

**TÀI LIỆU  
DƯỠNG HỌC SINH GIỎI**

**C++**

## GIỚI THIỆU

Trong những năm qua, ngành giáo dục và đào tạo đã tổ chức và triển khai nhiều cuộc thi học sinh giỏi môn Tin học. Về cơ bản, đa số các thầy / cô giáo đang hướng dẫn sử dụng ngôn ngữ lập trình Pascal – Đây là ngôn ngữ đã quá cổ điển và không còn nhận được sự hỗ trợ trên các hệ điều hành mới, đặc biệt là với những màn hình có độ phân giải cao, việc sử dụng công cụ này ngày càng trở lên khó khăn hơn rất nhiều.

Tài liệu được xây dựng dựa trên cơ sở các nguồn dữ liệu do các đồng nghiệp cung cấp và một số dữ liệu từ Internet cũng như các kiến thức tác giả đã tìm hiểu, nghiên cứu và phát triển trong quá trình giảng dạy.

Hiện tại, công tác thi học sinh giỏi ở Việt Nam đang sử dụng phần mềm Themis để chấm điểm với các ngôn ngữ lập trình được sử dụng là Pascal, C++, Python và Java. Các nội dung INPUT và OUPUT được ghi lại vào tệp \*.INP và \*.OUT, vì vậy tất cả các bài tập thực hành chúng tôi trình bày theo kiểu đề thi để các em học sinh có thể dễ dàng tiếp cận và tránh được những sai sót trong quá trình làm bài.

C++ là ngôn ngữ lập trình hướng đối tượng. Tuy nhiên, tác giả không đề cập đến vấn đề hướng đối tượng, chỉ tập trung vào phần lập trình hướng cấu trúc.

Về cơ bản, môn Tin học cũng giống như môn Toán. Chúng ta tập trung chủ yếu vào phần thực hành. Tác giả cũng không đi sâu vào khai thác các vấn đề về lý thuyết mà thông qua các bài tập sẽ giúp cho các em học sinh có thể dễ dàng nắm bắt được các kiến thức thông qua việc giải quyết các bài toán cụ thể.

Tài liệu được xây dựng từ những vấn đề đơn giản đến phức tạp theo các nội dung cơ bản của ngôn ngữ lập trình C++.

Tác giả rất mong nhận được sự đóng góp ý kiến của quý thầy / cô cũng như các em học sinh tham gia ôn luyện và thi học sinh giỏi bộ môn Tin học khối THCS.

Xin trân trọng cảm ơn./.

## MỤC LỤC

1. Giới thiệu ngôn ngữ lập trình C++ .....	1
2. Môi trường lập trình C++ .....	1
3. Các kiểu dữ liệu.....	2
4. Biến, hằng, các phép toán và toán tử.....	3
5. Hàm .....	4
6. Tệp và ghi tệp.....	5
7. Cấu trúc điều kiện.....	9
8. Vòng lặp.....	18
9. Mảng.....	35
10. Xâu ký tự (string) .....	55

## 1. Giới thiệu ngôn ngữ lập trình C++

C++ là ngôn ngữ lập trình phổ biến nhất trên thế giới được phát triển bởi Bjarne Stroustrup. C++ là một phần mở rộng của ngôn ngữ C. Về cơ bản C và C++ gần như là sự tương đồng.

C++ cung cấp cho người lập trình khả năng kiểm soát và tối ưu hóa đối với tài nguyên hệ thống và bộ nhớ, được sử dụng để phát triển các hệ điều hành phổ biến như hệ điều hành windows, linux, ios..., phát triển giao diện người dùng đồ họa và các hệ thống nhúng.

C++ là một ngôn ngữ lập trình hướng đối tượng mang lại cấu trúc rõ ràng cho các chương trình và cho phép mã được sử dụng lại, giảm chi phí phát triển.

C++ là nền tảng để phát triển các ngôn ngữ ứng dụng bậc cao hơn như PHP, Java, javascript, C#, ...

## 2. Môi trường lập trình C++

Để làm việc với C++ bạn cần phải có trình soạn thảo và cài đặt môi trường để biên dịch mã. Ở đây, có một số công cụ hoàn toàn miễn phí và sử dụng phổ biến như **Dev C++**, **Eclipse**, **Code::Block**, **Visual Studio**... Các phần mềm này có sự hỗ trợ cho việc soạn thảo cũng như tạo sẵn môi trường để biên dịch mã C++.

Để bắt đầu với C++, chúng ta đến với chương trình Hello World:

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello World!";
    return 0;
}
```

Trong đó:

- Dòng **#include <iostream>** là thư viện được khai báo trong mỗi chương trình C++. Đây là môi trường nhập / xuất (input / output) dữ liệu trong C++. Ngoài ra, với các bài toán chúng ta cần quan tâm đến 2 thư viện toán học đặc biệt quan trọng là **cmath** và **algorithm** ngoài ra chúng ta cũng cần bổ sung thêm thư viện **cstring** để xử lý các bài toán về chuỗi ký tự.

- Dòng **using namespace std** là không gian tên. Nếu ko có dòng này bạn cần nhập **std::cout << "Hello World!";**

- Toàn bộ chương trình chính sẽ nằm trong phần:

```
int main() {
    // Nội dung lệnh
    return 0;
}
```

**Lưu ý:**

- Mọi câu lệnh trong C++ đều kết thúc bằng dấu “;”.

---

- Tất cả các câu lệnh, từ khóa trong C++ đều là chữ **in thường**.
- Để nhập dữ liệu ta sử dụng **cin**, xuất dữ liệu ta sử dụng **cout**.

Ví dụ:

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

int main() {
    int a;
    cin >> a;
    cout << "a = " << a;
    return 0;
}
```

- Nếu bạn bỏ qua không gian tên **using namespace std** thì câu lệnh nhập / xuất dữ liệu sẽ cần khai báo **std**:

```
#include <iostream>
int main() {
    std::cout << "Hello World!";
    return 0;
}
```

- Hai câu lệnh **return 0** và **return 1** là hai giá trị trả về duy nhất của hàm **main()** trong C++.

Hai giá trị trả về này của hàm **int main()** có ý nghĩa như sau:

Chúng ta chỉ định **return 0** để kết thúc chương trình theo cách bình thường (normal termination). Điều đó có nghĩa là kể cả chương trình có xảy ra lỗi hay không, thì C++ vẫn ngầm định là chương trình đã được kết thúc mà không có lỗi xảy ra.

Chúng ta chỉ định **return 1** để kết thúc chương trình theo cách bất thường (abnormal termination). Điều đó có nghĩa là khi chương trình xảy ra lỗi, thì lỗi này sẽ được trả về khi kết thúc chương trình.

### 3. Các kiểu dữ liệu

Với các bài toán xử lý trên C++, chúng ta cần quan tâm một số kiểu dữ liệu như sau:

- Kiểu số nguyên **int** là kiểu dữ liệu thường xuyên sử dụng
  - Kiểu **string**, **char**: khai báo trong những bài toán về chuỗi ký tự
  - Kiểu **bool**: Trả về **true** (1) hoặc **false** (0)
  - Kiểu **float**, **double**: trả về kiểu số thực
  - Kiểu **long**: Trả về kiểu số nguyên lớn hơn **int**
-

Type name	Bytes	Range of values
bool	1	false or true
char	1	-128 to 127
int	4	-2147483648 to 2147483647
unsigned int	4	0 to 4294967295
__int16	2	-32768 to 32767
__int32	4	-2147483648 to 2147483647
__int64	8	-9223372036854775808 to 9223372036854775807
float	4	3.4E +/- 38 (7 digits)
double	8	1.7E +/- 308 (15 digits)

Tùy theo tình huống và dạng bài tập mà chúng ta sử dụng dữ liệu làm sao cho hợp lý. Có những bài toán yêu cầu về số nguyên lớn ta có thể sử dụng `int8_t`, `int32_t`, `int64_t` để tránh trường hợp file test với số nguyên lớn làm bạn mất điểm do bị tràn số. Còn với những trường hợp không yêu cầu về số nguyên lớn ta có thể sử dụng kiểu thông dụng là `int`.

Các kiểu dữ liệu thường dùng là `int`, `char`, `string`, `float`, `double`

#### 4. Biến, hằng, các phép toán và toán tử

- **Khai báo biến:** Việc khai báo biến trong C++ rất đơn giản, cấu trúc chung thường là:

```
Kiểu_dữ_liệu Tên_biến;
```

```
Hoặc: Kiểu_dữ_liệu biến1, biến2, biến3;
```

```
Ví dụ: int a, b, c;
```

Bạn hoàn toàn có thể khai báo biến đồng thời với việc gán một giá trị cụ thể nào đó cho biến.

```
Ví dụ: int a = 50, b = 100;
```

Bạn cần phân biệt **biến toàn cục và biến cục bộ**: Biến toàn cục là biến chạy trong toàn chương trình, còn biến cục bộ chỉ chạy nội bộ trong một hàm.

- **Khai báo hằng:** Việc khai báo hằng cũng giống kiểu khai báo biến

```
Ví dụ: const int a = 50;
```

- **Các phép toán:** C++ sử dụng các phép toán cơ bản giống như các ngôn ngữ lập trình khác:

+ Phép cộng: +

+ Phép trừ: -

+ Phép nhân: \*

+ Phép chia: /

+ Phép chia có dư: %

- **Toán tử:** Chúng ta cần lưu ý một số toán tử thông dụng trong C++

Toán tử	Ý nghĩa	Ví dụ
==	Bằng	$x == y$
!=	Không bằng	$x != y$
>	Lớn hơn	$x > y$
<	Nhỏ hơn	$x < y$
>=	Lớn hơn hoặc bằng	$x >= y$
<=	Nhỏ hơn hoặc bằng	$x <= y$

- **Toán tử logic:** Chúng ta quan tâm đến 3 toán tử logic thường dùng:

Toán tử	Ý nghĩa	Ví dụ
&&	AND	$x < 5 \ \&\& \ x < 10$
	OR	$x < 5 \    \ x < 4$
!	NOT	$!(x < 5 \ \&\& \ x < 10)$

- **Toán tử gán:** Giữ vai trò đặc biệt quan trọng trong lập trình. Chúng ta cần chú ý một số phép gán trong C++ dưới đây:

Toán tử	Ví dụ	Tương đương
=	$x = 5$	$x = 5$
+=	$x += 3$	$x = x + 3$
-=	$x -= 3$	$x = x - 3$
*=	$x *= 3$	$x = x * 3$
/=	$x /= 3$	$x = x / 3$
%=	$x \% = 3$	$x = x \% 3$
++	$x++$	$x = x + 1$
--	$x--$	$x = x - 1$

## 5. Hàm

Hàm là một nhóm các lệnh đi cùng nhau để thực hiện một nhiệm vụ nào đó trong chương trình. Mỗi chương trình C++ có ít nhất một hàm: **int main()**

Cấu trúc:

```

Kiểu_trả_về tên_Hàm(Danh_sách_tham_số) {
    Thân_hàm
}

```

Trong đó:

**Kiểu trả về:** Một hàm có thể trả về một giá trị. Kiểu\_trả\_về là dạng dữ liệu của giá trị mà hàm trả về. Vài hàm cung cấp các hoạt động và không trả về giá trị nào cả. Đó là hàm **void**.

**Tên hàm:** Đây là tên thực sự của hàm. Tên hàm và danh sách tham số cấu tạo nên dấu hiệu hàm.

**Danh sách tham số:** Khi hàm được gọi, bạn phải truyền vào danh sách các tham số. Một giá trị hướng đến một tham số thực tế. Danh sách tham số có các kiểu, thứ tự và số lượng các tham số của hàm. Các tham số trong hàm là tùy chọn, nghĩa là một hàm có thể không có tham số.

**Thân hàm:** Phần thân của một hàm bao gồm tập hợp các lệnh xác định những gì mà hàm thực hiện.

Để gọi **hàm**, đơn giản là chỉ cần truyền các tham số được yêu cầu cùng với tên của hàm và nếu hàm trả về các giá trị, bạn có thể dự trữ các giá trị trả về. Xét một ví dụ đơn giản về hàm tính tổng hai số:

```
#include <iostream>
using namespace std;
int sum(int a, int b) {
    int ketqua;
    ketqua = a + b;
    return ketqua;
}
int main () {
    // Khai bao bien cuc bo:
    int a = 100;
    int b = 200;
    int ketqua;
    // goi ham de tinh tong hai so.
    ketqua = sum(a, b);
    cout << "Tong gia tri la: " << ketqua << endl;
    // goi ham mot lan nua.
    ketqua = sum(a);
    cout << "Tong gia tri la: " << ketqua << endl;
    return 0;
}
```

## 6. Tập và ghi tập

Với việc ghi và đọc file chúng ta có thể sử dụng một trong 2 cách:

**Cách 1:** Sử dụng thư viện **fstream**

```
#include <iostream>
#include <fstream>
using namespace std;
int main() {
    ifstream fin;
    ofstream fout;
    int a, b, Tong = 0, Tich = 0;
    fin.open("Tong.INP",ios::in);
    fout.open("Tong.OUT",ios::out);
    fin>>a;
    fin>>b;
    fin.close();
    fout << a + b << endl;
    fout << a * b;
    fout.close();
    return 0;
}
```

## Cách 2: Sử dụng Hàm **freopen**:

```
#include <iostream>
using namespace std;
int main() {
    int a;
    freopen("TongChuSo.INP", "r", stdin);
    freopen("TongChuSo.OUT", "w", stdout);
    cin >> a;
    int c = a % 10;
    int d = a / 10;
    cout << c + d;
    return 0;
}
```

### Bài tập

**Bài 1.** Nhập số nguyên dương  $n$  là số có hai chữ số. Tính tổng các chữ số của  $n$ . Ví dụ:

PERM.INP	PERM.OUT
34	7

**Thuật toán:** Ta có  $\overline{ab} = 10a + b$ . Như vậy: nếu chia số nguyên  $a$  cho 10 ta được số dư là  $b$ , còn phần nguyên chính là  $a$ .

Code tham khảo:

```
#include <iostream>
using namespace std;
int main() {
    freopen("PERM.INP", "r", stdin);
    freopen("PERM.OUT", "w", stdout);
    int a;
    cin >> a;
    int T = a % 10 + a / 10;
    cout << T;
    return 0;
}
```

**Bài 2.** Nhập số nguyên dương  $N$  là số có 3 chữ số. Tìm tích các chữ số của  $N$ . Ví dụ:

TICH.INP	TICH.OUT
254	40

Code tham khảo:

```
#include <iostream>
using namespace std;
int main() {
    freopen("TICH.INP", "r", stdin);
    freopen("TICH.OUT", "w", stdout);
    int n;
```

```

cin >> n;
int T = n % 10;
n /= 10;
T *= n % 10;
T *= n / 10;
cout << T;
return 0;
}

```

**Bài 3.** Nhập số n nguyên dương ( $n \leq 2^{32}$ ). Tính  $S = \frac{n(n+1)(n+2)}{3}$

Ví dụ:

TS.INP	TS.OUT
7	168

**Thuật toán:** Để có được các chữ số của N, ta chia lần lượt cho 10 lấy phần nguyên và phần dư, phần dư đầu tiên sẽ là chữ số hàng đơn vị, phần dư lần chia thứ 2 sẽ là chữ số hàng chục và phần nguyên của lần chia thứ 2 sẽ là chữ số hàng trăm.

**Code tham khảo:**

```

#include <iostream>
using namespace std;
int main() {
    freopen("TS.INP", "r", stdin);
    freopen("TS.OUT", "w", stdout);
    int64_t n;
    cin >> n;
    int64_t s = n*(n+1)*(n+2) / 3;
    cout << s;
    return 0;
}

```

**Bài 4.** Nhập số thời gian là T (giây). Người ta đổi T thành a giờ, b phút, c giây.

Hãy tìm a, b, c. Ví dụ:

TIME.INP	TIME.OUT
6695	1:51:35

**Thuật toán:**

Ta có: 1 giờ = 3600 giây

1 phút = 60 giây

- Lấy số giây chia cho 3600 tính được số giờ.
- Tiếp tục chia phần dư cho 60 được số phút
- Cuối cùng, phần dư còn lại sẽ là số giây.

**Code tham khảo:**

```

#include <iostream>
using namespace std;
int main() {
    freopen("TIME.INP", "r", stdin);
    freopen("TIME.OUT", "w", stdout);
    int t;
    cin >> t;
    int a = t / 3600;
    t %= 3600;
    int b = t / 60;
    int c = t % 60;
    cout << a << ":" << b << ":" << c;
    return 0;
}

```

**Bài 5.** Nam có X đồng. Biết rằng tờ tiền có mệnh giá là 100, 50, 20, 10, 5, 2, 1. Hỏi số lượng tờ tiền ít nhất mà Nam cầm trên tay là bao nhiêu? Liệt kê từng loại tiền mà Nam có. Ví dụ: Dòng thứ nhất ghi tổng số tờ tiền, dòng thứ hai ghi số tờ tiền của từng loại (mỗi loại cách nhau một dấu cách). Ví dụ:

VND.INP	VND.OUT
6577	69
	65 11 0 1 1 0

Code tham khảo:

```

#include <iostream>
using namespace std;
int main() {
    freopen("VND.INP", "r", stdin);
    freopen("VND.OUT", "w", stdout);
    int t;
    cin >> t;
    int a = t / 100;
    t %= 100;
    int b = t / 50;
    t %= 50;
    int c = t / 20;
    t %= 20;
    int d = t / 10;
    t %= 10;
    int e = t / 5;
    t %= 5;
    int f = t / 2;
    t %= 2;
    int g = t;
    cout << a+b+c+d+e+f+g << endl;
    cout << a << " " << b << " " << c << " "
        << d << " " << e << " " << f << " " << g;
    return 0;
}

```

---

```
}
```

## 7. Cấu trúc điều kiện

Cũng giống như các ngôn ngữ lập trình khác, trong C++ chúng ta có thể sử dụng một trong 3 cấu trúc: **if**, **if... else**, **if... else if... else** hoặc **switch ... case**. Đây là những cấu trúc phổ biến và rất quen thuộc trong lập trình.

Cú pháp câu lệnh **if**:

```
if (điều kiện) {  
    // khối mã được thực thi nếu điều kiện là đúng  
}
```

Cú pháp câu lệnh **if ... else**

```
if (điều kiện) {  
    // khối mã được thực thi nếu điều kiện là đúng  
}  
else {  
    // Khối lệnh thực thi nếu điều kiện là sai  
}
```

Cú pháp câu lệnh **if ... else if ... else**:

```
if (điều kiện) {  
    // khối mã thực thi nếu điều kiện 1 là đúng  
}  
else if (điều kiện) {  
    // khối mã thực thi nếu điều kiện 2 là đúng  
} else {  
    // Khối lệnh thực thi nếu 2 đk trên là sai  
}
```

Cú pháp câu lệnh **switch ... case**

```
switch(expression) {  
    case x:  
        // code block  
        break;  
    case y:  
        // code block  
        break;  
    default:  
        // code block  
}
```

Ví dụ 1. Giải và biện luận phương trình bậc nhất  $ax + b = 0$  với khối lệnh điều kiện **if ... else**

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

---

```

int main() {
    freopen("PTBN.INP", "r", stdin);
    freopen("PTBN.OUT", "w", stdout);
    float a, b;
    cin >> a >> b;
    if (a == 0) {
        if (b == 0) {
            cout << "PT co vo so nghiem";
        }
        else {
            cout << "PT vo nghiem";
        }
    }
    else {
        cout << "x = " << -b/a;
    }
    return 0;
}

```

Ví dụ 2. Xét một nội dung sử dụng cấu trúc **switch ... case:**

```

int day = 4;
switch (day) {
    case 1:
        cout << "Monday";
        break;
    case 2:
        cout << "Tuesday";
        break;
    case 3:
        cout << "Wednesday";
        break;
    case 4:
        cout << "Thursday";
        break;
    case 5:
        cout << "Friday";
        break;
    case 6:
        cout << "Saturday";
        break;
    case 7:
        cout << "Sunday";
        break;
}

```

Ngoài cách trên, ta có thể có cách viết tắt của câu lệnh **if:**

```

#include <iostream>
#include <string>

```

---

```

using namespace std;
int main() {
    int time = 20;
    string result = (time < 18)
        ? "Good day." : "Good evening.";
    cout << result;
    return 0;
}

```

Câu lệnh trên tương đương với câu lệnh dưới đây:

```

#include <iostream>
#include <string>
using namespace std;
int main() {
    int time = 20;
    string result;
    if (time < 20) result = "Gooday";
    else result = "Good evening";
    cout << result;
    return 0;
}

```

### Bài tập

**Bài 1.** Nhập số nguyên dương x, kiểm tra xem x là số chẵn hay số lẻ?

Ví dụ:

CHANLE.INP	CHANLE.OUT
45	LE

Code tham khảo:

```

#include <iostream>
using namespace std;
int main() {
    freopen("CHANLE.INP", "r", stdin);
    freopen("CHANLE.OUT", "w", stdout);
    int n;
    cin >> n;
    if (n % 2 == 0) cout << "CHAN";
    else cout << "LE";
    return 0;
}

```

**Bài 2.** Nhập số nguyên n. Nếu n là số chính phương thông báo TRUE, ngược lại thông báo FALSE. Ví dụ:

CP.INP	CP.OUT
45	FALSE

Code tham khảo:

---

```

#include <iostream>
#include <cmath>
using namespace std;
int main() {
    freopen("CP.INP", "r", stdin);
    freopen("CP.OUT", "w", stdout);
    int a;
    cin >> a;
    if ((float)sqrt(a) == (int)sqrt(a)) cout << "TRUE";
    else cout << "FALSE";
    return 0;
}

```

**Bài 3.** Nhập số nguyên a và b. Tìm giá trị lớn nhất của a, b. Ví dụ:

MAX.INP	MAX.OUT
39 90	90

Code tham khảo:

```

#include <iostream>
#include <cmath>
using namespace std;
int main() {
    freopen("MAX.INP", "r", stdin);
    freopen("MAX.OUT", "w", stdout);
    int a, b;
    cin >> a >> b;
    int max = a;
    if (max < b) max = b;
    cout << max;
    return 0;
}

```

**Bài 4.** Nhập số nguyên a, b, c. Tìm giá trị lớn nhất của a, b và c. Ví dụ:

MAX.INP	MAX.OUT
35 46 25	46

Code tham khảo:

```

#include <iostream>
using namespace std;
int main() {
    freopen("MAX.INP", "r", stdin);
    freopen("MAX.OUT", "w", stdout);
    int a, b, c;
    cin >> a >> b >> c;
    int max = a;
    if (max < b) max = b;
    if (max < c) max = c;
}

```

---

```

    cout << max;
    return 0;
}

```

**Bài 5.** Nhập số nguyên a, b, c, d. Tìm giá trị lớn nhất của a, b, c, d. Ví dụ:

MAX.INP	MAX.OUT
27 45 22 36	45

Code tham khảo:

```

#include <iostream>
#include <cmath>
using namespace std;
int main() {
    freopen("MAX.INP", "r", stdin);
    freopen("MAX.OUT", "w", stdout);
    int a, b, c, d;
    cin >> a >> b >> c >> d;
    int max = a;
    if (max < b) max = b;
    if (max < c) max = c;
    if (max < d) max = d;
    cout << max;
    return 0;
}

```

**Bài 6.** Nhập 2 số nguyên a, b có giá trị trong đoạn [0, 9]. Bình ghép nó thành số có hai chữ số. Tìm số lớn nhất mà Bình có thể có được. Ví dụ:

MAX.INP	MAX.OUT
6 8	86

Code tham khảo:

```

#include <iostream>
#include <cmath>
using namespace std;
int main() {
    freopen("MAX.INP", "r", stdin);
    freopen("MAX.OUT", "w", stdout);
    int a, b, c, d;
    cin >> a >> b;
    c = 10*a + b;
    d = 10*b + a;
    int max = c;
    if (max < d) max = d;
    cout << max;
    return 0;
}

```

**Bài 7.** Nhập a, b nguyên. Hãy giải và biện luận phương trình  $ax + b = 0$ .

---

- Nếu vô nghiệm thông báo None.
- Nếu vô số nghiệm thông báo Multiple.
- Nếu có nghiệm, đưa ra dạng  $x = -b / a$ ;

Ví dụ:

PTBN.INP	PTBN.OUT
0 6	None

Code tham khảo:

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    freopen("PTBN.INP", "r", stdin);
    freopen("PTBN.OUT", "w", stdout);
    int a, b;
    cin >> a >> b;
    if (a == 0)
        if (b == 0) cout << "Multiple";
        else cout << "None";
    else cout << (float)-b / a;
    return 0;
}
```

**Bài 8.** Nhập a, b, c nguyên. Giải và biện luận phương trình  $ax^2 + bx + c = 0$ .

Ví dụ:

PTBH.INP	PTBH.OUT
1 -3 2	1 2

Code tham khảo:

```
#include<iostream>
#include <math.h>
using namespace std;
int main() {
    freopen("PTBH.INP", "r", stdin);
    freopen("PTBH.OUT", "w", stdout);
    int a, b, c, d;
    cin >> a >> b >> c;
    d = b*b - 4*a*c;
    if (d < 0) cout << "Vo nghiem";
    else if (d == 0) {
        double x = -b / (2*a);
        cout << x;
    }
    else {
        double x1 = (-b - sqrt(d))/(2*a);
        double x2 = (-b + sqrt(d))/(2*a);
    }
}
```

```

        cout << x1 << " " << x2;
    }
    return 0;
}

```

**Bài 9.** Nhập số thực x. Tìm số nguyên a lớn nhất và b nhỏ nhất sao cho  $a < x < b$ . Ví dụ:

SOTHUC.INP	SOTHUC.OUT
3.14	3 4

Code tham khảo:

```

#include <iostream>
#include <cmath>
using namespace std;
int main() {
    freopen("SOTHUC.INP", "r", stdin);
    freopen("SOTHUC.OUT", "w", stdout);
    float a;
    cin >> a;
    cout << (int)a << " " << (int)a + 1;
    return 0;
}

```

**Bài 10.** Nhập 2 số nguyên dương a, b. Nếu a là số chẵn thì giảm a đi một nửa và tăng b gấp đôi. Ngược lại, nếu a là số lẻ thì tăng a một đơn vị và tăng b thêm một lượng là a. Ví dụ:

TANGGIAM.INP	TANGGIAM.OUT
4 5	2 10

Code tham khảo:

```

#include <iostream>
#include <cmath>
using namespace std;
int main() {
    freopen("TANGGIAM.INP", "r", stdin);
    freopen("TANGGIAM.OUT", "w", stdout);
    int a, b;
    cin >> a >> b;
    if (a % 2 == 0) {
        a = a / 2;
        b = b * 2;
    }
    else {
        b += a;
        a += 1;
    }
    cout << a << " " << b;
}

```

```
    return 0;
}
```

**Bài 11.** Tìm chữ số lớn nhất trong số nguyên có 3 chữ số. Ví dụ:

MAX.INP	MAX.OUT
486	8

Code tham khảo:

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    freopen("MAX.INP", "r", stdin);
    freopen("MAX.OUT", "w", stdout);
    int n;
    cin >> n;
    int a = n % 10;
    n /= 10;
    int b = n % 10;
    n /= 10;
    int c = n;
    int max = a;
    if(max < b) max = b;
    if(max < c) max = c;
    cout << max;
    return 0;
}
```

**Bài 12.** Cho số nguyên dương  $n$  là số có 3 chữ số. Nam thực hiện xóa đi 1 chữ số trong 3 chữ số của  $n$  để thu được số  $n$  mới là số có 2 chữ số. Hãy tìm số  $n$  lớn nhất có thể. Ví dụ:

MAX.INP	MAX.OUT
486	86

Code tham khảo:

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    freopen("MAX.INP", "r", stdin);
    freopen("MAX.OUT", "w", stdout);
    int n;
    cin >> n;
    int a = n % 10;
    n /= 10;
    int b = n % 10;
    n /= 10;
```

```

int c = n;
int a1 = 10*a + b;
int a2 = 10*a + c;
int a3 = 10*b + a;
int a4 = 10*b + c;
int a5 = 10*c + a;
int a6 = 10*c + b;
int max = a1;
if (max < a2) max = a2;
if (max < a3) max = a3;
if (max < a4) max = a4;
if (max < a5) max = a5;
if (max < a6) max = a6;
cout << max;
return 0;
}

```

**Bài 13.** Dung có a đồng muốn đổi tiền gồm các tờ mệnh giá 50, 20 và 10. Hỏi Dung có thể đổi tiền được không? Nếu có ghi ra số tờ tiền ít nhất Dung có được, nếu không đổi được thì ghi ra -1. Ví dụ:

TIEN.INP	TIEN.OUT
440	1

Code tham khảo:

```

#include <iostream>
#include <cmath>
using namespace std;
int main() {
    freopen("TIEN.INP", "r", stdin);
    freopen("TIEN.OUT", "w", stdout);
    int a;
    cin >> a;
    a %= 50;
    a %= 20;
    a %= 10;
    if (a == 0) cout << 1;
    else cout << -1;
    return 0;
}

```

**Bài 14.** Thảo ghi lên bảng 4 số nguyên bất kì. Sau đó Thảo muốn biết trong 4 số trên, có số nào là số trung bình cộng hay không? Nếu có ghi ra 1 ngược lại ghi ra -1. Biết số trung bình cộng bằng tổng 4 số đó chia cho 4.

Ví dụ:

NUMBER.INP	NUMBER.OUT
4 6 7 8	-1

Code tham khảo:

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    freopen("NUMBER.INP","r",stdin);
    freopen("NUMBER.OUT","w",stdout);
    int a, b, c, d;
    cin >> a >> b >> c >> d;
    float tb = (float)(a+b+c+d)/4;
    if(tb==a || tb==b || tb==c || tb==d)
        cout << 1;
    else cout << -1;
    return 0;
}
```

**Bài 15.** Cho số  $n$  nguyên dương là số có 3 chữ số. Dũng muốn biết 3 chữ số của số  $n$  có là độ dài ba cạnh của một tam giác hay không? Nếu có ghi ra diện tích của tam giác đó. Nếu không ghi ra -1. Ví dụ:

TAMGIAC.INP	TAMGIAC.OUT
3 4 5	6

Code tham khảo:

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    freopen("TAMGIAC.INP","r",stdin);
    freopen("TAMGIAC.OUT","w",stdout);
    int a, b, c;
    cin >> a >> b >> c;
    if (a+b>c && a+c>b && b+c>a) {
        float p = (float) (a+b+c) / 2;
        cout << sqrt(p*(p-a)*(p-b)*(p-c));
    }
    else cout << -1;
    return 0;
}
```

## 8. Vòng lặp

Gồm 3 loại vòng lặp: for, while và do... while

### a) Vòng lặp for

Cú pháp:

```
for (bien; dieu_kien; tang_giam) {
    cac_lenh;
}
```

### ***b) Vòng lặp while***

Thực hiện lặp đi lặp lại một lệnh mục tiêu đến khi nào điều kiện đã cho còn là đúng.

Cú pháp:

```
while(dieu_kien) {  
    cac_lenh;  
}
```

### ***c) Vòng lặp do... while***

Tương tự vòng lặp while, khác ở chỗ nó sẽ kiểm tra điều kiện ở cuối vòng lặp. Đảm bảo vòng lặp được thực hiện ít nhất 1 lần

Cú pháp:

```
do {  
    cac_lenh;  
}  
while(dieu_kien);
```

Xét một ví dụ đơn giản về 3 loại vòng lặp:

```
#include <iostream>  
using namespace std;  
  
int main() {  
    int i;  
    for (i=1; i <= 10; i++) {  
        cout << i << " ";  
    }  
    i = 1; cout << endl;  
    while(i <= 10) {  
        cout << i << " ";  
        i++;  
    }  
    i = 1; cout << endl;  
    do {  
        cout << i << " ";  
        i++;  
    }  
    while(i <= 10);  
    return 0;  
}
```

### **Bài tập**

**Bài 1.** Liệt kê các số nguyên dương nhỏ hơn hoặc bằng n. Ví dụ:

LIETKE.INP	LIETKE.OUT
6	1 2 3 4 5 6

Code tham khảo:

```
#include <iostream>
using namespace std;
int main () {
    freopen("LIETKE.INP", "r", stdin);
    freopen("LIETKE.OUT", "w", stdout);
    int n;
    cin >> n;
    for (int i=1; i<=n; i++) {
        cout << i << " ";
        if (i%10==0) cout << endl;
    }
    return 0;
}
```

**Bài 2.** Nhập số nguyên dương n. Liệt kê các số chia hết cho 5 nhỏ hơn hoặc bằng n. Ví dụ:

LIETKE.INP	LIETKE.OUT
26	5 10 15 20 25

Code tham khảo:

```
#include <iostream>
using namespace std;
int main () {
    freopen("LIETKE.INP", "r", stdin);
    freopen("LIETKE.OUT", "w", stdout);
    int n;
    cin >> n;
    for (int i=1; i<=n; i++) {
        if(i % 5==0) cout << i << " ";
        if (i % 50==0) cout << endl;
    }
    return 0;
}
```

**Bài 3.** Nhập số nguyên dương n. Liệt kê các số chia hết cho 3 nhưng không chia hết cho 5 nhỏ hơn hoặc bằng n. Ví dụ:

LIETKE.INP	LIETKE.OUT
25	3 6 9 12 18 21 24

Code tham khảo:

```
#include <iostream>
using namespace std;
int main () {
    freopen("LIETKE.INP", "r", stdin);
    freopen("LIETKE.OUT", "w", stdout);
    int n;
    cin >> n;
```

```

    for (int i=1; i<=n; i++) {
        if(i%5!=0 && i%3==0) cout << i << " ";
        if (i%36==0) cout << endl;
    }
    return 0;
}

```

**Bài 4.** Cho số nguyên dương n. Tính tổng  $1 + 2 + 3 + \dots + n$ . Ví dụ:

TONG.INP	TONG.OUT
10	55

Code tham khảo:

```

#include <iostream>
using namespace std;
int main () {
    freopen("TONG.INP", "r", stdin);
    freopen("TONG.OUT", "w", stdout);
    int n;
    cin >> n;
    int tong = 0;
    for (int i=1; i<=n; i++) {
        tong += i;
    }
    cout << tong;
    return 0;
}

```

**Bài 5.** Cho số nguyên dương n. Tính tổng  $2022 + 2 + 4 + 6 + \dots + 2n$ . Ví dụ:

TONG.INP	TONG.OUT
20	2442

Code tham khảo:

```

#include <iostream>
using namespace std;
int main () {
    freopen("TONG.INP", "r", stdin);
    freopen("TONG.OUT", "w", stdout);
    int n;
    cin >> n;
    int tong = 2022;
    for (int i=1; i<=2*n; i++) {
        if (i%2 == 0) tong += i;
    }
    cout << tong;
    return 0;
}

```

**Bài 6.** Cho số nguyên dương n. Tính tổng  $1 + 3 + 5 + \dots + (2n-1)$ . Ví dụ:

TONG.INP	TONG.OUT
50	2500

Code tham khảo:

```
#include <iostream>
using namespace std;
int main () {
    freopen("TONG.INP", "r", stdin);
    freopen("TONG.OUT", "w", stdout);
    int n;
    cin >> n;
    int tong = 0;
    for (int i=1; i<=(2*n-1); i+=2) {
        tong += i;
    }
    cout << tong;
    return 0;
}
```

**Bài 7.** Nhập n. Liệt kê và tính tổng các ước nguyên dương của n. Ví dụ:

UND.INP	UND.OUT
24	1 2 3 4 6 8 12 24 60

Code tham khảo:

```
#include <iostream>
using namespace std;
int main () {
    freopen("UND.INP", "r", stdin);
    freopen("UND.OUT", "w", stdout);
    int n;
    cin >> n;
    int tong = 0;
    for (int i = 1; i <= n; i++) {
        if (n % i == 0) {
            tong += i;
            cout << i << " ";
        }
    }
    cout << endl << tong;
    return 0;
}
```

**Bài 8.** Nhập số n, kiểm tra xem n có là số nguyên tố hay không? Nếu là nguyên tố ghi 1, ngược lại ghi -1. Ví dụ:

NGUYENTO.INP	NGUYENTO.OUT
17	1

---

Code tham khảo:

```
#include <iostream>
#include <cmath>
using namespace std;
int nguyenTo(int n) {
    if (n < 2) return 0;
    for (int i = 2; i <= sqrt(n); i++) {
        if (n % i == 0) return 0;
    }
    return 1;
}
int main() {
    freopen("NGUYENTO.INP", "r", stdin);
    freopen("NGUYENTO.OUT", "w", stdout);
    int n;
    cin >> n;
    if (nguyenTo(n)) cout << 1;
    else cout << -1;
    return 0;
}
```

**Bài 9.** Nhập số n, liệt kê các số nguyên tố nhỏ hơn hoặc bằng n. Ví dụ:

NGUYENTO.INP	NGUYENTO.OUT
17	2 3 5 7 11 13 17

Code tham khảo:

```
#include <iostream>
#include <cmath>
using namespace std;
int nguyenTo(int n) {
    if (n < 2) return 0;
    for (int i = 2; i <= sqrt(n); i++) {
        if (n % i == 0) return 0;
    }
    return 1;
}
int main() {
    freopen("NGUYENTO.INP", "r", stdin);
    freopen("NGUYENTO.OUT", "w", stdout);
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) {
        if(nguyenTo(i)) {
            cout << i << " ";
        }
    }
    return 0;
}
```

---

**Bài 10.** Những số nguyên tố khi chia cho 4 dư 1 thì luôn biểu diễn được dưới dạng tổng của 2 số chính phương. Nhập số n, kiểm tra xem n có là số nguyên tố hay không? Nếu thỏa mãn ghi ra hai số chính phương có tổng bằng n. Ngược lại, ghi ra -1. Ví dụ:

BDNT.INP	BDNT.OUT
61	25 36

Code tham khảo:

```
#include <iostream>
#include <cmath>
using namespace std;
int nguyenTo(int n) {
    if (n < 2) return 0;
    for (int i = 2; i <= sqrt(n); i++) {
        if (n % i == 0) return 0;
    }
    return 1;
}
int cp(int n) {
    if ((float)sqrt(n) == (int)sqrt(n))
        return 1;
    else return 0;
}
int main() {
    freopen("BDNT.INP", "r", stdin);
    freopen("BDNT.OUT", "w", stdout);
    int n;
    cin >> n;
    if(nguyenTo(n)) {
        if (n % 4 == 1 || n == 2) {
            int a = 1;
            while (!cp(a) || !cp(n-a)) {
                a++;
            }
            cout << a << " " << n-a;
        } else cout << -1;
    } else cout << -1;
    return 0;
}
```

**Bài 11.** Tìm số nguyên dương  $n = x^2 = y^3$  (với x, y nguyên, n có 4 chữ số),

Với bài toán này, ta cần giải bằng thuật toán.

Vì  $1000 \leq y^3 \leq 9999$ .

Từ đó suy ra:  $10 \leq y \leq 21$ .

Chính vì vậy ta sử dụng vòng lặp for duyệt từ 10 đến 21, tìm ra một số chính phương bằng  $y^3$  thỏa mãn là một số chính phương. Trong bài toán này, chúng ta

chỉ cần tìm số nguyên dương  $n$  thỏa mãn điều kiện ghi vào file CPLP.OUT với dòng thứ nhất ghi ra số  $n$  cần tìm và dòng thứ hai ghi ra hay số  $x, y$  (cách nhau một dấu cách). Số cần tìm là số  $4096 = 16^3 = 64^2$ .

Code tham khảo như sau:

```
#include <iostream>
#include <cmath>
using namespace std;
int cp(int n) {
    if ((float)sqrt(n) == (int)sqrt(n))
        return 1;
    else return 0;
}
int main() {
    freopen("CPLP.OUT", "w", stdout);
    for (int i = 10; i < 22; i++) {
        if (cp(i*i*i)) {
            cout << i*i*i << endl;
            cout << i << " " << sqrt(i*i*i);
        }
    }
    return 0;
}
```

**Bài 12.** Tìm tất cả các số có 3 chữ số sao cho:  $\overline{abc} = a^3 + b^3 + c^3$ . Kết quả tìm được lưu vào tệp LAPPHUONG.OUT là dãy số có 3 chữ số thỏa mãn điều kiện trên (mỗi số cách nhau một dấu cách)

Cách 1:

```
#include <iostream>
using namespace std;
int main() {
    freopen("LAPPHUONG.OUT", "w", stdout);
    for (int i = 100; i < 1000; i++) {
        int a = i / 100;
        int b = (i % 100) / 10;
        int c = (i % 100) % 10;
        if (i == a*a*a + b*b*b + c*c*c)
            cout << a << b << c << " ";
    }
    return 0;
}
```

Cách 2:

```
#include <iostream>
using namespace std;
int main() {
    freopen("LAPPHUONG.OUT", "w", stdout);
    for (int a = 1; a <= 9; a++) {
```

```

        for (int b = 0; b <=9; b++) {
            for (int c = 0; c <= 9; c++) {
                if (100*a + 10*b + c == a*a*a + b*b*b +
c*c*c)
                    cout << a << b << c << " ";
            }
        }
    }
    return 0;
}

```

**Bài 13.** Số n được gọi là số hoàn thiện nếu tổng các ước của n gấp đôi n.

Ví dụ: 6 là số hoàn thiện vì 6 chia hết cho 1, 2, 3, 6 và:  $1 + 2 + 3 + 6 = 12 = 2.6$

- Kiểm tra n có là số hoàn thiện không? Nếu có ghi ra 1 ngược lại ghi 0
- Tìm các số hoàn thiện nhỏ hơn hoặc bằng n
- Tìm các số hoàn thiện nằm trên đoạn [m, n].

Yêu cầu: nhập vào 2 số m, n. Kết quả ghi thành 3 dòng thể hiện 3 nội dung trên.

Ví dụ:

SOHT.INP	SOHT.OUT
16 1000	0
	6 28 496
	28 496

Code tham khảo:

```

#include <iostream>
using namespace std;
int soht(int n) {
    int tong = 0;
    for (int i = 1; i <= n/2; i++) {
        if (n % i == 0) tong += i;
    }
    if (tong+n == 2*n) return 1;
    else return 0;
}
int main() {
    freopen("SOHT.INP", "r", stdin);
    freopen("SOHT.OUT", "w", stdout);
    int m, n;
    cin >> m >> n;
    if(soht(n)) cout << 1 << endl;
    else cout << 0 << endl;
    for (int i = 1; i <= n; i++) {
        if (soht(i)) cout << i << " ";
    }
    cout << endl;
}

```

```

    for (int i = m; i < n; i++) {
        if (soht(i)) cout << i << " ";
    }
    return 0;
}

```

**Bài 14.** Nhập số nguyên dương n. Đếm số cặp (a, b) đôi một khác nhau sao cho  $a + b = n$ . Ví dụ:

CAPSO.INP	CAPSO.OUT
10	1 9 2 8 3 7 4 6

Code tham khảo:

```

#include <iostream>
using namespace std;
int main() {
    freopen("CAPSO.INP", "r", stdin);
    freopen("CAPSO.OUT", "w", stdout);
    int n;
    cin >> n;
    for (int i = 1; i < n-1; i++) {
        for (int j = i+1; j < n; j++) {
            if (i + j == n)
                cout << i << " " << j << endl;
        }
    }
    return 0;
}

```

**Bài 15.** Nhập m, n. Tìm các số nguyên tố nằm trên đoạn [m, n]. Ví dụ:

NGUYENTO.INP	NGUYENTO.OUT
5 20	7 11 13 17 19

Code tham khảo:

```

#include <iostream>
#include <cmath>
using namespace std;
int nguyenTo(int n) {
    if (n < 2) return 0;
    for (int i = 2; i <= sqrt(n); i++) {
        if (n % i == 0) return 0;
    }
    return 1;
}

```

```

int main() {
    freopen("NGUYENTO.INP", "r", stdin);
    freopen("NGUYENTO.OUT", "w", stdout);
    int m, n;
    cin >> m >> n;
    for (int i = m; i <= n; i++) {
        if(nguyenTo(i)) {
            cout << i << " ";
        }
    }
    return 0;
}

```

**Bài 16.** Nhập số nguyên dương n. Tìm số nguyên tố nhỏ nhất lớn hơn n. Ví dụ:

NGUYENTO.INP	NGUYENTO.OUT
9	11

Code tham khảo:

```

#include <iostream>
#include <cmath>
using namespace std;
int nguyenTo(int n) {
    if (n < 2) return 0;
    for (int i = 2; i <= sqrt(n); i++) {
        if (n % i == 0) return 0;
    }
    return 1;
}
int main() {
    freopen("NGUYENTO.INP", "r", stdin);
    freopen("NGUYENTO.OUT", "w", stdout);
    int n;
    cin >> n;
    int x = n + 1;
    while (!nguyenTo(x)) x++;
    cout << x;
    return 0;
}

```

**Bài 17.** Số Fibonacci:

- Tìm số Fibonacci thứ n ( $n \leq 40$ )
- Tìm các số Fibonacci nhỏ hơn hoặc bằng n.
- Tìm các số Fibonacci nằm trên đoạn  $[m, n]$ .
- Tìm các số vừa là số Fibonacci vừa là số nguyên tố trên đoạn  $[m, n]$ .

Yêu cầu: Nhập vào số n. Trong file đầu ra ghi 4 kết quả như trên. Ví dụ:

---

FIBONACCI.INP	FIBONACCI.OUT
3 40	102334155 11 2 3 5 8 13 21 34 5 8 13 21 34 5 13

Code tham khảo:

```

#include <iostream>
#include <cmath>
using namespace std;
int nguyenTo(int n) {
    if (n < 2) return 0;
    int i;
    int s = sqrt(n);
    for (i = 2; i <= s; i++) {
        if (n % i == 0) {
            return 0;
        }
    }
    return 1;
}
int Fibonacci(int n) {
    int a1 = 1, a2 = 1;
    if (n == 1 || n == 2) return 1;
    int i = 3, a;
    while (i <= n) {
        a = a1 + a2;
        a1 = a2;
        a2 = a;
        i++;
    }
    return a;
}
int main() {
    freopen("FIBONACCI.INP", "r", stdin);
    freopen("FIBONACCI.OUT", "w", stdout);
    int m, n, i = 0;
    cin >> m >> n;
    // So Fibonacci thu n:
    cout << Fibonacci(n) << endl;
    // So Fibonacci < n:
    for (i = 0; i < n; i++) {
        if (Fibonacci(i) < n) cout<<Fibonacci(i)<<" ";
    }
    cout << endl;
    // So Fibonacci nam trong doan [m,n]:
    for (i = 0; i < n; i++) {

```

---

```

        if (Fibonacci(i) > m && Fibonacci(i) < n)
            cout << Fibonacci(i) << " ";
    }
    cout << endl;
    // So Fibonacci vua la so nguyen to doan [m,n]:
    i = m;
    while (Fibonacci(i) < n) {
        int fi = Fibonacci(i);
        if(nguyenTo(fi)) {
            if (fi > m)
                cout << fi << " ";
        }
        i++;
    }
    return 0;
}

```

**Bài 18.** Phân tích số nguyên dương  $n$  thành các thừa số nguyên tố. Ví dụ:

TSNT.INP	TSNT.OUT
150	$2 * 3 * 5^2$

Code tham khảo:

```

#include <iostream>
using namespace std;
int main() {
    freopen("TSNT.INP", "r", stdin);
    freopen("TSNT.OUT", "w", stdout);
    int n;
    cin >> n;
    int dem;
    for(int i = 2; i <= n; i++){
        dem = 0;
        while(n % i == 0){
            ++dem;
            n /= i;
        }
        if(dem){
            cout << i;
            if(dem > 1) cout << "^" << dem;
            if(n > i){
                cout << " * ";
            }
        }
    }
    return 0;
}

```

**Bài 19.** Nhập số  $n$ , kiểm tra xem  $n$  có là số chính phương không? Tìm tất cả các số chính phương dạng  $\overline{22ab}$ . Ví dụ:

FIBONACCI.INP	FIBONACCI.OUT
25	1 2209

Code tham khảo:

```
#include <iostream>
#include <cmath>
using namespace std;
int cp(int n) {
    if ((float)sqrt(n) == (int)sqrt(n))
        return 1;
    else return 0;
}
int main() {
    freopen("SOCP.INP", "r", stdin);
    freopen("SOCP.OUT", "w", stdout);
    int n;
    cin >> n;
    if(cp(n)) cout << 1 << endl;
    else cout << 0 << endl;
    for (int a = 0; a < 10; a++) {
        for (int b = 0; b < 10; b++) {
            int c = 2200 + 10*a + b;
            if (cp(c)) cout << c << " ";
        }
    }
    return 0;
}
```

**Bài 20.** Tìm ước chung lớn nhất, Bội chung nhỏ nhất của hai số  $a, b$ . Yêu cầu: Nhập vào hai số nguyên dương  $a, b$ , kết quả xuất ra theo thứ tự là ƯCLN và BCNN của hai số  $a$  và  $b$ . Ví dụ:

UCBC.INP	UCBC.OUT
24 30	6 120

Code tham khảo:

```
#include <iostream>
using namespace std;
int uc(int a, int b) {
    if(a == 0 || b == 0) { return a;}
    while (a!=b){
        if (a > b) {a -= b;}
        else {b -= a;}
    }
    return a;
}
```

---

```

}
int main() {
    freopen("UCBC.INP", "r", stdin);
    freopen("UCBC.OUT", "w", stdout);
    int a, b;
    cin >> a >> b;
    cout << uc(a, b) << " " << a*b / uc(a, b);
    return 0;
}

```

**Bài 21.** Số  $P$  là *số Mersenne* nếu  $P$  nguyên tố và  $P = 2^n - 1$ , mà  $n$  cũng là số nguyên tố.

- Kiểm tra  $P$  có là số Mersenne hay không? Nếu có ghi ra 1 ngược lại ghi 0.
- Tìm các số Mersenne nhỏ hơn hoặc bằng  $n$ .
- Tìm các số Mersenne nằm trên đoạn  $[m, n]$ .

Ví dụ:

MERSENNE.INP	MERSENNE.OUT
11 127	1
	3 7 31 127
	31 127

Code tham khảo:

```

#include <iostream>
#include <cmath>
using namespace std;
int nguyenTo(int n) {
    if (n < 2) return 0;
    for (int i = 2; i <= sqrt(n); i++) {
        if (n % i == 0) return 0;
    }
    return 1;
}
int Mersenne(int n){
    int i = 0;
    n += 1;
    while (n > pow(2, i)) {
        i++;
    }
    if (n == pow(2, i))
        return 1;
    else return 0;
}
int main() {
    freopen("MERSENNE.INP", "r", stdin);
    freopen("MERSENNE.OUT", "w", stdout);
    int m, n;

```

```

cin >> m >> n;
if (nguyenTo(n) && Mersenne(n))
    cout << 1 << endl;
else cout << 0 << endl;
for (int i = 2; i <= n; i++) {
    if (nguyenTo(i) && Mersenne(i))
        cout << i << " ";
}
cout << endl;
for (int i = m; i <= n; i++) {
    if (nguyenTo(i) && Mersenne(i))
        cout << i << " ";
}
return 0;
}

```

**Bài 22.** Con ốc sên đang ở gốc một cái cây cao  $v$  mét tính từ gốc. Ốc sên muốn bò lên ngọn cây để ăn những lá non trên đó. Ban ngày ốc sên bò được  $a$  mét lên trên, nhưng ban đêm khi ngủ nó bị trôi xuống  $b$  mét.

Yêu cầu: Cho các số nguyên  $v, a, b$  ( $1 \leq b < a \leq v \leq 10^9$ ). Hãy xác định số ngày cần thiết để ốc sên lên tới ngọn cây. Ví dụ:

MERSENNE.INP	MERSENNE.OUT
2 1 5	4

Code tham khảo:

```

#include <iostream>
using namespace std;

int main () {
    freopen("OCSEN.INP", "r", stdin);
    freopen("OCSEN.OUT", "w", stdout);
    long a, b, v;
    cin >> a >> b >> v;
    if (a==v) cout << 1;
    else {
        if ((v-b)%(a-b)==0)
            cout<<(v-b)/(a-b);
        else cout<<(v-b)/(a-b)+1;
    }
    return 0;
}

```

**Bài 23.** Tìm một số nguyên dương  $k$  nhỏ nhất sao cho tích các chữ số của  $k$  bằng  $n$  ( $k, n$  là 2 số nguyên dương). Ví dụ:

TIMSO.INP	TIMSO.OUT
72	89

Code tham khảo:

---

```

#include <iostream>
using namespace std;
int main() {
    freopen("TIMSO.INP", "r", stdin);
    freopen("TIMSO.OUT", "w", stdout);
    int n, a = 0;
    cin >> n;
    for(int i = 9; i > 1; i--) {
        while(n % i == 0 && n > 1) {
            a = a*10 + i;
            n = n/i;
        }
    }
    int b = 0;
    while(a > 0)
    {
        int tmp = a % 10;
        b = b * 10 + tmp;
        a = a / 10;
    }
    if(n > 1)
        cout << -1;
    else
        cout << b;
    return 0;
}

```

**Bài 24.** Số Palidrom là số đối xứng, nghĩa là đọc từ trái sang phải hay từ phải sang trái ta đều được 1 số. Ví dụ: 11, 121, 1331, .... Nhập 2 số nguyên dương m, n. Liệt kê dãy số vừa là số nguyên tố vừa là số Palidrom trong đoạn [m, n]. Ví dụ:

PALIDROM.INP	PALIDROM.OUT
100 200	101 131 151 181 191

Code tham khảo:

```

#include <iostream>
#include <cmath>
using namespace std;
int palidrom(int n) {
    int x = n;
    int d = 0;
    while (n > 0) {
        d = d*10 + n%10;
        n /= 10;
    }
    if (x == d) return 1;
    else return 0;
}
int nguyenTo(int n) {

```

```

    if (n < 2) return 0;
    for (int i = 2; i <= sqrt(n); i++) {
        if (n % i == 0) return 0;
    }
    return 1;
}
int main () {
    freopen("PALIDROM.INP", "r", stdin);
    freopen("PALIDROM.OUT", "w", stdout);
    long m, n;
    cin >> m >> n;
    for (int i = m; i < n; i++) {
        if (palidrom(i) && nguyenTo(i))
            cout << i << " ";
    }
    return 0;
}

```

## 9. Mảng

Mảng là một loại cấu trúc dữ liệu, nó lưu trữ một tập hợp tuần tự các phần tử cùng kiểu với độ dài cố định. Mảng thường được sử dụng để lưu trữ tập hợp dữ liệu, nhưng nó cũng hữu dụng khi dùng để lưu trữ một tập hợp biến có cùng kiểu.

Mảng được sử dụng để lưu trữ nhiều giá trị trong một biến duy nhất, thay vì khai báo các biến riêng biệt cho từng giá trị.

Để khai báo một mảng, hãy xác định kiểu biến, chỉ định tên của mảng, theo sau là **dấu ngoặc vuông** và chỉ định số phần tử mà nó sẽ lưu trữ:

Cú pháp: `string cars[4];`

Khai báo kiểu danh sách:

```

string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};
int myNum[3] = {10, 20, 30};

```

Truy cập vào mỗi phần tử của mảng: Ta sử dụng chỉ số trực tiếp, cần lưu ý rằng: Trong C++ **các chỉ số của mảng bắt đầu từ số 0**

Ví dụ: `cout << cars[0] // Trả về kết quả là Volvo`

Với mảng dữ liệu, ta thường xuyên sử dụng vòng lặp để đọc, ghi hay xử lý các giá trị của mảng. Ví dụ dưới đây sẽ hiển thị dãy số chẵn, lẻ của mảng.

```

#include <iostream>
using namespace std;
int main() {
    freopen("CHANLE.INP", "r", stdin);
    freopen("CHANLE.OUT", "w", stdout);
    int n; int a[1000];
    cin >> n;
    for (int i=0;i<n;i++) {

```

```

        cin >> a[i];
    }
    // Day so chan:
    for (int i=0;i<n;i++) {
        if (a[i] % 2 == 0) cout << a[i] << " ";
    }
    cout << endl;
    // Day so le:
    for (int i=0;i<n;i++) {
        if (a[i] % 2 != 0) cout << a[i] << " ";
    }
    return 0;
}

```

Để xác định kích thước của mảng được khai báo, ta sử dụng lệnh như ví dụ dưới đây:

```

int arr[] = { 2, 3, 4, 10, 40 };
int x = 10;
int n = sizeof(arr) / sizeof(arr[0]);

```

### Bài tập

**Bài 1.** Đọc số nguyên dương  $n$  và dãy số  $a_1, a_2, \dots, a_n$  từ file DAYS0.INP. Ghi lại vào file DAYS0.OUT. Dòng thứ nhất ghi  $n$ , dòng thứ hai ghi dãy số, mỗi phần tử cách nhau một dấu cách. Ví dụ:

DAYSO.INP	DAYSO.OUT
6	6
11 21 15 17 13 19	11 21 15 17 13 19

Code tham khảo:

```

#include <iostream>
using namespace std;

int main () {
    freopen("DAYSO.INP","r",stdin);
    freopen("DAYSO.OUT","w",stdout);
    int n, a[1000];
    cin >> n;
    // Doc day so:
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    // Ghi day so:
    cout << n << endl;
    for (int i = 0; i < n; i++) {
        cout << a[i] << " ";
    }
}

```

---

```
    return 0;
}
```

**Bài 2.** Sắp xếp với hàm sort trong thư viện algorithm.

Dưới đây là code tham khảo cho bạn về việc lấy dữ liệu từ file SORT.INP, sử dụng hàm sắp xếp **sort** trong thư viện **algorithm** có sẵn để thực hiện việc sắp xếp tăng hoặc sắp xếp giảm và ghi lại vào file SORT.OUT. Ví dụ:

SORT.INP	SORT.OUT
6	11 13 15 17 19 21
11 21 15 17 13 19	21 19 17 15 13 11

Code tham khảo:

```
#include <iostream>
#include <algorithm>
using namespace std;
int main() {
    int n;
    int a[1000];
    freopen("SORT.INP", "r", stdin);
    freopen("SORT.OUT", "w", stdout);
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    // sap xep tang:
    sort(a, a + n);
    for (int i = 0; i < n; i++) {
        cout << a[i] << " ";
    }
    cout << endl;
    // Sap xep giam:
    sort(a, a + n, greater<int>());
    for (int i = 0; i < n; i++) {
        cout << a[i] << " ";
    }
    return 0;
}
```

**Bài 3.** Tính tổng và tính trung bình cộng các phần tử của một dãy số. Yêu cầu đọc số phần tử n và dãy số trong file DAYSO.INP, tính toán ghi vào file DAYSO.OUT vào hai dòng: dòng thứ nhất ghi tổng, dòng thứ hai ghi giá trị trung bình cộng. Ví dụ:

DAYSO.INP	DAYSO.OUT
6	96
11 21 15 17 13 19	16

Code tham khảo:

---

```

#include <iostream>
using namespace std;

int main () {
    freopen("DAYSO.INP","r",stdin);
    freopen("DAYSO.OUT","w",stdout);
    int n, a[1000];
    cin >> n;
    int tong = 0;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
        tong += a[i];
    }
    cout << tong << endl << (float)tong / n;
    return 0;
}

```

**Bài 4.** Đếm xem có bao nhiêu phần tử có giá trị lớn hơn hoặc bằng trung bình cộng của dãy số, liệt kê các phần tử này. Ví dụ:

DAYSO.INP	DAYSO.OUT
6	21 17 19
11 21 15 17 13 19	3 16

Code tham khảo:

```

#include <iostream>
using namespace std;

int main () {
    freopen("DAYSO.INP","r",stdin);
    freopen("DAYSO.OUT","w",stdout);
    int n, a[1000];
    cin >> n;
    int tong = 0;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
        tong += a[i];
    }
    float tb = (float)tong / n;
    int dem = 0;
    for (int i = 0; i < n; i++) {
        if (a[i] >= tb) {
            dem++;
            cout << a[i] << " ";
        }
    }
    cout << endl << dem << " " << tb;
    return 0;
}

```

**Bài 5.** Tìm các vị trí  $i$  (nếu có) chia dãy số thành 2 phần có tổng bằng nhau. Nếu không có ghi ra -1.

Ví dụ:  $n = 6; 1\ 3\ 4\ 2\ 7\ 3 \rightarrow i = 4$  (giải thích:  $1+3+4+2=7+3$ )

Code tham khảo:

```
#include <iostream>
using namespace std;
int main() {
    int a[100], s[100], n, i, p;
    freopen("BEAUTIFULL.INP", "r", stdin);
    freopen("BEAUTIFULL.OUT", "w", stdout);
    cin >> n;
    for (i=0; i<n; i++) cin >> a[i];
    s[0] = a[0];
    for (i=1; i<n; i++) s[i] = s[i-1] + a[i];
    p = 0;
    for (i=0; i < n; i++) {
        if (s[i] == s[n-1] / 2) {
            cout << i+1 << " ";
            p = 1;
        }
    }
    if (p == 0) cout << -1;
    return 0;
}
```

**Bài 6.** Tìm giá trị lớn nhất và giá trị nhỏ nhất của dãy số. Ví dụ:

MINMAX.INP	MINMAX.OUT
6	10
12 15 11 17 19 10	19

Code tham khảo:

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

int main () {
    freopen("MINMAX.INP", "r", stdin);
    freopen("MINMAX.OUT", "w", stdout);
    int n, a[1000];
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    int min = a[0], max = a[0];
    for (int i = 1; i < n; i++) {
        if(min > a[i]) min = a[i];
        if(max < a[i]) max = a[i];
    }
}
```

```

    }
    cout << min << endl << max;
    return 0;
}

```

**Bài 7.** Tìm các vị trí phần tử đạt giá trị nhỏ nhất của dãy số. Ví dụ:

MINMAX.INP	MINMAX.OUT
6 12 10 11 17 10 11	2 5

Code tham khảo:

```

#include <iostream>
using namespace std;

int main () {
    freopen("MIN.INP", "r", stdin);
    freopen("MIN.OUT", "w", stdout);
    int n, a[1000];
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    int min = a[0];
    for (int i = 1; i < n; i++) {
        if(min > a[i]) min = a[i];
    }
    for (int i = 0; i < n; i++) {
        if (a[i] == min) cout << i + 1 << " ";
    }
    return 0;
}

```

**Bài 8.** Tìm giá trị nhỏ nhì (nếu có) của dãy số và đưa ra các vị trí đạt giá trị đó. Nếu không có thì ghi ra -1. Ví dụ:

MIN2.INP	MIN2.OUT
6 13 4 2 7 3	4

Code tham khảo:

```

#include <iostream>
#include <algorithm>
using namespace std;

int main () {
    freopen("MIN2.INP", "r", stdin);
    freopen("MIN2.OUT", "w", stdout);
    int n, a[1000], b[1000];
    cin >> n;

```

```

for (int i = 0; i < n; i++) {
    cin >> a[i];
}
for (int i = 0; i < n; i++) {
    b[i] = a[i];
}
sort(b, b+n);
for (int i = 0; i < n; i++) {
    if (a[i] == b[1])
        cout << i+1;
}
if (b[1] == b[0]) cout << -1;
return 0;
}

```

**Bài 9.** Tìm giá trị lớn thứ k của dãy số. Nhập vào số lượng phần tử n và vị trí lớn thứ k (1 dòng), dòng thứ 2 là dãy số. Yêu cầu: Ghi ra vị trí lớn thứ k trong dãy số. Ví dụ:

MIN2.INP	MIN2.OUT
6 3 27 13 24 19 17 20	6

Code tham khảo:

```

#include <iostream>
#include <algorithm>
using namespace std;

int main () {
    freopen("MINK.INP", "r", stdin);
    freopen("MINK.OUT", "w", stdout);
    int n, k, a[1000], b[1000];
    cin >> n >> k;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    for (int i = 0; i < n; i++) {
        b[i] = a[i];
    }
    sort(b, b+n, greater<int>());
    if (b[k-1] == b[0]) cout << -1;
    else cout << b[k-1];
    return 0;
}

```

**Bài 10.** Tìm dãy 3 phần tử liên tiếp có tổng lớn nhất trong một dãy số có n phần tử ( $n > 3$ ). Đưa tổng ra màn hình. Ví dụ:

MAX3.INP	MAX3.OUT
----------	----------

---

6 13 4 2 7 3	13
-----------------	----

Code tham khảo:

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

int main () {
    freopen("MAX3.INP", "r", stdin);
    freopen("MAX3.OUT", "w", stdout);
    int n, a[1000], s[1000];
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    for (int i = 0; i < n-2; i++) {
        s[i] = a[i] + a[i+1] + a[i+2];
    }
    int m = s[0];
    for (int i = 1; i < n-2; i++) {
        if(m < s[i]) m = s[i];
    }
    cout << m;
    return 0;
}
```

**Bài 11.** Nhập vào giá trị x. Tìm vị trí x (nếu có) trong dãy số có n phần tử và đếm số lượng phần tử có giá trị bằng x. Nếu không có ghi ra -1. Ví dụ:

SEARCH.INP	SEARCH.OUT
6 2 13 3 5 7 3	2 3 6 3

Code tham khảo:

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

int main () {
    freopen("SEARCH.INP", "r", stdin);
    freopen("SEARCH.OUT", "w", stdout);
    int n, x, a[1000];
    cin >> n >> x;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    int tam = a[x-1], dem = 0;
    for (int i = 0; i < n; i++) {
        if(a[i] == tam) {
            cout << i + 1 << " ";
        }
    }
}
```

```

        dem++;
    }
}
if (dem == 0) cout << -1;
else cout << endl << dem;
return 0;
}

```

**Bài 12.** Cho số n, tìm số Fibonacci thứ n. Ví dụ:

SEARCH.INP	SEARCH.OUT
30	832040

Code tham khảo:

```

#include <iostream>
using namespace std;
int64_t Fibonacci(int n)
{
    int64_t a1 = 1, a2 = 1;
    if (n == 1 || n == 2)
        return 1;
    int i = 3; int64_t a;
    while (i <= n)
    {
        a = a1 + a2;
        a1 = a2;
        a2 = a;
        i++;
    }
    return a;
}
int main() {
    freopen("FIBONACCI.INP", "r", stdin);
    freopen("FIBONACCI.OUT", "w", stdout);
    int n;
    cin >> n;
    cout << Fibonacci(n);
    cout << endl;
    for (int i=1; i <= n; i++) {
        cout << Fibonacci(i) << " ";
    }
    return 0;
}

```

**Bài 13.** Liệt kê các dãy con liên tiếp không giảm (có nhiều hơn 1 phần tử) của dãy ban đầu, mỗi dãy trên 1 dòng. Ví dụ:

SEARCH.INP	SEARCH.OUT
6	3 4 6
3 4 6 2 7 3	2 7

Code tham khảo:

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

int main () {
    freopen("KGIAM.INP","r",stdin);
    freopen("KGIAM.OUT","w",stdout);
    int n, a[1000];
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    for (int i = 0; i < n; i++) {
        int dem = 0;
        while (a[i] < a[i+1]) {
            if (dem == 0)
                cout << a[i] << " " << a[i+1] << " ";
            else cout << a[i+1];
            i++; dem++;
        }
        cout << endl;
    }
    return 0;
}
```

**Bài 14.** Tìm số cặp  $(i, j)$  sao cho  $a_i + a_j = k$ . Nếu không có ghi ra -1. Ví dụ:

Ví dụ:  $n = 6$ ; dãy 1 3 4 2 7 3;  $k = 4 \rightarrow res = 2$

SEARCH.INP	SEARCH.OUT
6 4	2
3 5 9 4 6 2	

Code tham khảo:

```
#include <iostream>
using namespace std;
int main () {
    freopen("SEARCH.INP","r",stdin);
    freopen("SEARCH.OUT","w",stdout);
    int n, k, a[1000];
    cin >> n >> k;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    int dem = 0;
    for(int i=0; i<n-1; i++) {
        for(int j=i+1;j<n;j++) {
            if(a[i] + a[j] == k) dem++;
        }
    }
}
```

```

    if(dem==0) cout << -1;
    else cout << dem;
    return 0;
}

```

**Bài 15.** Tạo mảng b lưu các số chẵn, mảng c lưu các số lẻ từ mảng a. Ví dụ:  
 $n = 6$ ; dãy 1 3 4 2 7 3;  $k = 4 \rightarrow res = 2$

CHANLE.INP	CHANLE.OUT
6	3 5 1
3 5 4 6 1 8	4 6 8

Code tham khảo:

```

#include <iostream>
using namespace std;
int main() {
    freopen("CHANLE.INP", "r", stdin);
    freopen("CHANLE.OUT", "w", stdout);
    int n; int a[1000];
    cin >> n;
    for (int i=0; i<n; i++) {
        cin >> a[i];
    }
    // Day so chan:
    for (int i=0; i<n; i++) {
        if (a[i] % 2 == 0) cout << a[i] << " ";
    }
    cout << endl;
    // Day so le:
    for (int i=0; i<n; i++) {
        if (a[i] % 2 != 0) cout << a[i] << " ";
    }
    return 0;
}

```

**Bài 16.** Cho dãy số a (m phần tử), dãy số b (n phần tử). Tìm giá trị của  $a_i + b_j$  nhỏ nhất. Đưa giá trị nhỏ nhất đó ra màn hình. Ví dụ:

MAXARR.INP	MAXARR.OUT
6 4	16
3 4 6 2 7 3	
3 9 5 7	

Code tham khảo:

```

#include <iostream>
using namespace std;

int main() {
    freopen("MAXARR.INP", "r", stdin);
    freopen("MAXARR.OUT", "w", stdout);
    int m, n, a[1000], b[1000];

```

```

cin >> m >> n;
for (int i=0; i<m; i++) {
    cin >> a[i];
}
for(int j=0; j< n; j++) {
    cin >> b[j];
}
int d = a[0] + b[0];
for (int i=0; i<m; i++) {
    for(int j=0; j<n; j++) {
        if (d < a[i] + b[j])
            d = a[i] + b[j];
    }
}
cout << d;
return 0;
}

```

**Bài 17.** Xóa phần tử thứ k của mảng a có n phần tử. Đưa ra số phần tử còn lại và mảng a sau khi xóa. Ví dụ:

DELARR.INP	DELARR.OUT
6 4	5
5 7 3 4 6 9	5 7 3 6 9

Code tham khảo:

```

#include <iostream>
using namespace std;

void DelK(int a[],int &n, int k) {
    for (int i = k; i < n; i++) {
        a[i-1] = a[i];
    }
    n--;
}

int main() {
    freopen("DELARR.INP","r",stdin);
    freopen("DELARR.OUT","w",stdout);
    int n, k; int a[100];
    cin >> n >> k;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    DelK(a,n,k);
    cout << n << endl;
    for (int i = 0; i < n; i++) {
        cout << a[i] << " ";
    }
}

```

---

**Bài 18.** Xóa tất cả các phần tử chia hết cho k trong mảng a có n phần tử. Đưa ra số phần tử còn lại và mảng a sau khi xóa. Ví dụ:

MODARR.INP	MODARR.OUT
6 2	3
3 6 5 4 7 2	3 5 7

Code tham khảo:

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

void DelK(int a[],int &n, int k) {
    for (int i = k+1; i < n; i++) {
        a[i-1] = a[i];
    }
    n--;
}

void DelArr(int a[], int &n, int k) {
    for (int i = 0; i < n; i++) {
        if(a[i] % k == 0) {
            DelK(a, n, i);
        }
    }
}

int main() {
    freopen("MODARR.INP","r",stdin);
    freopen("MODARR.OUT","w",stdout);
    int n, k; int a[100];
    cin >> n >> k;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    DelArr(a,n,k);
    cout << n << endl;
    for (int i = 0; i < n; i++) {
        cout << a[i] << " ";
    }
}
```

**Bài 19.** Xóa các phần tử giống nhau trong dãy số, chỉ giữ lại những phần tử đại diện mỗi số. Sau khi xóa, đưa dãy số ra màn hình. Ví dụ:

XOATRUNG.INP	XOATRUNG.OUT
6	4
3 6 5 3 7 3	3 6 5 7

Code tham khảo:

```
#include <iostream>
using namespace std;
// Xoa phan tu cua mang tai vi tri bat ky
```

```

void Xoa(int a[],int &n, int vitri) {
    for (int i = vitri + 1; i < n; i++) {
        a[i-1] = a[i];
    }
    n--;
}
// Xoa cac phan tu trung nhau
void Xoa_Trung(int a[], int &n) {
    for (int i = 0; i < n-1; i++) {
        for (int j = i+1; j < n; j++) {
            if (a[i]==a[j]) {
                Xoa(a,n,j);
                i--;
            }
        }
    }
}
int main() {
    freopen("XOATRUNG.INP","r",stdin);
    freopen("XOATRUNG.OUT","w",stdout);
    int n; int a[100];
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    Xoa_Trung(a,n);
    cout << n << endl;
    for (int i = 0; i < n; i++) {
        cout << a[i] << " ";
    }
}

```

**Bài 20.** Chèn x vào phần tử thứ k của dãy số a có n phần tử. Đưa ra số phần tử và dãy số a sau khi chèn. Ví dụ:

INSERTARR.INP	INSERTARR.OUT
6 4 8	7
4 6 5 9 3 7	4 6 5 8 9 3 7

Code tham khảo:

```

#include <iostream>
using namespace std;

void InserArr(int a[],int &n, int k, int x) {
    n++;
    for (int i = n-1; i >= k-1; i--) {
        a[i+1] = a[i];
    }
    a[k-1] = x;
}

```

---

```

}
int main() {
    freopen("INSERTARR.INP", "r", stdin);
    freopen("INSERTARR.OUT", "w", stdout);
    int n, k, x; int a[100];
    cin >> n >> k >> x;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    InserArr(a, n, k, x);
    cout << n << endl;
    for (int i = 0; i < n; i++) {
        cout << a[i] << " ";
    }
}

```

**Bài 21.** Đếm số lần xuất hiện của mỗi phần tử trong dãy số. Ví dụ:

XUATHIEN.INP	XUATHIEN.OUT
6	3 2
3 5 3 6 5 7	5 2
	6 1
	7 1

Code tham khảo:

```

#include <iostream>
using namespace std;

int main() {
    freopen("XUATHIEN.INP", "r", stdin);
    freopen("XUATHIEN.OUT", "w", stdout);
    int n, a[1000], b[1000];
    cin >> n;
    for(int i=0; i<n; i++) {
        cin >> a[i];
        b[i] = 1;
    }
    for(int i=0; i<n; i++) {
        int d = 1;
        if(b[i]) {
            b[i] = 0;
            for (int j=i+1; j<n; j++)
                if(a[j] == a[i]) {
                    d++;
                    b[j] = 0;
                }
            cout << a[i] << " " << d << endl;
        }
    }
}

```

---

```

    return 0;
}

```

**Bài 22.** Đảo ngược dãy số. Đưa dãy số đảo ngược ra màn hình. Ví dụ:

DAODS.INP	DAODS.OUT
6	6
3 5 3 6 5 7	7 5 6 3 5 3

Code tham khảo:

```

#include <iostream>
using namespace std;

int main() {
    freopen("DAODS.INP", "r", stdin);
    freopen("DAODS.OUT", "w", stdout);
    int n, a[1000];
    cin >> n;
    for(int i=0; i < n; i++) {
        cin >> a[i];
    }
    cout << n << endl;
    for(int i=n-1; i >= 0; i--) {
        cout << a[i] << " ";
    }
    return 0;
}

```

**Bài 23.** Tìm dãy con liên tiếp không giảm dài nhất của 1 dãy số. Yêu cầu: in ra số phần tử và dãy con liên tiếp dài nhất. Ví dụ:

DAYKT.INP	DAYKT.OUT
6	3
3 2 5 7 4 6	2 5 7

Code tham khảo:

```

#include <iostream>
using namespace std;

int main() {
    freopen("DAYKT.INP", "r", stdin);
    freopen("DAYKT.OUT", "w", stdout);
    int n, a[1000], b[1000];
    cin >> n;
    for(int i=0; i < n; i++) {
        cin >> a[i];
        b[i] = 1;
    }
    for(int i=n-1; i>0; i--) {
        if(a[i] >= a[i-1]) {

```

```

        b[i-1] += b[i];
    }
}
int m = b[0];
for (int i=1; i<n; i++) {
    if(m < b[i]) m = b[i];
}
cout << m << endl;
int vt = 0;
for (int i = 0; i < n; i++) {
    if(b[i] == m) vt = i;
}
for (int i = vt; i < vt + m; i++) {
    cout << a[i] << " ";
}
return 0;
}

```

**Bài 24.** Tìm dãy con liên tiếp không giảm có tổng lớn nhất. Kết quả ghi ra tổng và liệt kê dãy các phần tử có tổng lớn nhất. Ví dụ:

DAYKT.INP	DAYKT.OUT
6	14
3 2 5 7 4 6	2 5 7

Code tham khảo:

```

#include <iostream>
using namespace std;

int main() {
    freopen("SUMDS.INP", "r", stdin);
    freopen("SUMDS.OUT", "w", stdout);
    int n, a[1000], b[1000], c[1000], d[1000];
    cin >> n;
    for(int i=0; i < n; i++) {
        cin >> a[i];
        b[i] = 1;
    }
    for(int i=n-1; i>0; i--) {
        if(a[i] >= a[i-1]) {
            b[i-1] += b[i];
        }
    }
    int i = 0;
    int k = 0;
    int m = 0;
    int index = 0;
    int sl = 0;
    while (i < n) {

```

---

```

        if(b[i] == 1) {
            c[k] = a[i];
            i++;
            if (m < c[k]) {
                m = c[k];
                index = i;
                sl = b[i];
            }
        }
        else {
            c[k] = 0;
            for(int j=i; j < i+b[i]; j++) {
                c[k] += a[j];
            }
            if (m < c[k]) {
                m = c[k];
                index = i;
                sl = b[i] + 1;
            }
            i += b[i];
        }
        k++;
    }
    for (i=0; i < k; i++) {
        if(m < c[i]) m = c[i];
    }
    cout << m << " " << endl;
    for (int i = index; i < sl; i++)
        cout << a[i] << " ";
    return 0;
}

```

**Bài 25.** Tìm dãy con bằng nhau liên tiếp trong 1 dãy số. Yêu cầu: liệt kê các phần tử của dãy con bằng nhau. Ví dụ:

BANGNHAU.INP	BANGNHAU.OUT
6	2 2 2
2 2 2 7 4 4	4 4

Code tham khảo:

```

#include <iostream>
using namespace std;

int main() {
    freopen("BANGNHAU.INP", "r", stdin);
    freopen("BANGNHAU.OUT", "w", stdout);
    int n, a[1000], b[1000], c[1000], d[1000];
    cin >> n;
    for(int i=0; i < n; i++) {

```

---

```

        cin >> a[i];
        b[i] = 1;
    }
    for(int i=n-1; i>0; i--) {
        if(a[i] == a[i-1]) {
            b[i-1] += b[i];
        }
    }
    int i = 0;
    while (i < n) {
        if(b[i] != 1) {
            for(int j=i; j < i+b[i]; j++) {
                cout << a[j] << " ";
            }
            cout << endl;
            i += b[i];
        } else i++;
    }
    return 0;
}

```

**Bài 26.** Tìm dãy con bằng nhau liên tiếp dài nhất trong 1 dãy số. Yêu cầu: ghi ra tổng số phần tử và liệt kê các phần tử của dãy con bằng nhau. Ví dụ:

MAXBN.INP	MAXBN.OUT
10	4
2 2 1 1 1 1 3 4 4 4	1 1 1 1

Code tham khảo:

```

#include <iostream>
using namespace std;

int main() {
    freopen("MAXBN.INP", "r", stdin);
    freopen("MAXBN.OUT", "w", stdout);
    int n, a[1000], b[1000];
    cin >> n;
    for(int i=0; i < n; i++) {
        cin >> a[i];
        b[i] = 1;
    }
    for(int i=n-1; i>0; i--) {
        if(a[i] == a[i-1]) {
            b[i-1] += b[i];
        }
    }
    int m = b[0];
    for (int i=1; i<n; i++) {

```

```

        if(m < b[i]) m = b[i];
    }
    cout << m << endl;
    int vt = 0;
    for (int i = 0; i < n; i++) {
        if(b[i] == m) vt = i;
    }
    for (int i = vt; i < vt + m; i++) {
        cout << a[i] << " ";
    }
    return 0;
}

```

**Bài 27.** Liệt kê các số chính phương có trong dãy số a. Nếu không có ghi số -1. Ví dụ:

SOCP.INP	SOCP.OUT
10	4 9 16 36
3 4 7 5 9 3 16 21 36 8	

Code tham khảo:

```

#include <iostream>
#include <cmath>
using namespace std;

int main() {
    freopen("SOCP.INP", "r", stdin);
    freopen("SOCP.OUT", "w", stdout);
    int n, a[1000];
    cin >> n;
    for(int i=0; i < n; i++) {
        cin >> a[i];
    }
    int p = 0;
    for(int i = 1; i < n; i++)
        if(sqrt(a[i]) == trunc(sqrt(a[i]))) {
            cout << a[i] << " ";
            p = 1;
        }
    if(p == 0) cout << -1;
    return 0;
}

```

**Bài 28.** Liệt kê các số nguyên tố theo thứ tự từ nhỏ đến lớn trong dãy số a. Ví dụ:

NTO.INP	NTO.OUT
10	2 3 7 29 31 37
3 4 7 2 9 31 16 29 37 9	

---

Code tham khảo:

```
#include <iostream>
#include <algorithm>
#define nmax 1000007
using namespace std;

int n,a[nmax],f[nmax],c[nmax];
void sangNto(int u) {
    for(int i = 1;i <= u;i++)
        f[i] = 1;
    f[1] = 0;
    for(int i = 2;i <= u/2;i++)
        if(f[i] == 1)
            for(int j=i;j<=u/2;j++)
                f[i*j] = 0;
}
int main() {
    freopen("NTO.INP","r",stdin);
    freopen("NTO.OUT","w",stdout);
    int h = 0;
    cin >> n;
    int rmax = -999999999;
    for(int i = 1;i <= n;i++) {
        cin >> a[i];
        rmax = max(rmax,a[i]);
    }
    sangNto(rmax);
    for(int i = 1;i <= n;i++)
        if(f[a[i]] == 1)
            c[++h] = a[i];
    sort(c + 1, c + h + 1);
    for(int i = 1;i <= h;i++)
        cout << c[i]<<" ";
    if(h == 0) cout << -1;
    return 0;
}
```

## 10. Xâu ký tự (string)

Trong C++, bạn có thể tạo ra một đối tượng string để lưu trữ chuỗi ký tự. Không giống mảng ký tự, đối tượng string không có kích thước cố định và có thể mở rộng nếu cần.

Đối tượng string được tạo bởi lớp string trong thư viện `#include <cstring>`

Cú pháp: `string <tên_đối_tượng>`

Thư viện `<string>` rất nhiều hàm cho phép xử lý chuỗi. Ví dụ: tìm chiều dài chuỗi, so sánh hai chuỗi, tìm kiếm/rút trích chuỗi con, nối chuỗi,...

---

## Một số hàm thường sử dụng với chuỗi ký tự s:

`s.length()`: Trả về kích thước của chuỗi ký tự s

`s.erase(i, n)`: Xóa n ký tự từ ký tự thứ i

`s.find(s1)`: Tìm kiếm vị trí bắt đầu của chuỗi ký tự s1 có trong chuỗi ký tự s

Một điểm cần lưu ý khi làm việc với chuỗi ký tự đó là chuẩn hóa chuỗi. Chúng ta cần đổi các ký tự in thường thành ký tự IN HOA và ngược lại. Để thực hiện công việc này ta sử dụng các lệnh dưới đây:

- Đổi từ chữ in thường sang IN HOA:

```
if(s[i] >= 'a' && s[i] <= 'z')
    s[i] -= 32;
```

- Đổi từ chữ IN HOA sang in thường:

```
if(s[i] >= 'A' && s[i] <= 'Z')
    s[i] += 32;
```

**Lưu ý:** Khi làm việc với các chuỗi ký tự, ta có hai loại ký tự đó là các ký tự số và các ký tự chữ. Các ký tự số gồm 10 ký tự từ **0 đến 9**, còn các ký tự chữ từ **a đến z**. Các bài toán về chuỗi ký tự chủ yếu tập trung vào một số hàm chúng tôi đã nêu ra ở đây. Các bạn sẽ hiểu rõ hơn thông qua việc giải quyết một số bài toán về chuỗi ký tự trong tài liệu này.

## Bài tập

**Bài 1.** Cho một chuỗi ký tự s. Yêu cầu: Ghi lại, tính độ dài chuỗi và sắp xếp lại các ký tự trong chuỗi s. Ví dụ:

STRING.INP	STRING.OUT
ABEHGCDVIM	ABEHGCDVIM
	10
	ABCDEFGHIIMV

Code tham khảo:

```
#include <iostream>
#include <cstring>
using namespace std;
int main() {
    freopen("STRING.INP", "r", stdin);
    freopen("STRING.OUT", "w", stdout);
    string s;
    char temp;
    cin >> s;
    cout << s << endl;
    int m = s.length();
    cout << s.length() << endl;
    for(int i=1; i<=m-2; i++)
        for(int j=i+1; j<=m-1; j++)
```

```

        if(s[i] > s[j]) {
            temp = s[i];
            s[i] = s[j];
            s[j] = temp;
        }
    for(int i=0;i<=m-1;i++) cout<<s[i];
    return 0;
}

```

**Bài 2.** Cho xâu ký tự s. Đếm ký tự a và xóa ký tự này trong xâu s, hiển thị xâu s sau khi xóa ký tự a. Ví dụ:

STRING.INP	STRING.OUT
abaebcda	3 bebcd

Code tham khảo:

```

#include <iostream>
#include <cstring>
using namespace std;

int main() {
    freopen("COUTS.INP", "r", stdin);
    freopen("COUTS.OUT", "w", stdout);
    string s;
    cin>>s;
    int dem = 0;
    for(int i=0; i<=s.length()-1; i++)
        if(s[i] == 'a') dem++;
    int i = 0;
    int k = 1;
    while(i <= s.length()-1) {
        if(s[i] == 'a') s.erase(i,1);
        else i++;
    }
    cout << dem << endl;
    for(int i=0; i<=s.length()-1; i++)
        cout<<s[i];
    return 0;
}

```

**Bài 3.** Cho xâu ký tự s, các số nguyên dương k và n. Yêu cầu:

- Xóa n ký tự của xâu s từ vị trí thứ k
- Xóa k vị trí đầu tiên
- Xóa n ký tự cuối cùng của xâu s.

Ví dụ:

---

DELS.INP	DELS.OUT
sfasecwbdgmxv 2 5	

Code tham khảo:

```
#include <iostream>
#include <cstring>
using namespace std;
int main() {
    freopen("DELS.INP", "r", stdin);
    freopen("DELS.OUT", "w", stdout);
    string s;
    int k, n;
    cin >> s;
    cin >> k >> n;
    s.erase(k, n);
    cout << s << endl;
    s.erase(0, k);
    cout << s << endl;
    s.erase(s.length()-n, n);
    cout << s;
    return 0;
}
```

**Bài 4.** Cho hai chuỗi ký tự s1 và s2. Xóa tất cả các ký tự của chuỗi s1 trong chuỗi s2. Ví dụ:

DELA.INP	DELA.OUT
ab sfabsabcwbdgbab	sfscwbdgb

Code tham khảo:

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

int main() {
    freopen("DELA.INP", "r", stdin);
    freopen("DELA.OUT", "w", stdout);
    string s1, s2;
    cin >> s1 >> s2;
    while(s2.find(s1) >= 0) {
        int p = s2.find(s1);
        if(p >= 0)
            s2.erase(p, s1.length());
        else break;
    }
    cout << s2;
```

---

```

    return 0;
}

```

**Bài 5.** Cho chuỗi ký tự s, thực hiện chuẩn hóa chuỗi s bằng cách xóa tất cả các dấu cách thừa, chỉ giữ lại một dấu cách. Ví dụ:

CHUANS.INP	CHUANS.OUT
Do Trung Thanh	Do Trung Thanh

Code tham khảo:

```

#include <iostream>
#include <cstring>
using namespace std;
string s;
void chuanXau() {
    while(s.find(" ") >= 0) {
        int p = s.find(" ");
        if(p >= 0)
            s.erase(p,1);
        else break;
    }
    if(s[0] == ' ')
        s.erase(0,1);
    if(s[s.length()] == ' ')
        s.erase(s.length(),1);
}
int main() {
    freopen("CHUANA.INP","r",stdin);
    freopen("CHUANA.OUT","w",stdout);
    getline(cin, s);
    chuanXau();
    cout << s;
    return 0;
}

```

**Bài 6.** Cho chuỗi ký tự s. Chuẩn hóa chuỗi bằng cách:

- Xóa tất cả các ký tự thừa (2 dấu cách trở lên để lại 1 dấu cách)
- Định dạng chữ cái đầu tiên của chuỗi thành chữ IN HOA.

Ví dụ:

CHUANB.INP	CHUANB.OUT
do Trung Thanh	Do Trung Thanh

Code tham khảo:

```

#include <iostream>
#include <cstring>
using namespace std;
string s;

```

```

void chuanXau() {
    while(s.find(" ") >= 0) {
        int p = s.find(" ");
        if(p >= 0)
            s.erase(p,1);
        else break;
    }
    if(s[0] == ' ')
        s.erase(0,1);
    if(s[s.length()] == ' ')
        s.erase(s.length(),1);
}
char chuHoa(char x) {
    if(x >= 'a' && x <= 'z')
        x -= 32;
    return x;
}
int main() {
    freopen("CHUANB.INP","r",stdin);
    freopen("CHUANB.OUT","w",stdout);
    getline(cin, s);
    chuanXau();
    s[0] = chuHoa(s[0]);
    cout << s;
    return 0;
}

```

**Bài 7.** Cho xâu ký tự s. Chuẩn hóa xâu bằng cách:

- Xóa tất cả các ký tự thừa (2 dấu cách trở lên để lại 1 dấu cách)
- Định dạng toàn bộ xâu ký tự s thành chữ IN HOA.

Ví dụ:

CHUANC.INP	CHUANC.OUT
do trung thanh	DO TRUNG THANH

Code tham khảo:

```

#include <iostream>
#include <cstring>
using namespace std;
string s;
void chuanXau() {
    while(s.find(" ") >= 0) {
        int p = s.find(" ");
        if(p >= 0)
            s.erase(p,1);
        else break;
    }
}

```

```

        if(s[0] == ' ')
            s.erase(0,1);
        if(s[s.length()] == ' ')
            s.erase(s.length(),1);
    }
    char chuHoa(char x) {
        if(x >= 'a' && x <= 'z')
            x -= 32;
        return x;
    }
    int main() {
        freopen("CHUANC.INP","r",stdin);
        freopen("CHUANC.OUT","w",stdout);
        getline(cin, s);
        chuanXau();
        for (int i=0; i < s.length(); i++)
            s[i] = chuHoa(s[i]);
        cout << s;
        return 0;
    }

```

**Bài 8.** Cho xâu ký tự s. Chuẩn hóa xâu bằng cách:

- Xóa tất cả các ký tự thừa (2 dấu cách trở lên để lại 1 dấu cách)
- Định dạng các ký tự đầu câu thành chữ IN HOA.

Ví dụ:

CHUANC.INP	CHUANC.OUT
dO truNg tHanh	Do Trung Thanh

Code tham khảo:

```

#include <iostream>
#include <cstring>
using namespace std;
string s;
void chuanXau() {
    while(s.find(" ") >= 0) {
        int p = s.find(" ");
        if(p >= 0)
            s.erase(p,1);
        else break;
    }
    if(s[0] == ' ')
        s.erase(0,1);
    if(s[s.length()] == ' ')
        s.erase(s.length(),1);
}
void chuHoa() {

```

```

for(int i=1; i<=s.length()-1; i++)
    if(s[i] >= 'A' && s[i] <= 'Z')
        s[i] += 32;
if(s[0] >= 'a' && s[0] <= 'z')
    s[0] -= 32;
for(int i=1; i<=s.length()-1; i++)
    if(s[i] == ' ' && s[i+1] >= 'a' && s[i+1] <= 'z')
        s[i+1] -= 32;
}
int main() {
    freopen("CHUAND.INP", "r", stdin);
    freopen("CHUAND.OUT", "w", stdout);
    getline(cin, s);
    chuanXau();
    chuHoa();
    cout << s;
    return 0;
}

```

**Bài 9.** Cho hai chuỗi ký tự s1, s2. Hãy đếm số lần xuất hiện của chuỗi s1 trong chuỗi s2 đồng thời xóa các chuỗi s1 có trong s2. Ví dụ:

DEMKT.INP	DEMKT.OUT
ab	3
abxytabuvrbabxy	xytuvrbxy

Code tham khảo:

```

#include <iostream>
#include <cstring>
using namespace std;
int main() {
    freopen("DEMKT.INP", "r", stdin);
    freopen("DEMKT.OUT", "w", stdout);
    string s1, s2;
    getline(cin, s1);
    getline(cin, s2);
    int dem = 0;
    while(s1.find(s2) >= 0)
    {
        int p = s2.find(s1);
        if(p == -1) break;
        dem++;
        s2.erase(p, s1.length());
    }
    cout << dem << endl << s2;
    return 0;
}

```

**Bài 10.** Đếm số lần xuất hiện của mỗi ký tự trong chuỗi s. Sắp xếp các ký tự

---

và số lần xuất hiện theo thứ tự tăng. Ví dụ:

DEMKT.INP	DEMKT.OUT
abceacbde	a 2 b 2 c 2 d 1 e 2

Code tham khảo:

```
#include <iostream>
#include <cstring>
using namespace std;
int main() {
    freopen("DEMKT.INP", "r", stdin);
    freopen("DEMKT.OUT", "w", stdout);
    string s;
    getline(cin, s);
    for(int i=0; i<=s.length()-2; i++)
        for(int j=i+1; j<=s.length()-1; j++)
            if(s[i] >= s[j]) {
                int temp = s[i];
                s[i] = s[j];
                s[j] = temp;
            }
    int dem = 1;
    for(int i=0; i<=s.length()-1; i++)
        if(s[i] == s[i+1]) dem++;
        else {
            cout<<s[i]<<" " <<dem<<endl;
            dem = 1;
        }
    return 0;
}
```

**Bài 11.** Chuẩn hóa xâu s nhập vào. Yêu cầu:

- Xóa tất cả dấu cách thừa
- Thay các ký tự đầu mỗi từ bằng chữ IN HOA
- Tách ra: Họ và tên đệm trên 1 dòng, tên trên một dòng.

Ví dụ:

CHUANE.INP	CHUANE.OUT
dO truNg tHanh	Do Trung Thanh

Code tham khảo:

```
#include <iostream>
```

---

```

#include <cstring>
using namespace std;
string s;
void chuanXau() {
    while(s.find(" ") >= 0) {
        int p = s.find(" ");
        if(p >= 0)
            s.erase(p,1);
        else break;
    }
    if(s[0] == ' ')
        s.erase(0,1);
    if(s[s.length()-1] == ' ')
        s.erase(s.length(),1);
}
void chuHoa() {
    for(int i=1; i<=s.length()-1; i++)
        if(s[i] >= 'A' && s[i] <= 'Z')
            s[i] += 32;
    if(s[0] >= 'a' && s[0] <= 'z')
        s[0] -= 32;
    for(int i=1; i<=s.length()-1; i++)
        if(s[i] == ' ' && s[i+1] >= 'a' && s[i+1] <= 'z')
            s[i+1] -= 32;
}

int main() {
    freopen("CHUANE.INP", "r", stdin);
    freopen("CHUANE.OUT", "w", stdout);
    getline(cin, s);
    chuanXau();
    chuHoa();
    string s2 = "";
    int i = s.length()-1;
    while(s[i] != ' ') {
        s2 = s[i] + s2;
        s.erase(i,1);
        if(s[i] == ' ') break;
        i--;
    }
    cout << s << endl << s2;
    return 0;
}

```

**Bài 12.** Tách riêng các số trong chuỗi. Ví dụ:

TACH.INP	TACH.OUT
dggfdgf15đff24gfh57uy	15 24 57

Code tham khảo:

```
#include <iostream>
#include <cstring>
using namespace std;
string s;
int main() {
    freopen("TACH.INP", "r", stdin);
    freopen("TACH.OUT", "w", stdout);
    getline(cin, s);
    string p = "";
    for(int i=0; i<=s.length(); i++)
        if(s[i] >= '0' && s[i] <= '9')
            p = p + s[i];
    else
        if(p != "") {
            int pp = stoi(p);
            cout << pp << " ";
            p="";
        }
}
```

---