu nhat: 6 Hay nhap vao tu so thu hai: 7 Hay nhap vao mau so thu hai: 0 Mau so khong duoc bang 0, vui long nhap lai! Hay nhap vao tu so thu hai: █
Người dùng nhập giá trị hợp lệ	5/2	4/3	Hay nhap vao tu so thu nhat: 5 Hay nhap vao mau so thu nhat: 2 Hay nhap vao tu so thu hai: 4 Hay nhap vao mau so thu hai: 3 Tong hai phan so la: 23/6 Hieu hai phan so la: 7/6 Tich hai phan so la: 10/3 Thuong hai phan so la: 15/8
	1/2	0/5	Hay nhap vao tu so thu nhat: 1 Hay nhap vao mau so thu nhat: 2 Hay nhap vao tu so thu hai: 0 Hay nhap vao mau so thu hai: 5 Tong hai phan so la: 1/2 Hieu hai phan so la: 1/2 Tich hai phan so la: 0 Khong the tinh thuong!!
	-2/3	5/8	Hay nhap vao tu so thu nhat: -2 Hay nhap vao mau so thu nhat: 3 Hay nhap vao tu so thu hai: 5 Hay nhap vao mau so thu hai: 8 Tong hai phan so la: -1/24 Hieu hai phan so la: -31/24 Tich hai phan so la: -5/12 Thuong hai phan so la: -16/15

Bảng 3: Mô tả Testcase bài tập 3

**Bài tập 4:** Lập chương trình tính Sin(x) với độ chính xác 0.00001 theo công thức

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

#### 4.1 Sơ đồ khối (Flow Chart)



Hình 6: Sơ đồ khối bài tập 4

## 4.2 Đoạn mã chương trình

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  const double eps = 0.00001;
6
7  double cal_sin(double x){
8      x = fmod(x, 2 * 3.14);
9      double ans = x, tmp = x;
10     int i = 1;
11     while (fabs(tmp) > eps){
12         tmp *= -(x*x)/((2*i)*(2*i+1));
13         ans += tmp;
14         i++;
15     }
16     return ans;
17 }
18
19 int main(){
20     double x;
21     cout << "Hay nhap gia tri x: "; cin >> x;
22     cout << "Gia tri cua sin(x) la: " << cal_sin(x);
23
24     return 0;
25 }
    
```

Hình 7: Đoạn mã chương trình bài tập 4

**Mô tả đầu vào (Input) và đầu ra (Output) của thuật toán:**

- Đầu vào (Input): Một số thực  $x$  cần tính giá trị sin. ( $x \in \mathbb{R}$ )
- Đầu ra (Output): Một số thực là giá trị  $\sin(x)$  cần tìm.

**Hướng giải quyết:**

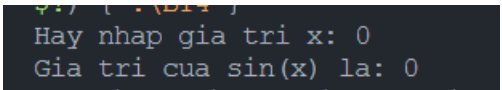
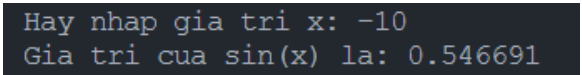
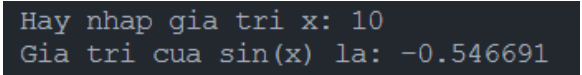
- Sử dụng vòng lặp để tính  $\sin(x)$  dựa trên khai triển Taylor. Quá trình tính sẽ kết thúc khi số hạng thứ  $i$  nhỏ hơn **0.00001** (Epsilon).

**Mô tả chương trình:**

- Giới hạn  $x$  vào khoảng  $[-2\pi, 2\pi]$  để giảm khối lượng tính toán.
- Lặp liên tục dựa theo khai triển Taylor đến khi số hạng thứ  $i$  trong khai triển nhỏ hơn giá trị **eps** (Epsilon).

**4.3 Kiểm thử chương trình**

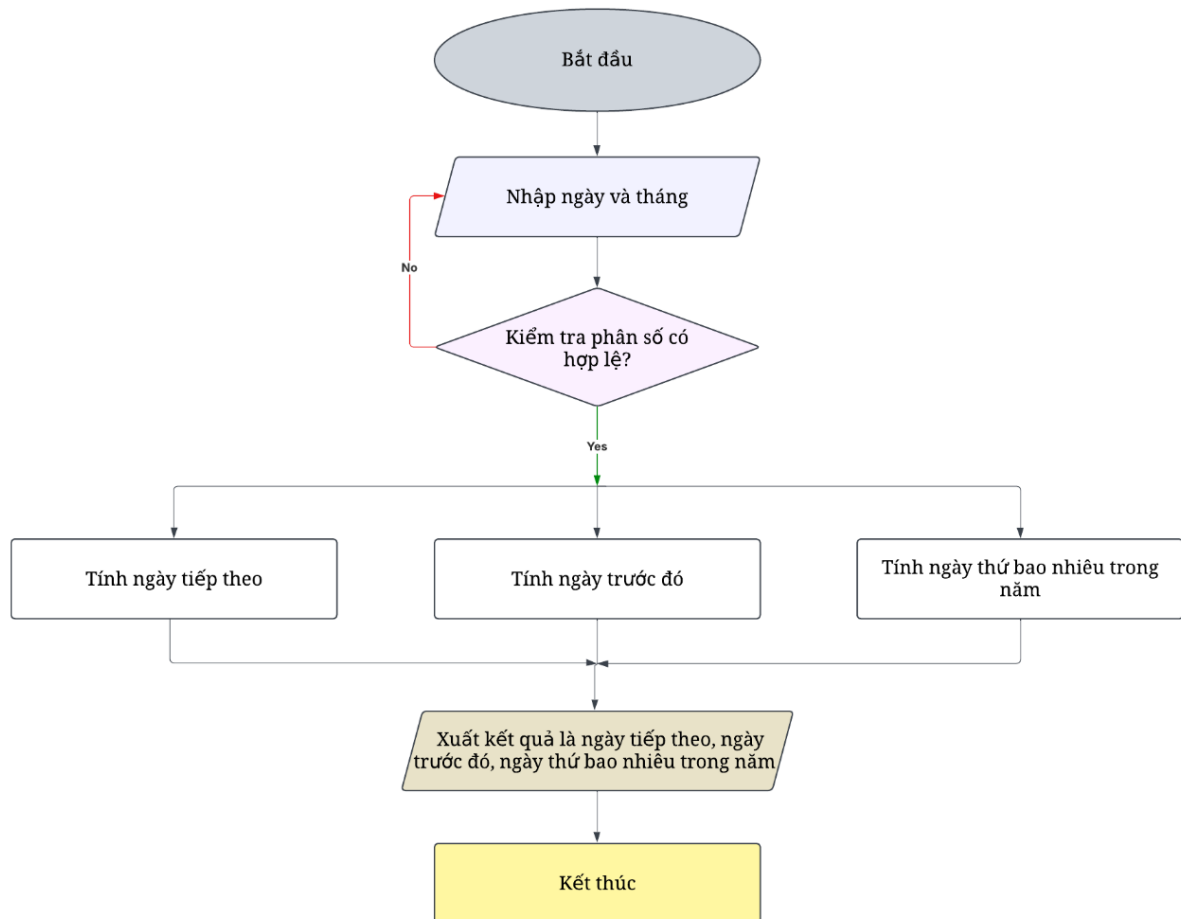
Chương trình được kiểm thử với các giá trị sau.

Mô tả trường hợp	Giá trị của x	Kết quả
Các giá trị x	0	
	-10	
	10	

Bảng 4: Mô tả Testcase bài tập 4

**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. Tính xem ngày đó là ngày thứ bao nhiêu trong năm và xuất kết quả.

### 5.1 Sơ đồ khối (Flow Chart)

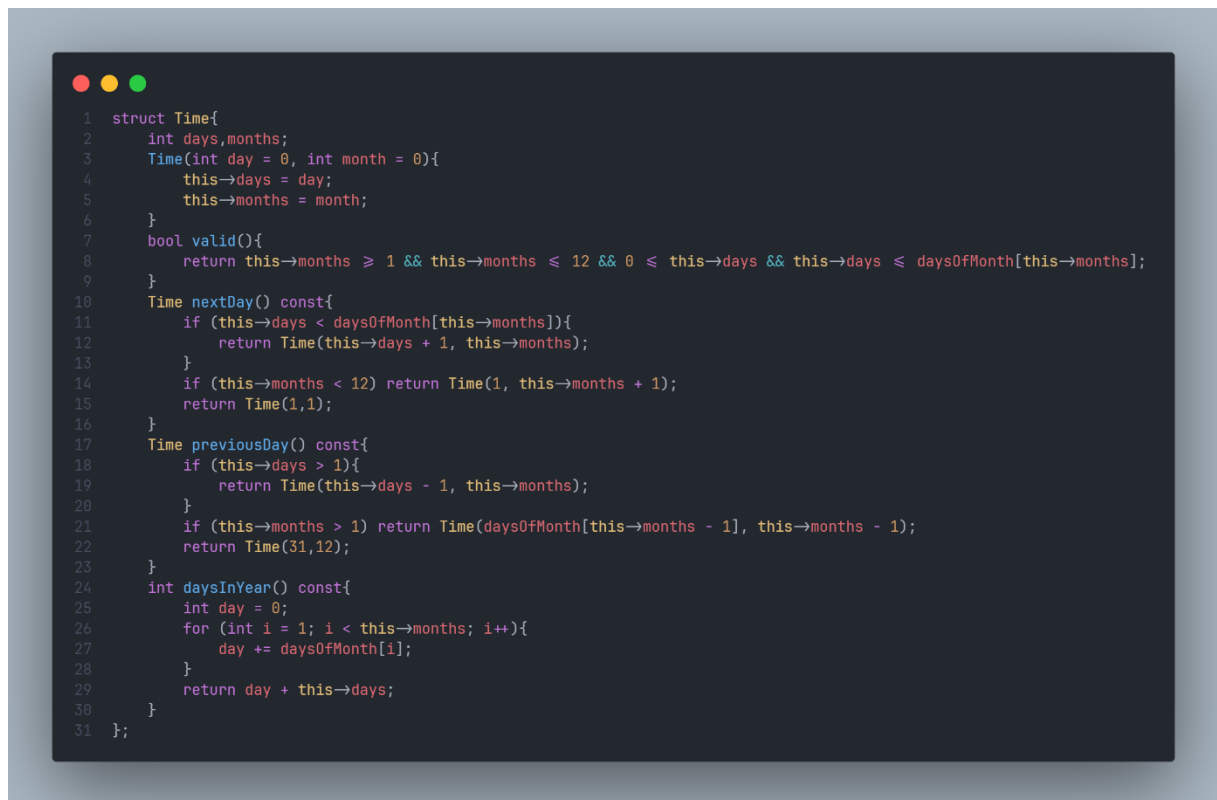


Hình 8: Sơ đồ khối bài tập 5

### 5.2 Đoạn mã chương trình

Chương trình khởi tạo cấu trúc **struct Time** chứa các thuật tính days, months lưu ngày và tháng từ input. Struct Time định nghĩa hàm kiểm tra hợp lệ, tính ngày tiếp theo, tính ngày trước đó và tính ngày thứ bao nhiêu trong năm.





Hình 9: Đoạn mã chương trình bài tập 5

### Mô tả đầu vào (Input) và đầu ra (Output) của thuật toán:

- Đầu vào (Input): Hai số nguyên dương a và b lần lượt là ngày tháng cần tính.
- Đầu ra (Output): Các số nguyên dương biểu thị ngày tiếp theo, ngày trước đó và ngày thứ bao nhiêu trong năm.

### Hướng giải quyết và mô tả chương trình:

- Tạo cấu trúc **Time** lưu trữ thông tin về ngày tháng và các hàm xác định ngày tháng hợp lệ, tính ngày tiếp theo, tính ngày trước đó và ngày thứ bao nhiêu trong năm.
- Hàm **Time nextDay()** xác định ngày tiếp theo bằng cách tăng ngày lên một đơn vị và cập nhật tháng (nếu có).
- Hàm **Time previousDay()** xác định ngày trước đó bằng cách giảm ngày đi một đơn vị và cập nhật tháng (nếu có).
- Hàm **int daysInYear()** tính ngày thứ bao nhiêu trong năm bằng cách cộng dồn các ngày của tháng trước đó và ngày trong tháng hiện tại.

### 5.3 Kiểm thử chương trình

Chương trình được kiểm thử với các giá trị sau.

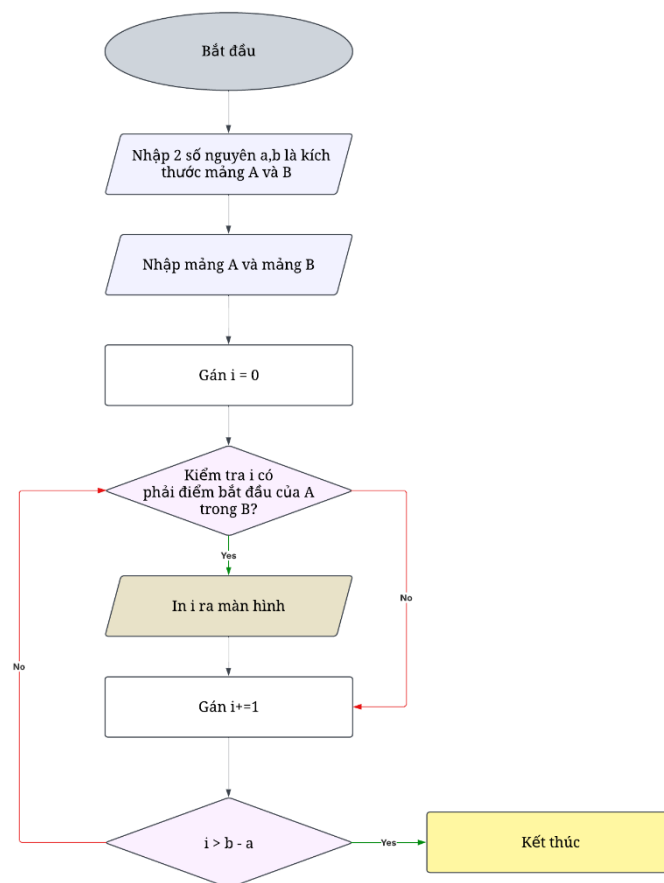
Mô tả trường hợp	Ngày và tháng	Kết quả
Ngày hoặc tháng không hợp lệ	5/13	Hay nhập vào ngày: 5 Hay nhập vào tháng: 13 Ngày và tháng không hợp lệ, vui lòng nhập lại! Hay nhập vào ngày:
	32/10	Hay nhập vào ngày: 32 Hay nhập vào tháng: 10 Ngày và tháng không hợp lệ, vui lòng nhập lại! Hay nhập vào ngày:

Ngày và tháng hợp lệ	31/10	Hay nhập vào ngày: 31 Hay nhập vào tháng: 10 Ngày tiếp theo là: 1/11 Ngày trước đó là: 30/10 Ngày 31/10 là ngày thu: 304
	31/12	Hay nhập vào ngày: 31 Hay nhập vào tháng: 12 Ngày tiếp theo là: 1/1 Ngày trước đó là: 30/12 Ngày 31/12 là ngày thu: 365

Bảng 5: Mô tả Testcase bài tập 5

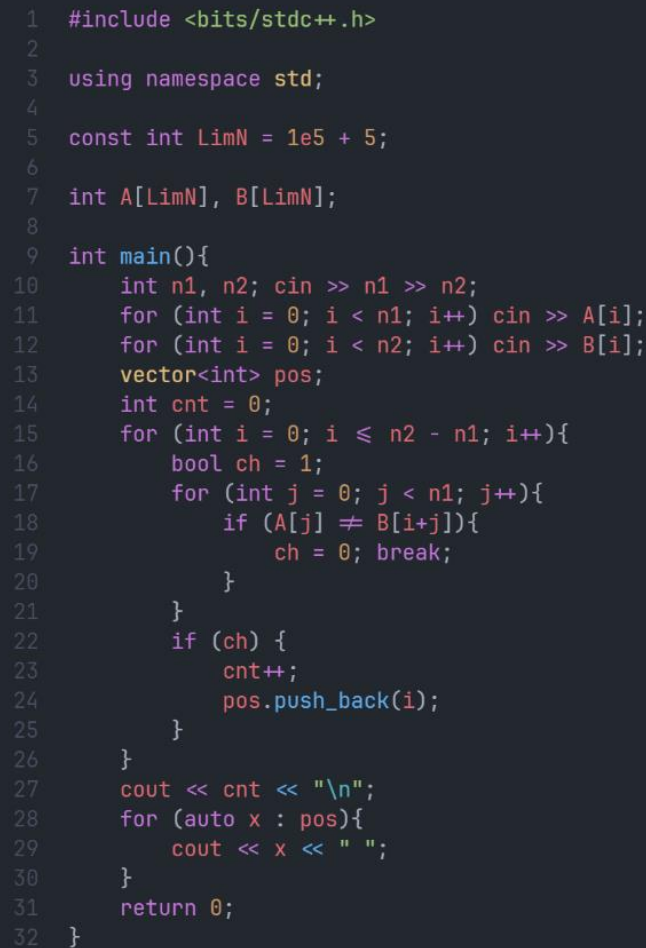
**Bài tập 6:** Cho hai mảng A và B chứa các số nguyên. Hãy xây dựng một chương trình để xác định xem có bao nhiêu lần mảng A xuất hiện liên tiếp dưới dạng một dãy con trong mảng B.

### 6.1 Sơ đồ khối (Flow Chart)



Hình 10: Sơ đồ khối bài tập 6

## 6.2 Đoạn mã chương trình



```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  const int LimN = 1e5 + 5;
6
7  int A[LimN], B[LimN];
8
9  int main(){
10     int n1, n2; cin >> n1 >> n2;
11     for (int i = 0; i < n1; i++) cin >> A[i];
12     for (int i = 0; i < n2; i++) cin >> B[i];
13     vector<int> pos;
14     int cnt = 0;
15     for (int i = 0; i ≤ n2 - n1; i++){
16         bool ch = 1;
17         for (int j = 0; j < n1; j++){
18             if (A[j] ≠ B[i+j]){
19                 ch = 0; break;
20             }
21         }
22         if (ch) {
23             cnt++;
24             pos.push_back(i);
25         }
26     }
27     cout << cnt << "\n";
28     for (auto x : pos){
29         cout << x << " ";
30     }
31     return 0;
32 }

```

Hình 11: Đoạn mã chương trình bài tập 6

### Mô tả đầu vào (Input) và đầu ra (Output) của thuật toán:

- Đầu vào (Input): Dòng đầu tiên là hai số nguyên dương  $a$  và  $b$  là kích thước mảng  $A$  và mảng  $B$ . Hai dòng tiếp theo lần lượt là các phần tử mảng  $A$  và  $B$ .
- Đầu ra (Output): Dòng đầu tiên là số lần mảng  $A$  xuất hiện trong mảng  $B$ . Dòng thứ hai là các số nguyên lần lượt là vị trí xuất hiện của mảng  $A$  trong mảng  $B$  theo thứ tự tăng dần.

### Hướng giải quyết và mô tả chương trình:

- Sử dụng kỹ thuật Sliding Window duyệt lần lượt qua các vị trí có thể xuất hiện của mảng  $A$  và kiểm tra vị trí đó có hợp lệ hay không.
- Độ phức tạp  $O(n * m)$  với  $n, m$  là số lượng phần tử của mảng  $A$  và mảng  $B$ .

## 6.3 Kiểm thử chương trình

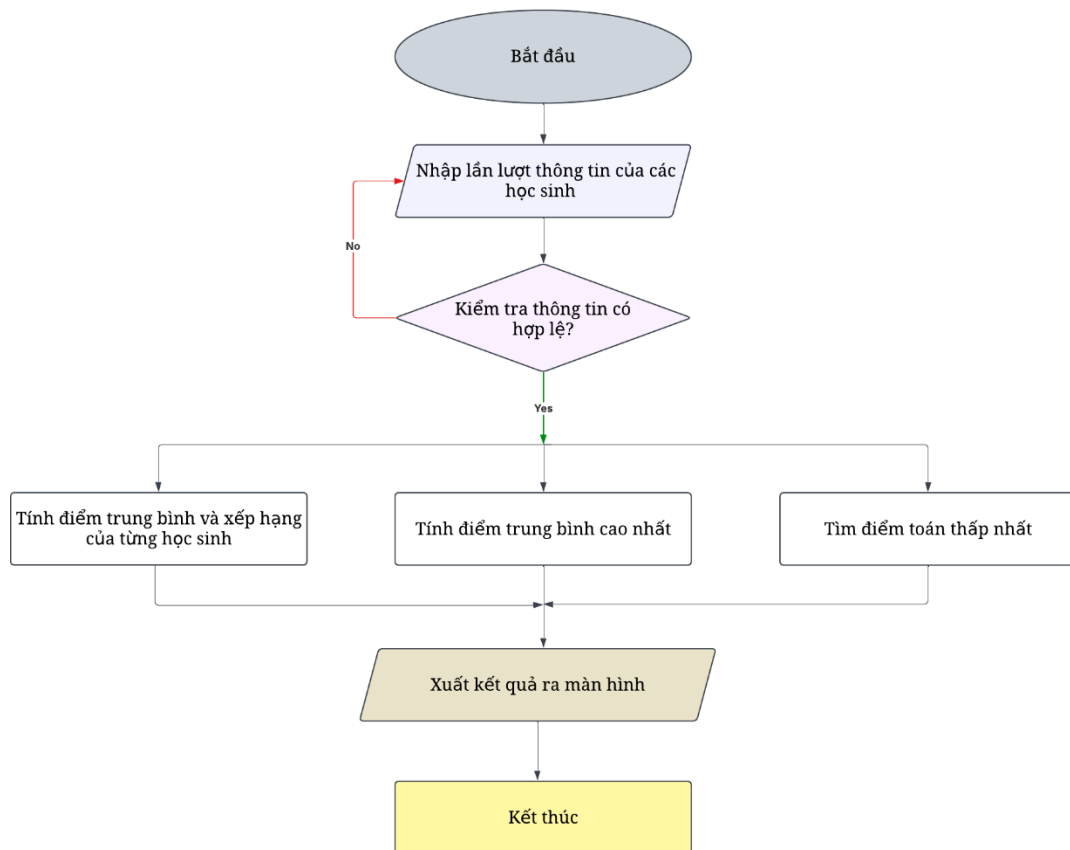
Chương trình được kiểm thử bằng các bộ giá trị sau:

Mảng A	Mảng B	Kết quả
[1, 2]	[1, 2, 1, 2, 3, 1, 2]	<pre> 2 7 1 2 1 2 1 2 3 1 2 Số phần xuất hiện là: 3 Các vị trí xuất hiện là: 0 2 5 </pre>
[5, 6, 8, 2]	[5, 6, 8, 2, 5, 6, 8, 1]	<pre> 4 8 5 6 8 2 5 6 8 2 5 6 8 1 Số phần xuất hiện là: 1 Các vị trí xuất hiện là: 0 </pre>

Bảng 6: Mô tả Testcase bài tập 6

**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.

### 7.1 Sơ đồ khối (Flow Chart)



Hình 12: Sơ đồ khối bài tập 7

## 7.2 Đoạn mã chương trình

```

1  class Student{
2  private:
3      double diemToan, diemVan, diemNgoaiNgu;
4      string hoVaTen;
5  public:
6      Student() : hoVaTen(""), diemToan(0), diemVan(0), diemNgoaiNgu(0) {}
7      bool valid() const{
8          return ch(diemNgoaiNgu) && ch(diemToan) && ch(diemVan);
9      }
10     double diemTrungBinh() const{
11         return ((2*diemToan) + diemVan + diemNgoaiNgu)/4;
12     }
13     double getToan() const{
14         return diemToan;
15     }
16     string phanLoai(){
17         double dtb = this->diemTrungBinh();
18         if (dtb ≥ 9) return "Xuat sac";
19         if (dtb ≥ 8) return "Gioi";
20         if (dtb ≥ 6.5) return "Kha";
21         if (dtb ≥ 5) return "Trung binh";
22         return "Yeu";
23     }
24     friend ostream& operator>>(ostream& is, Student& st){
25         return is >> st.diemToan >> st.diemVan >> st.diemNgoaiNgu;
26     }
27 };

```

Hình 13: Đoạn mã chương trình bài tập 7

### Mô tả đầu vào (Input) và đầu ra (Output) của thuật toán:

- Đầu vào (Input): Dòng đầu tiên là số nguyên dương  $n$  biểu thị số lượng học sinh.  $n$  dòng tiếp theo lần lượt là điểm toán, văn, ngoại của học sinh thứ  $i$ . ( $i \leq n$ )
- Đầu ra (Output):  $n$  dòng đầu tiên lần lượt là điểm trung bình và xếp loại của học sinh. Hai dòng tiếp theo lần lượt là điểm trung bình cao nhất và điểm toán thấp nhất.

### Hướng giải quyết và mô tả chương trình:

- Tạo một **class Student** lưu trữ thông tin điểm toán, văn, ngoại ngữ. Thông tin điểm được lưu trong thuộc tính **private**. **class Student** khai báo các hàm trong thuộc tính **public** gồm **bool valid()** để kiểm tra thông tin có hợp lệ không; hàm **double diemTrungBinh()** để tính điểm trung bình; hàm **double getToan()** để lấy điểm toán và hàm **string phanLoai()** để phân loại học sinh dựa trên điểm trung bình.
- Tạo mảng **Student st[n]** với  $n$  là số lượng học sinh để lưu thông tin các học sinh.

## 7.3 Kiểm thử chương trình

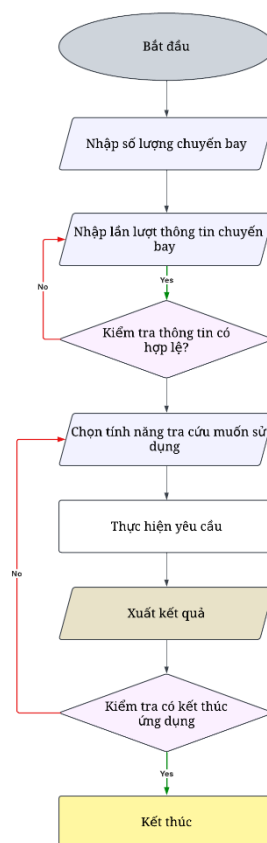
Chương trình được kiểm thử với các giá trị sau.

Mô tả trường hợp	Kết quả
Thông tin không hợp lệ	<pre> Số lượng học sinh là: 3 Hay nhập điểm toán, văn, ngoại ngữ của học sinh thu 1: 5.6 5.6 8.5 Hay nhập điểm toán, văn, ngoại ngữ của học sinh thu 2: 4.2 3.6 8.9 Hay nhập điểm toán, văn, ngoại ngữ của học sinh thu 3: 5.6 10.5 8.5 Không hợp lệ, hay nhập lại! Hay nhập điểm toán, văn, ngoại ngữ của học sinh thu 3: </pre>
Thông tin hợp lệ	<pre> Số lượng học sinh là: 2 Hay nhập điểm toán, văn, ngoại ngữ của học sinh thu 1: 8.5 8.6 8.7 Hay nhập điểm toán, văn, ngoại ngữ của học sinh thu 2: 7.9 8.0 8.1 ..... Điểm trung bình và xếp loại học sinh thu 1 là: 8.575 Giỏi Điểm trung bình và xếp loại học sinh thu 2 là: 7.975 Kha Điểm trung bình cao nhất là: 8.575 Điểm toán thấp nhất là: 7.9 </pre>

Bảng 7: Mô tả Testcase bài tập 7

## Bài tập 8: Xây dựng một chương trình quản lý thông tin chuyến bay

### 8.1 Sơ đồ khối (Flow Chart)



Hình 14: Sơ đồ khối bài tập 8

## 8.2 Đoạn mã chương trình

```

1  struct Flights{
2      string maChuyenBay, noiDen, noiDi;
3      pair<int,int> gioDen, gioDi, ngayDi;
4      bool isValid(){
5          if (maChuyenBay.size() > 5 || !ktraKitu(maChuyenBay)) return 0;
6          if (!ktraThoiGian(gioDen) || !ktraThoiGian(gioDi)) return 0;
7          if (noiDen.size() > 20 || !ktraKitu(noiDen)) return 0;
8          if (noiDi.size() > 20 || !ktraKitu(noiDi)) return 0;
9          if (!ktraNgay(ngayDi)) return 0;
10         return 1;
11     }
12     friend istream &operator>>(istream &is, Flights &FL){
13         do{
14             cout << "Hay nhap thong tin chuyen bay!\n";
15             cout << "Hay nhap ma chuyen bay: "; cin >> FL.maChuyenBay;
16             cout << "Hay noi di cua chuyen bay: "; cin >> FL.noiDi;
17             cout << "Hay noi den cua chuyen bay: "; cin >> FL.noiDen;
18             cout << "Hay gio di cua chuyen bay: "; cin >> FL.gioDi.first >> FL.gioDi.second;
19             cout << "Hay gio den cua chuyen bay: "; cin >> FL.gioDen.first >> FL.gioDen.second;
20             cout << "Hay ngay bay cua chuyen bay: "; cin >> FL.ngayDi.first >> FL.ngayDi.second;
21
22             if (!FL.isValid()){
23                 cout << "Thong tin khong hop le, hay nhap lai!\n";
24             }
25             cout << ".....\n";
26         }while(!FL.isValid());
27         return is;
28     }
29 };
30
31 void sapXep(Flights *FL, const int &n){
32     auto compare = [](const Flights &a, const Flights &b){
33         if (a.ngayDi.second == b.ngayDi.second) return a.ngayDi.second > b.ngayDi.second;
34         return a.ngayDi.first > a.ngayDi.first;
35     };
36     sort(FL, FL + n, compare);
37 }
38
39 void ktraMaChuyenBay(Flights *FL, const int &n, const string &name){
40     for (int i=0; i < n; i++){
41         if (name == FL[i].maChuyenBay){
42             cout << "Ton tai!\n"; return;
43         }
44     }
45     cout << "Khong ton tai!\n";
46 }

```

Hình 15: Đoạn mã chương trình bài tập 8

### Mô tả đầu vào (Input) và đầu ra (Output) của chương trình:

- Đầu vào (Input): Dòng đầu tiên chứa một số nguyên dương là số lượng chuyến bay. Các dòng tiếp theo lần lượt là thông tin của chuyến bay.
- Đầu ra (Output): Kết quả sau khi xử lý các yêu cầu từ người dùng.

### Hướng giải quyết và mô tả chương trình:

- Tạo **struct Flights** lưu trữ thông tin của một chuyến bay, bao gồm các thuộc tính **maChuyenBay**, **noiDen**, **noiDi**, **gioDen**, **gioDi**, **ngayDi** và hàm nhập dữ liệu. Hàm nhập dữ liệu đảm bảo dữ liệu chuẩn xác và yêu cầu nhập lại khi thông tin nhập chưa chuẩn xác.
- Hàm **void sapXep** có chức năng sắp xếp các chuyến bay theo thứ tự tăng dần ngày tháng. Hàm sử dụng phép so sánh **compare()** để so sánh hai chuyến bay với nhau và hàm **sort** từ thư viện **algorithm** để sắp xếp.
- Hàm **void ktraMaChuyenBay** có chức năng kiểm tra xem chuyến bay có tồn tại hay không và in kết quả ra màn hình.

### 8.3 Kiểm thử chương trình

Chương trình được kiểm thử với các giá trị sau.

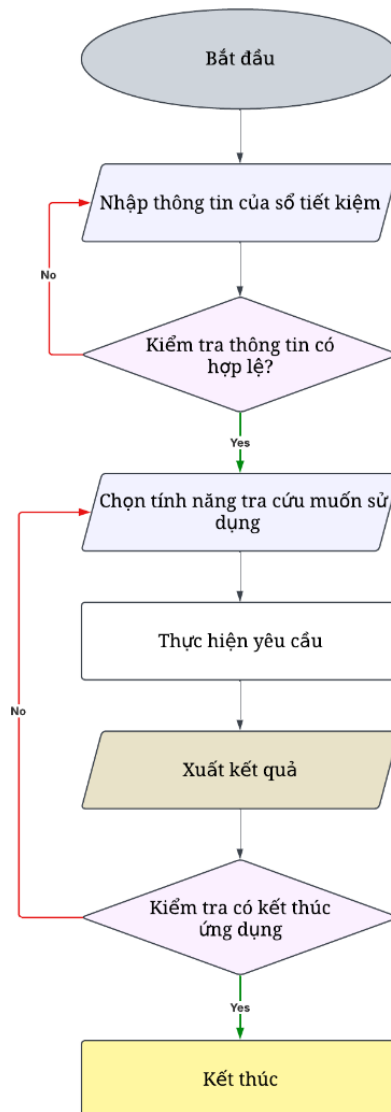
Mô tả trường hợp	Kết quả
Yêu cầu tìm kiếm mã chuyến bay có tồn tại không	<pre> Hay nhap so luong chuyen bay: 3 Hay nhap thong tin chuyen bay! Hay nhap ma chuyen bay: VN Hay noi di cua chuyen bay: DN Hay noi den cua chuyen bay: SGN Hay gio di cua chuyen bay: 12 30 Hay gio den cua chuyen bay: 15 00 Hay ngay bay cua chuyen bay: 12 2 ..... Hay nhap thong tin chuyen bay! Hay nhap ma chuyen bay: SN Hay noi di cua chuyen bay: DN Hay noi den cua chuyen bay: HAN Hay gio di cua chuyen bay: 12 20 Hay gio den cua chuyen bay: 15 20 Hay ngay bay cua chuyen bay: 13 2 ..... Hay nhap thong tin chuyen bay! Hay nhap ma chuyen bay: VJ Hay noi di cua chuyen bay: SGN Hay noi den cua chuyen bay: HAN Hay gio di cua chuyen bay: 15 20 Hay gio den cua chuyen bay: 17 00 Hay ngay bay cua chuyen bay: 18 2 ..... Hay chon yeu cau!2 1. Sap xep cac chuyen bay! 2. Kiem tra ma may bay! 3. Dem so chuyen bay tu dia diem nay den dia diem khac 4. Danh sach cac chuyen bay khoi hanh tu Ha Noi Hay nhap ma muonkiem tra: VJ Ton tai! ..... </pre>
Yêu cầu sắp xếp các chuyến bay	<pre> Hay nhap so luong chuyen bay: 3 Hay nhap thong tin chuyen bay! Hay nhap ma chuyen bay: VN Hay noi di cua chuyen bay: SGN Hay noi den cua chuyen bay: HAN Hay gio di cua chuyen bay: 12 50 Hay gio den cua chuyen bay: 15 20 Hay ngay bay cua chuyen bay: 12 3 ..... Hay nhap thong tin chuyen bay! Hay nhap ma chuyen bay: VJ Hay noi di cua chuyen bay: SGN Hay noi den cua chuyen bay: DAD Hay gio di cua chuyen bay: 12 50 Hay gio den cua chuyen bay: 15 00 Hay ngay bay cua chuyen bay: 13 3 ..... Hay nhap thong tin chuyen bay! Hay nhap ma chuyen bay: QH Hay noi di cua chuyen bay: DAD Hay noi den cua chuyen bay: NAH Hay gio di cua chuyen bay: 15 20 Hay gio den cua chuyen bay: 23 00 Hay ngay bay cua chuyen bay: 15 3 ..... Hay chon yeu cau!1 1. Sap xep cac chuyen bay! 2. Kiem tra ma may bay! 3. Dem so chuyen bay tu dia diem nay den dia diem khac 4. Danh sach cac chuyen bay khoi hanh tu Ha Noi 12/3: VN 13/3: VJ 15/3: QH </pre>

Bảng 8: Mô tả Testcase bài tập 8



**Bài tập 9:** Xây dựng chương trình quản lý sổ tiết kiệm với các yêu cầu mở rộng

**9.1 Sơ đồ khối (Flow Chart)**



Hình 16: Sơ đồ khối bài tập 9

## 9.2 Đoạn mã chương trình

```

1  class tietKiem{
2  private:
3      string loiTietKiem, hoVaTen, CMND;
4      double soTienGui;
5      double laiSuat;
6
7  public:
8      pair<int,int> ngayMo;
9      bool isValid(){
10         if (loiTietKiem.size() > 10 || !ktraKitu(loiTietKiem)) return 0;
11         if (hoVaTen.size() > 30 || !ktraKitu(hoVaTen)) return 0;
12         if (soTienGui ≤ 0) return 0;
13         if (!ktraKitu(CMND)) return 0;
14         if (!ktraNgay(ngayMo)) return 0;
15         return 1;
16     }
17     friend istream &operator>>(istream &is, tietKiem &TK){
18         do{
19             cout << "Hay nhap thong tin so tiet kiem!\n";
20             cout << "Hay nhap loi tiet kiem: "; cin >> TK.loiTietKiem;
21             cout << "Hay nhap ho va ten chu so huu: "; cin >> TK.hoVaTen;
22             cout << "Hay nhap CMND cua chu so huu: "; cin >> TK.CMND;
23             cout << "Hay nhap ngay mo: "; cin >> TK.ngayMo.first >> TK.ngayMo.second;
24             cout << "Hay nhap so tien gui: "; cin >> TK.soTienGui;
25
26             if (!TK.isValid()){
27                 cout << "Thong tin khong hop le, hay nhap lai!\n";
28             }
29             cout << ".....\n";
30             }while(!TK.isValid());
31             return is;
32         }
33     friend ostream &operator<<(ostream& os, tietKiem &TK){
34         cout << "Thong tin so tiet kiem: " << TK.loiTietKiem << '\n';
35         cout << "Thong tin ho va ten chu so huu: " << TK.hoVaTen << '\n';
36         cout << "Thong tin CMND cua chu so huu: " << TK.CMND << '\n';
37         cout << "Thong tin ngay mo: " << TK.ngayMo.first << "/" << TK.ngayMo.second << '\n';
38         cout << "Thong tin so tien gui: " << TK.soTienGui << '\n';
39         return os;
40     }
41     void capNhat(const double &DATA){
42         laiSuat = DATA;
43     }
44     double tienLai() const {
45         return laiSuat * soTienGui;
46     }
47 };
48
49 void sapXep(tietKiem *FL, const int &n){
50     auto compare = [](const tietKiem &a, const tietKiem &b){
51         if (a.ngayMo.second ≠ b.ngayMo.second) return a.ngayMo.second > b.ngayMo.second;
52         return a.ngayMo.first > a.ngayMo.first;
53     };
54     sort(FL, FL + n, compare);
55 }

```

Hình 17: Đoạn mã chương trình bài tập 9

### Mô tả đầu vào (Input) và đầu ra (Output) của chương trình:

- Đầu vào (Input): Dòng đầu tiên chứa một số nguyên dương là số lượng sổ tiết kiệm. Các dòng tiếp theo lần lượt là thông tin của sổ tiết kiệm tương ứng.
- Đầu ra (Output): Kết quả sau khi xử lý các yêu cầu từ người dùng.

### Hướng giải quyết và mô tả chương trình:

- Tạo **struct tietKiem** lưu trữ thông tin của một sổ tiết kiệm, bao gồm các thuộc tính **loiTietKiem**, **hoVaTen**, **CMND**, **soTienGui**, **laiSuat** và hàm nhập dữ liệu. Hàm nhập dữ liệu đảm bảo dữ liệu chuẩn xác và yêu cầu nhập lại khi thông tin nhập chưa chuẩn xác.

- Hàm **void sapXep** có chức năng sắp xếp các số tiết kiệm theo thứ tự tăng dần ngày tháng. Hàm sử dụng phép so sánh **compare()** để so sánh hai chuyến bay với nhau và hàm **sort** từ thư viện **algorithm** để sắp xếp.
- Hàm **bool ktraKitu** có chức năng kiểm tra chuỗi có hợp lệ hay không (chuỗi hợp lệ là chuỗi chỉ chứa kí tự chữ cái hoặc dấu cách).
- Hàm **bool ktraNgay** có chức năng kiểm tra ngày tháng có hợp lệ hay không.

```

1  bool ktraKitu(const string &x){
2      for (auto ch : x){
3          if (!('a' ≤ ch && ch ≤ 'z') || ('A' ≤ ch && ch ≤ 'Z') || ch == ' ')) return 0;
4      }
5      return 1;
6  }
7
8  bool ktraNgay(const pair<int,int> &x){
9      if (x.first < 0 || x.second < 0 || x.first > 31 || x.second > 12) return 0;
10     if (x.second == 2) return x.first ≤ 29;
11     if (x.second == 4 || x.second == 6 || x.second == 9 || x.second == 11) return (x.first ≤ 30);
12     return 1;
13 }

```

Hình 18: Đoạn mã các hàm kiểm tra chương trình bài tập 9

### 9.3 Kiểm thử chương trình

Chương trình được kiểm thử với các giá trị sau.

Mô tả trường hợp	Kết quả
Yêu cầu thay đổi lãi suất và tính tiền lãi	<pre> Hay nhập số lượng số tiết kiệm: 1 Hay nhập thông tin số tiết kiệm! Hay nhập loại tiết kiệm: nganHan Hay nhập họ và tên chủ số huu: Nhan Hay nhập CMND của chủ số huu: 123456 Hay nhập ngày m: 12 3 Hay nhập số tiền gửi: 1200 ..... Hay chọn yêu cầu! 2 1. Sắp xếp các số tiết kiệm! 2. Cập nhật lãi suất! 3. Tính toán tiền lãi Chọn số muốn cập nhật lãi suất: 0 Lãi suất mới: 0.06 ..... Hay chọn yêu cầu! 3 1. Sắp xếp các số tiết kiệm! 2. Cập nhật lãi suất! 3. Tính toán tiền lãi Chọn số muốn cập nhật lãi suất: 0 Số tiền lãi: 72 </pre>

<p>Yêu cầu sắp xếp các sổ tiết kiệm</p>	<pre> Hay nhap so luong so tietkiem: 2 Hay nhap thong tin so tietkiem! Hay nhap loai tietkiem: nganHan Hay nhap ho va ten chu so huu: Nhan Hay nhap CMND cua chu so huu: 123456 Hay nhap ngay mo: 12 3 Hay nhap so tien gui: 1200  ..... Hay nhap thong tin so tietkiem! Hay nhap loai tietkiem: daiHan Hay nhap ho va ten chu so huu: Nhan Hay nhap CMND cua chu so huu: 123456 Hay nhap ngay mo: 12 4 Hay nhap so tien gui: 1300  ..... Hay chon yeu cau! 1 1. Sap xep cac so tietkiem! 2. Cap nhat lai suat! 3. Tinh toan tien lai So thu 1: 12/3 So thu 2: 12/4 </pre>
---	--

Bảng 9: Mô tả Testcase bài tập 9

## Phụ lục

**Lớp:** Lập trình hướng đối tượng (IT002.P212)

**Sinh viên thực hiện:**

- Họ và tên: Trần Phước Thanh Nhân
- MSSV: 24521241

**Đường dẫn Github:** [Github Link](#)

## BẢN QUYỀN

**Tác giả:** Trần Phước Thanh Nhân

**Nơi lưu trữ:** [github.com/normalman159/IT002-OOP-UIT](https://github.com/normalman159/IT002-OOP-UIT)

Tài liệu được thực hiện và sở hữu bản quyền bởi Trần Phước Thanh Nhân (UIT-VNUHCM).

Tài liệu được sử dụng với mục đích tham khảo hoặc học tập môn học Lập trình hướng đối tượng (OOP) trong và ngoài trường Đại học Công nghệ thông tin (UIT). Khi chia sẻ tài liệu này cần ghi rõ đường dẫn đến Github và thông tin tác giả.

Các hành vi sao chép, sử dụng ngoài mục đích tham khảo và học tập như mục đích thương mại,... dưới bất kì hình thức nào mà không thông báo đến tác giả đều không được phép.

### Thông tin liên hệ

- Email: [nhantt.work@gmail.com](mailto:nhantt.work@gmail.com)
- Github Profile: <https://github.com/normalman159>

Chúc các bạn học tốt môn học này.