

## CHUYÊN ĐỀ 1. BÀI TOÁN - THUẬT TOÁN

### Bài 1. NGÔN NGỮ LẬP TRÌNH

#### 1- Một số khái niệm cơ bản:

Trước hết dạng chương trình duy nhất mà máy tính có thể thực thi trực tiếp là **ngôn ngữ máy** hay **mã máy**. Nó có dạng dãy các số nhị phân, thường được ghép thành nhóm **1byte** (8 bit). Để có được bộ mã này, ngày nay người ta dùng **ngôn ngữ lập trình** để viết ra chương trình ở dạng văn bản và dùng **trình dịch** để chuyển sang mã máy.

- **Hợp ngữ** (assembly languages) ra đời, là "ngôn ngữ lập trình thế hệ 2". Lập trình thuận lợi hơn, khi dịch có thể liên kết với thư viện chương trình con ở cả dạng macro (đoạn chưa dịch) và lần mã đã dịch. Hợp ngữ hiện được dùng là ngôn ngữ bậc thấp (low-level programming languages) để tinh chỉnh ngôn ngữ bậc cao thực hiện truy nhập trực tiếp phần cứng, thường dùng trong việc lập trình hệ thống, tạo các hiệu ứng đặc biệt cho chương trình.

- Ngôn ngữ bậc cao (high-level programming languages) hay "*ngôn ngữ lập trình thế hệ 3*" ra đời vào những năm 1950. Đây là các ngôn ngữ hình thức, dùng trong lập trình máy điện toán và không lệ thuộc vào các thế hệ máy tính cụ thể nào. Nó giải phóng người lập trình ứng dụng làm việc trong hệ điều hành xác định mà không phải quan tâm đến phần cứng cụ thể. Các ngôn ngữ lập trình bậc cao được phát triển liên tục với các dạng và biến thể mới, theo bước phát triển của kỹ thuật điện toán.

Theo định nghĩa ở trên thì một ngôn ngữ lập trình phải thỏa mãn được hai điều kiện cơ bản sau:

1. Dễ hiểu và dễ sử dụng đối với người lập trình, để có thể dùng để giải quyết nhiều bài toán khác nhau.
2. Miêu tả một cách đầy đủ và rõ ràng các tiến trình (*process*), để chạy được trên các hệ máy tính khác nhau.

Hiện nay có nhiều ngôn ngữ lập trình bậc cao như: Ngôn ngữ lập trình Pascal, C/C++, Visual Basic, Delphi, Python, Java, pHp, Asp, .... Tùy theo mục đích của chương trình mà ta chọn ngôn ngữ cho phù hợp.

Một tập hợp các chỉ thị được biểu thị qua ngôn ngữ lập trình nhằm mục đích thực hiện các thao tác của máy tính được gọi là một **chương trình**, thường được gọi là **phần mềm máy tính**.

Ví dụ: chương trình Microsoft Word là một cách gọi chung chung; cách gọi *phần mềm Microsoft Word* chỉ rõ hơn nó là một chương trình ứng dụng.

- **Lập trình** dùng để chỉ quá trình con người tạo ra chương trình máy tính thông qua ngôn ngữ lập trình. Người ta còn gọi đó là **quá trình mã hoá** thông tin tự nhiên thành ngôn ngữ máy (viết mã nguồn).

Thao tác chuyển đổi từ mã nguồn thành chuỗi các chỉ thị cho máy tính thực hiện gọi là **biên dịch** (dịch). Nếu quá trình dịch diễn ra đồng thời với quá trình thực thi, ta gọi đó là **thông dịch**; nếu quá trình chạy chương trình diễn ra sau khi dịch ta gọi đó là **biên dịch**. Phần mềm dịch tương ứng được gọi là phần mềm thông dịch và phần mềm biên dịch.

Điểm khác nhau giữa thông dịch và biên dịch là: trình thông dịch dịch từng câu lệnh theo yêu cầu thực thi và chương trình đích vừa tạo ra sẽ không được lưu lại; trong khi đó, trình biên

dịch sẽ dịch toàn bộ chương trình, cho ra chương trình đích được lưu lại trong máy tính rồi mới thực hiện **chương trình**.

Để đạt được yêu cầu về độ chính xác và tính hiệu quả, mã viết ra nhiều khi khó đọc ngay cả với chính người viết ra mã đó, chưa kể tới người khác. Chính vì lý do đó, mọi tài liệu, hướng dẫn lập trình đều khuyên nên thêm các chú giải vào mã nguồn trong quá trình viết mã. Các chú giải giúp người khác rất nhiều trong việc đọc hiểu mã nguồn; đối với chương trình phức tạp, chú giải là thành phần vô cùng quan trọng trong mã nguồn.

## **2. Đặc điểm chung của ngôn ngữ lập trình:**

Mỗi ngôn ngữ lập trình có thể được xem như là một tập hợp của các chi tiết kỹ thuật chú trọng đến cú pháp, từ vựng, và ý nghĩa của ngôn ngữ.

Những chi tiết kỹ thuật này thường bao gồm:

- Dữ liệu và cấu trúc dữ liệu
- Câu lệnh và dòng điều khiển
- Các tên và các tham số
- Các cơ chế tham khảo và sự tái sử dụng

Đối với các ngôn ngữ phổ biến hoặc có lịch sử lâu dài, người ta thường tổ chức các hội thảo chuẩn hoá nhằm tạo ra và công bố các tiêu chuẩn chính thức cho ngôn ngữ đó, cũng như thảo luận về việc mở rộng, bổ sung cho các tiêu chuẩn trước đó.

## **CHUYÊN ĐỀ 1. BÀI TOÁN - THUẬT TOÁN**

### **Bài 2. THUẬT TOÁN**

**1. Bài toán:** Trong phạm vi Tin học, ta có thể quan niệm một bài toán là một công việc, một vấn đề nào đó cần giải quyết với sự trợ giúp của máy tính. Để giải bài toán trên máy tính cần thực hiện năm công việc chính sau:

- Xác định bài toán: Xác định dữ liệu vào, dữ liệu đã cho trước (input) và tập kết quả ra ta cần nhận được (output).

Ví dụ: Bài toán tìm bội chung nhỏ nhất của hai số nguyên dương  $a, b$ .

+ input:  $a, b$  ( $a, b$  hai số nguyên dương)

+ output: BCNN( $a, b$ ).

- Thiết kế hoặc lựa chọn thuật toán: Tìm cách xác lập mối quan hệ giữa input và output từ đó ta thiết lập một dãy các thao tác để từ input bài toán ta được output cần tìm. Dãy các thao tác đó gọi là thuật toán.

Một thuật toán chỉ giải được một bài toán nhưng một bài toán có thể có nhiều thuật toán để giải bài toán đó. Việc lựa chọn thuật toán tốt dựa vào nhiều tiêu chí như: thời gian, bộ nhớ (không gian), cài đặt. Trong đó thời gian chạy chương trình là tiêu chí quan trọng nhất.

- Viết chương trình: Bao gồm công việc lựa chọn ngôn ngữ lập trình, cấu trúc dữ liệu. Có thuật toán tốt ta cần chọn cấu trúc dữ liệu phù hợp đồng thời phải có kỹ thuật cài đặt chương trình tốt thì công việc lập trình mới đạt hiệu quả cao.

Cấu trúc dữ liệu: (Data structures) là các đơn vị cấu trúc của ngôn ngữ lập trình dùng để biểu diễn các mô hình dữ liệu. Chẳng hạn như: mảng (array), bảng ghi (record), tệp (file), xâu (string), ngăn xếp (stack), hàng đợi (queue), ...

- Kiểm thử và chạy chương trình: Chương trình được viết có thể còn có lỗi, vì vậy ta cần chạy chương trình trên nhiều bộ dữ liệu vào (bộ test) để kiểm tra tính đúng đắn của chương trình.



Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

Nếu chương trình có lỗi thì người lập trình cần hiệu chỉnh cho đến khi hết lỗi cú pháp và ngữ nghĩa. Chương trình dịch giúp người lập trình phát hiện lỗi dùng cú pháp, còn lỗi ngữ nghĩa người lập trình xem lại toàn bộ chương trình, đôi lúc phải xem lại tính đúng đắn của thuật toán.

- **Viết tài liệu:** Viết tài liệu hướng dẫn sử dụng chương trình, kể cả viết chú thích để người sử dụng thuật lợi và sau này người lập trình để nâng cấp, cập nhật khi cần thiết.

## 2. Thuật toán:

**Thuật toán**, còn gọi là **giải thuật**, là một tập hợp hữu hạn hay một dãy các qui tắc chặt chẽ được sắp xếp theo một trật tự nhất định, sao cho khi thực hiện các qui tắc này từ input bài toán ta được output cần tìm

Ví dụ: Thuật toán giải phương trình bậc nhất  $P(x): ax + b = c$ , ( $a, b, c$  là các số thực), trong tập hợp các số thực có thể là một bộ các bước sau đây:

1. Nếu  $a = 0$   
 $b = c$  thì  $P(x)$  có nghiệm bất kì  
 $b \neq c$  thì  $P(x)$  vô nghiệm
2. Nếu  $a \neq 0$   
 $P(x)$  có duy nhất một nghiệm  $x = (c - b)/a$

**3. Biểu diễn thuật toán:** Có hai cách biểu diễn thuật toán, liệt kê tuần bước và dùng sơ đồ khối. Trong cách biểu diễn thuật toán bằng sơ đồ khối dùng 5 hình cơ bản:

- + Hình Oval: Thao tác bắt đầu và kết thúc.
- + Hình bình hành: Thao tác tính điều kiện
- + Hình chữ nhật: Thao tác tính toán.
- + Hình mũi tên: Đường đi của thuật toán

Ví dụ: Thuật toán tìm BCNN(a,b).

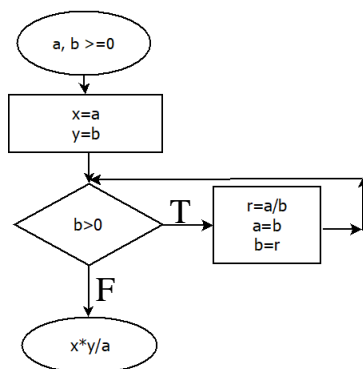
+ **Liệt kê từng bước.**

B1. Input: a, b không âm.

**B2. Tìm UCLN(a,b). (thuật toán Euclid)**

B3. Output:  $a*b/UCLN(a,b)$

+ **Sơ đồ khối.**



## 4- Tính chất của thuật toán:

- Tính chính xác: để đảm bảo kết quả tính toán hay các thao tác mà máy tính thực hiện được là chính xác.
- Tính rõ ràng: Thuật toán phải được thể hiện bằng các câu lệnh minh bạch; các câu lệnh được sắp xếp theo thứ tự nhất định.



- Tính khách quan: Một thuật toán dù được viết bởi nhiều người trên nhiều máy tính vẫn phải cho kết quả như nhau.
- Tính phổ dụng: Thuật toán không chỉ áp dụng cho một bài toán nhất định mà có thể áp dụng cho một lớp các bài toán có đầu vào tương tự nhau.
- Tính kết thúc: Thuật toán phải gồm một số hữu hạn các bước tính toán.
  - Từ các tính nhất trên, thuật toán để giải một bài toán có tính tổng quát, tức là nếu ta chỉ ra được một trường hợp sai của thuật toán thì thuật toán đó sai.

### 5- Độ phức tạp của thuật toán: Ký hiệu O

Tiêu chí quan trọng đánh giá thuật toán là thời gian chạy chương trình của thuật toán đó, thời gian chạy chương trình phụ thuộc vào nhiều yếu tố. Để đánh giá chương trình (thuật toán) là dựa vào mỗi câu lệnh của chương trình nguồn sẽ thực hiện bao nhiêu lần trên tập dữ liệu vào. Để tính toán độ phức tạp của thuật toán ta có các qui tắc sau: Quy tắc hằng số -  $O(1)$ , Quy tắc cộng, qui tắc lấy max, qui tắc nhân.

Trong ngôn ngữ lập trình bậc cao có các câu lệnh thường dùng như sau:

- ✓ Câu lệnh đơn  $O(1)$ : chẳng hạn câu lệnh gán ( $x=5$ ), lệnh đọc /ghi đơn giản ( $\text{cin}>>x/\text{cout}<<x$ ), câu lệnh chuyển điều khiển (break, continue, return, exit, goto)
- ✓ Câu lệnh ghép (hợp thành): là dãy các câu lệnh đơn tạo thành một khối lệnh, Được tính độ phức tạp theo qui tắc cộng và qui tắc max.
- ✓ Câu lệnh rẽ nhánh ( if, case):  $f(n)$ ,  $g(n)$  là độ phức tạp của hai nhánh của lệnh thì  $O(\max(f(n),g(n)))$ .
- ✓ Câu lệnh lặp (for, while): Độ phức tạp được áp dụng theo qui tắc nhân, nghĩa là  $O(k(n)f(n))$  trong đó  $k(n)$  là số lần lặp,  $f(n)$  là độ phức tạp các lệnh bên trong vòng lặp.

#### Chú ý:

- Hàm/ thủ tục có thể xem như một chương trình độc lập, có thể áp dụng qui tắc trên để tính thời gian thực hiện chương trình. Tuy nhiên việc tính thời gian thực hiện hàm/thủ tục đệ qui khá phức tạp.

- Thời gian thực hiện chương trình ở máy tính chấm bài chuẩn là  $10^6/(1\text{giây})$ .

**Ví dụ:** Tính độ phức tạp của thuật toán của các đoạn lệnh:

<pre>#include &lt;bits/stdc++.h&gt; using namespace std; int n, s1, s2; main(){     cin&gt;&gt;n;        //1     s1=0;           //2     for(int i=1; i&lt;=n; i++) s1=s1+i; //3     for(int j=1; j&lt;=n; j++) s2=s2+j*j //4     cout&lt;&lt;s1&lt;&lt;endl; //5     cout&lt;&lt;s2&lt;&lt;endl; //6 }</pre>	Các lệnh 1, 2, 5, 6 -> $O(1)$ Các lệnh 3, 4 -> $O(n)$ Vậy thời gian thực hiện chương trình là $\text{Max}(O(1), O(1), O(n), O(n), O(1), O(1)) = O(n)$ ,
<pre>main(){     cin&gt;&gt;n;        //1     d=0;           //2     for(int i=1; i&lt;=2*n; i++) d=d+1; //3</pre>	Thời gian thực hiện chương trình phụ thuộc vào n. Các lệnh 1, 2, 5 -> $O(1)$ Lệnh 3 -> $O(2n)$

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

<pre>for(int i=1; i&lt;=n; j++) //4     for(int j=1; j&lt;=n; j++) d=d+1; cout&lt;&lt;d&lt;&lt;endl; //5 }</pre>	<p>Lệnh 4 -&gt; <math>O(n^2)</math>  <b>→ <math>O(n^2)</math></b></p>
<pre>for(int i=1; i&lt;=n; j++) //1     for(int j=1; j&lt;=i; j++) //2         d=d+1; //3</pre>	<p>Câu lệnh 3 -&gt; <math>O(1)</math>          Khi <math>i=1</math>, <math>j</math> chạy từ 1 đến 1          Khi <math>i=2</math>, <math>j</math> chạy từ 1 đến 2          Khi <math>i=n</math>, <math>j</math> chạy từ 1 đến <math>n</math>          Như vậy lệnh 3 được lặp  <math>1+2+3+\dots+n = n(n+1)/2</math>  <b>→ <math>O(n^2)</math></b></p>

## BÀI TẬP 1 THUẬT TOÁN VÀ ĐỘ PHỨC TẠP THUẬT TOÁN.

### Bài 1: TÌM SỐ LỚN NHẤT

Nêu ý tưởng thuật toán tìm số lớn nhất của dãy có  $n$  số nguyên  $a_1, a_2, \dots, a_n$

### Bài 2: SẮP XẾP DÃY SỐ

Nêu ý tưởng thuật toán sắp xếp dãy số có  $n$  số nguyên  $a_1, a_2, \dots, a_n$  theo thứ tự không giảm

### Bài 3- SỐ CHÍNH PHƯƠNG

Số chính phương là số khi ta khai căn bậc hai là một số nguyên. Chẳng hạn các số 4, 9, 16 là số chính phương. Viết thuật toán (liệt kê từng bước và tính độ phức tạp) kiểm tra một số nguyên dương  $x$  có phải là số chính phương hay không? Nếu  $x$  là số chính phương thì xuất ra “YES” ngược lại xuất ra “NO”.

### Bài 4- SỐ PHONG PHÚ

Số phong phú là các số mà tổng các ước số của số đó (không kể chính nó) lớn hơn số đó. Chẳng hạn, số 12 có tổng các ước số (không kể 12) là  $1 + 2 + 3 + 4 + 6 = 16 > 12$ . Do đó 12 là một số phong phú. Hãy viết thuật toán (liệt kê từng bước và tính độ phức tạp) kiểm tra số nguyên  $x$  có phải là số phong phú hay không? Nếu  $x$  là số phong phú thì xuất ra “YES” ngược lại xuất ra “NO”.

### Bài 5- SỐ NGUYÊN TỐ.

Số nguyên tố là số có đúng hai ước số. Viết thuật toán (liệt kê từng bước và tính độ phức tạp) kiểm tra một số nguyên dương  $x$  có phải là số nguyên tố hay không? Nếu là số nguyên tố thì xuất ra “YES” ngược lại xuất ra “NO”.

### Bài 6: XÂU ĐỐI XỨNG - Palindrome

Xâu đối xứng là chuỗi khi ta viết từ trái sang phải cũng chính là chuỗi khi ta viết từ phải sang trái. Chẳng hạn: chuỗi “madam” gọi là chuỗi đối xứng, chuỗi “nqd” không gọi là chuỗi đối xứng. Hãy viết thuật toán kiểm tra một chuỗi có phải là chuỗi đối xứng hay không? và đánh giá độ phức tạp của thuật toán.

### Bài 7- DÃY SỐ FIBONACCI.

Dãy số Fibonacci được Fibonacci, một nhà toán học người Ý, công bố vào năm 1202 qua hai bài toán: Bài toán con thỏ và bài toán số các “cụ tổ” của một ong đực.

Dãy số được tính như sau:  $F(1)=1, F(2)=1, F(3)=2, F(4)=3, \dots, F(n)=F(n-1)+F(n-2)$ .





Hãy viết thuật toán tính giá trị của  $F(n)$  (liệt kê từng bước và tính độ phức tạp).

## Bài 8- Số T-PRIME

Số T-Prime là số có đúng ba ước số. Hãy viết thuật toán (liệt kê từng bước và tính độ phức tạp) đếm đoạn từ 1 đến  $n$  có bao nhiêu số T-Prime. Chẳng hạn,  $n = 6$  có một số T-Prime đó là số 4 vì có đúng ba ước số là 1, 2, 4.

## CHUYÊN ĐỀ 1. BÀI TOÁN - THUẬT TOÁN

### Bài 3: CÀI ĐẶT VÀ SỬ DỤNG CODE BLOCKS

## I- GIỚI THIỆU NGÔN NGỮ LẬP TRÌNH C/C++

### a- Lịch sử hình thành:

Ngôn ngữ lập trình C do Dennis Ritchie (người Mỹ) phát triển năm 1972. Năm 1980 Bjarne Stroustrup (người Mỹ) phát triển ngôn ngữ lập trình C++ bao trùm lên C.

Ngôn ngữ lập trình C++ là ngôn ngữ lập trình mạnh mẽ, cung cấp nhiều thư viện và hướng đối tượng.

### b- Môi trường lập trình C/C++

Trong giáo dục thường sử dụng Code Block vì miễn phí và hỗ trợ tương đối tốt cho người bắt đầu học ngôn ngữ lập trình C/C++, ngoài ra lập trình viên cũng có thể sử dụng Turbo C++, Dev C++, Visual C++, ...

## II- CÀI ĐẶT CODE BLOKS

### Bước 1: Tải phần mềm CodeBlock:

- Vào địa chỉ website: <http://www.codeblocks.org/downloads/binaries>

- Nháy vào nhãn: codeblocks-17.12mingw-setup.exe để tải IDE cài đặt trong môi trường windows (32 bit/64 bit)



Windows XP / Vista / 7 / 8.x / 10:

File	Date	Download from
codeblocks-17.12-setup.exe	30 Dec 2017	FossHUB or Sourceforge.net
codeblocks-17.12-setup-nonadmin.exe	30 Dec 2017	FossHUB or Sourceforge.net
codeblocks-17.12-nosetup.zip	30 Dec 2017	FossHUB or Sourceforge.net
codeblocks-17.12mingw-setup.exe	30 Dec 2017	FossHUB or Sourceforge.net
codeblocks-17.12mingw-nosetup.zip	30 Dec 2017	FossHUB or Sourceforge.net

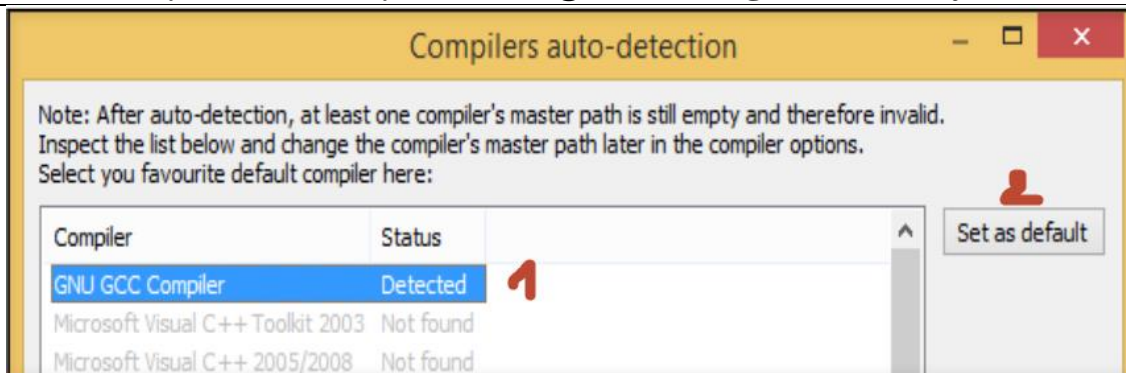
### Bước 2: Cài đặt: (giống cài đặt các phần mềm khác)

- Chạy codeblocks-17.12mingw-setup.exe đã tải về máy.

- Nháy lệnh Next, ..., Install ->Chờ phần mềm cài vào máy -> Nháy lệnh Yes để mở CodeBlocks lần làm việc đầu tiên.

- Lần làm việc đầu tiên CodeBlocs mở hộp cho ta xác lập chương trình dịch:

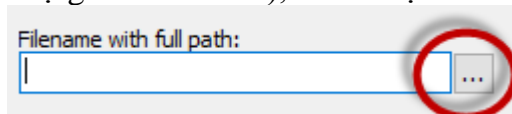
+ Chọn GNU GCC Compiler + Nháy lệnh Set Default



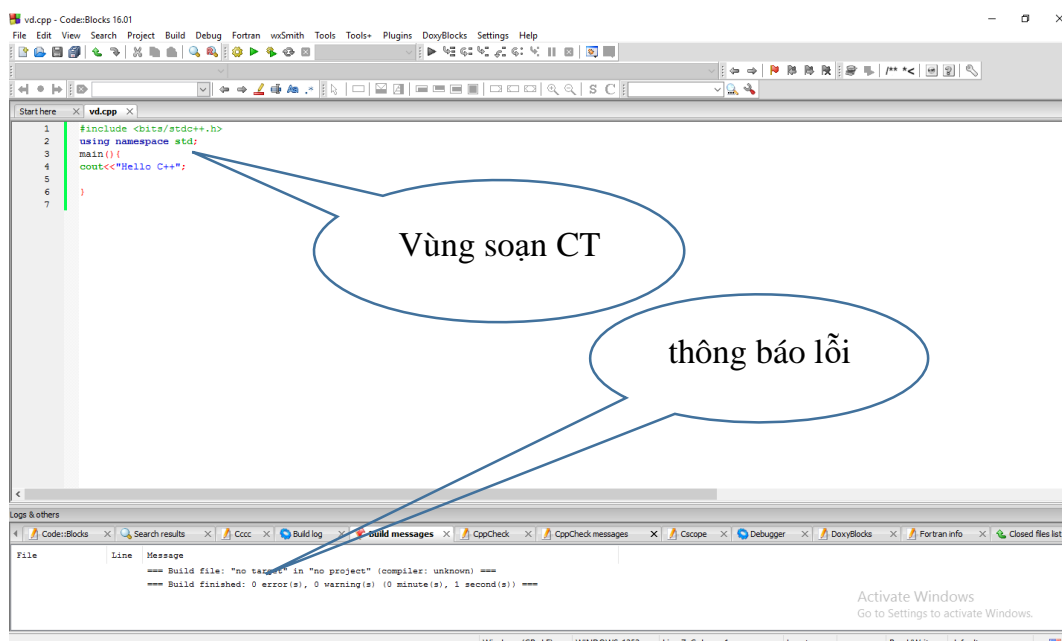
### III- TẠO TẬP CHƯƠNG TRÌNH:

#### - Bước 1: Tạo tập mới

- + Nháy menu File, New, File
- + Nháy lệnh C/C++ Source, Nháy lệnh Go
- + Nháy lệnh C++, Nháy nút lệnh Next
- + Đặt tên tập (CodeBlocks tự gán đuôi .CPP), để chỉ định nơi lưu tập nháy vào nút ...



#### - Bước 2: Soạn chương trình:




**Ví dụ:** Viết chương trình xuất ra màn hình dòng chữ: Hello C++


- Gõ vào các dòng lệnh như sau:


```
#include <bits/stdc++.h>
using namespace std;
main(){
cout<<"Hello C++";
}
```

- Lưu chương trình: Nháy File, Save (Ctrl –S)-> Tên tập có đuôi .CPP

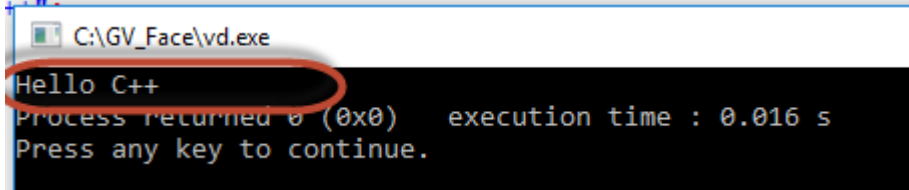
#### Bước 3: Kiểm lỗi và chạy chương trình:

- Nháy lệnh:  Build (Ctrl-F9) để kiểm lỗi cú pháp các câu lệnh trong chương trình. Nếu chương trình có lỗi tiến hành sửa lỗi cho đến khi chương trình hết lỗi (thông thường lỗi tại dòng có vạch đỏ hoặc dòng phía trên dòng vạch đỏ).

- Nháy lệnh  Run (Ctrl-F10) để chạy chương trình: test sự đúng đắn của chương trình bằng cách nhập các bộ input, nếu output sai đó là lỗi ngữ nghĩa, ta cũng phải sửa chương trình cho đến khi output đúng.

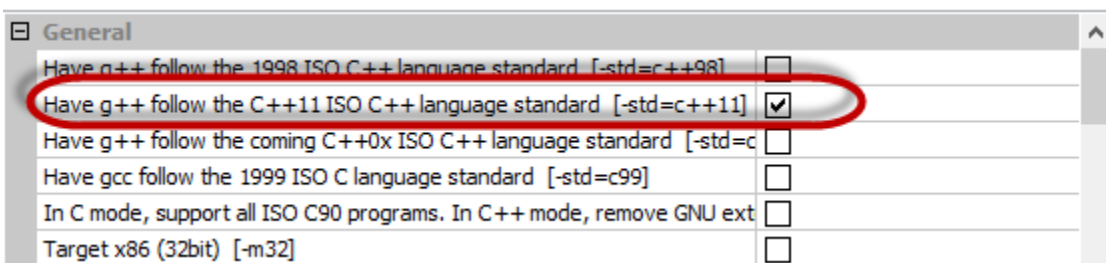
- Nút lệnh  Build/Run (F9) gồm lệnh Build và lệnh Run

#### Kết quả khi chạy chương trình trên:



- Ấn phím enter để trở về CodeBlock

Ghi chú: Nếu CodeBlocs không chạy được chương trình do thiếu bộ dịch. Thiết lập như sau: Nháy menu Settings, Compiler -> Chọn std=c++11. Ta nên dùng C++98 hoặc C++11 vì C++14 thì HSG quốc gia Bộ chưa cho phép.





## CHUYÊN ĐỀ 2: NGÔN NGỮ LẬP TRÌNH C++ Bài 1. NHẬP XUẤT DỮ LIỆU

### 1. Cấu trúc chương trình đơn giản:

```
#include <bits/stdc++.h>
using namespace std;
[Khai báo hằng]
[Phân khai báo biến toàn cục]
[Khai báo chương trình con]
main(){
    [<Các câu lệnh>;]
}
```

#### Giải thích:

- Khai báo hằng nếu có; có thể là hằng xâu hay hằng số. Để khai báo hằng ta dùng cú pháp:

**const <kiểu dữ liệu> <tên hằng>=<Giá trị>;**

Ví dụ: const int N=1e9; N là số nguyên, có giá trị =  $10^9$

- Khai báo biến toàn cục nếu có. Để khai báo biến toàn cục ta dùng cú pháp

**<tên kiểu dữ liệu> <tên biến>** biến có giá trị khởi tạo mặc định = **0**. Nếu khai báo biến trong hàm thì giá trị khởi tạo là giá trị ngẫu nhiên.

Ví dụ: int x, y; Biến x và biến y nhận giá trị là các số nguyên (4B) trong phạm vi - 2147483648 tới 2147483647. Để dễ nhớ biến nguyên nhận giá trị trong đoạn  $[-2 \times 10^9 .. 2 \times 10^9]$

- Khai báo chương trình con nếu có.

+ Khai báo hàm dùng cú pháp:

```
<tên kiểu dữ liệu> <tên hàm>(<tham số hình thức>){
    [<Các câu lệnh>;]
    return <giá trị trả về của hàm>;
}
```

+ Khai báo thủ tục dùng cú pháp: (còn gọi là hàm không trả về giá trị)

```
void <tên thủ tục>(<tham số hình thức>){
    [<các câu lệnh>;]
}
```

**main(){.....}** là hàm chính trong chương trình, bắt buộc phải có. Nếu dùng **int main()** thì hết hàm main có từ khóa **return 0;**

- Chương trình viết bằng C++ sẽ thực hiện tuần tự các câu lệnh từ trên xuống dưới. Nếu chương trình con A, gọi chương trình con B thì chương trình con B phải được viết trước chương trình con A.

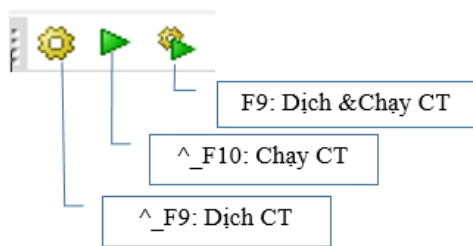
Ví dụ: Viết chương trình xuất ra thông báo: **“Helo C++”**

<pre>#include &lt;bits/stdc++.h&gt; using namespace std; main(){     cout&lt;&lt;"Helo C++"; }</pre>	<pre>// Chèn các thư viện //sử dụng thư viện chuẩn //Hàm chính-&gt;Bắt buộc phải có //Xuất xâu: Helo C++ //kết thúc hàm chính</pre>
--	---

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

Code và dịch/chạy chương trình trong Codeblock

**Chú ý:** Thanh công cụ có nút lệnh:



Khi Dịch/Chạy chương trình nếu chương trình còn lỗi -> Xuất hiện màu đỏ -> Xem sửa tại dòng màu đỏ hoặc dòng phía trên chỉ thị màu đỏ.

**Ghi nhớ:** - Ngôn ngữ C++ phân biệt kí tự hoa, kí tự thường. Các câu lệnh trong C++ được viết chữ thường.

- Để ghi chú mã lệnh ta có hai cách: (dòng ghi chú không có tác dụng trong chương trình)

+ Đặt dấu // trước dòng chú thích

+ Đặt dấu /\* trước khối chú thích và \*/ hết khối chú thích

- Các câu lệnh đặt trên một dòng lệnh ngăn cách bởi dấu phẩy. Dấu chấm phẩy để hết một dòng lệnh.

- cặp ngoặc {...} dùng để ghép các câu lệnh đơn thành câu lệnh phức hợp.

## 2- Câu lệnh nhập/ xuất:

- Xuất ra màn hình: **cout<<"Biểu thức";**

Biểu thức có thể là hằng xâu, biến. Hằng xâu được đặt trong cặp nháy đôi

- Đọc dữ liệu từ bàn phím: **cin>><tên biến>;**

Ví dụ: Chương trình xuất ra màn hình tuổi hiện tại (năm 2019)

<pre>#include &lt;bits/stdc++.h&gt; using namespace std; int ns; main(){     cout&lt;&lt;"Nhập nam sinh";     cin&lt;&lt;ns;     cout&lt;&lt;endl;      cout&lt;&lt;"Tuoi ban den nam 2019 la: "&lt;&lt;2019- ns; }</pre>	<pre>// Chèn các thư viện //sử dụng thư viện chuẩn // khai báo biến ns nhận giá trị nguyên //Hàm chính-&gt;Bắt buộc phải có //Xuất xâu: Nhập nam sinh //ns= năm sinh người dùng nhập Ví dụ: nhập 2004 //xuống dòng //Xuất ra màn hình: Tuoi ban den nam 2019 la: 15</pre>
---	---

## 3- Kiểu dữ liệu cơ bản: (Để dễ nhớ miền giá trị, tài liệu ghi giá trị gần đúng)

Tên kiểu	Ý nghĩa	Dung lượng	Miền giá trị
bool	Kiểu logic	1B	false (0)/ True(1)
Char Unsigned char	Kiểu kí tự	1B	[-127..127] 0 tới 255
int	Kiểu số nguyên	4B	-2x10 <sup>9</sup> tới 2x10 <sup>9</sup>
long long	Kiểu số nguyên	8B	-9x10 <sup>18</sup> đến 9x10 <sup>18</sup>



Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

float	Kiểu số thực	4B	+/- 3.4e +/- 38 (~7 chữ số)
double	Kiểu số thực	8B	+/- 1.7e +/- 308 (~15 chữ số)
string	Kiểu chuỗi ký tự	4B	Độ dài $10^9$ (còn phụ thuộc trình dịch)

- Kiểu int: Giá trị nhỏ nhất INT\_MIN, giá trị lớn nhất INT\_MAX

- Kiểu long long: Giá trị nhỏ nhất LLONG\_MIN, giá trị lớn nhất LLONG\_MAX

#### 4- Phép toán, biểu thức và hàm cơ bản:

##### a- Phép toán:

- Toán tử số học: +, -, \*, / (Cộng, trừ, nhân, chia); Chia lấy nguyên /; chia lấy dư: %. ++x tăng 1 (tiền tố), x++ tăng 1 (hậu tố); --x giảm 1 (tiền tố), x-- giảm 1 (hậu tố)

- Toán tử quan hệ: == (bằng), != (khác), > (lớn) >= (lớn hơn =), < (nhỏ), <= (nhỏ hơn =)

- Toán tử logic: && (và), || (hoặc), ! (phủ định)

**b- Biểu thức:** là tập hợp các biến, các hằng được nối với nhau bởi các phép toán.

Trong biểu thức có thứ tự độ ưu tiên các phép toán là Số học, quan hệ, logic. Nếu phép toán đồng cấp thì thực hiện từ trái sang phải. Ta chỉ dùng cặp ngoặc đơn để nhóm các độ ưu tiên của phép toán. Chẳng hạn:  $-b/2*a$  và  $-b/(2*a)$  là khác nhau.

##### c- Hàm cơ bản:

- Hàm abs(x): |x|, sqr(x):  $x^2$ , sqrt(x):  $\sqrt{x}$ ,

- Hàm min(a, b); max(a, b): cho kết quả là số nhỏ nhất/lớn nhất trong hai số a, b

- Hàm swap(a,b): đổi giá trị a và b, pow(a,b) tính a lũy thừa b (a, b có thể là số thực).

- Hàm round(x): Làm tròn số thực x tới hàng đơn vị.

Ví dụ: Câu lệnh xuất số thực x ra màn hình và lấy 1 chữ số thập phân (1 chữ số lẻ)

```
cout<< round(x*10)/10;
```

- Hàm return: kết thúc chương trình. Hàm \_\_gcd(a,b): Ước chung lớn nhất của hai số nguyên a, b

- Chuyển đổi chuỗi dạng số nguyên sang số nguyên: Dùng hàm stoi(xâu số). Hoặc sử dụng hàm atoi(s.c\_str()). Ví dụ: string s="12345"; int x=atoi(s.c\_str()); cout<<x;

- Chuyển đổi số nguyên thành chuỗi: Dùng hàm to\_string(số nguyên). Hoặc sử dụng hàm itoa().

Ví dụ:

```
#include <bits/stdc++.h>
using namespace std;
main(){
    string s="1234";
    int x=stoi(s);
    cout<<x<<endl; // x là số có giá trị 1234
    int y=100;
    string r = to_string(y);
    cout<<r; // r là chuỗi có giá trị "100"
}
```

Hàm stoi() và hàm to\_string() chỉ áp dụng được C++11 trở lên

#### 5- Khai báo biến:



- Cú pháp: <Tên kiểu dữ liệu> <tên biến>;

Tên biến chỉ bao gồm chữ cái, chữ số và dấu gạch nối, nhưng bắt đầu không được chữ số.

- Biến có hai loại: biến toàn cục và biến cục bộ. Biến toàn cục các chương trình con có thể sử dụng và có giá trị tự khởi tạo là 0. Biến cục bộ chỉ có hiệu lực nơi nó khai báo và giá trị khởi tạo là giá trị ngẫu nhiên.

Ví dụ: int x, y; // Khai báo hai biến nguyên (biến nguyên 4B x 2 = 8B bộ nhớ)

int z=10; // Khai báo biến nguyên z và khởi tạo ban đầu có giá trị bằng 10.

## 6- Câu lệnh gán:

- Cú pháp: <Biến> = <Biểu thức>;

- Ý nghĩa: Tính giá trị biểu thức và gán giá trị này cho biến. Biến và biểu thức phải cùng kiểu dữ liệu.

- Chẳng hạn:  $x=14/3$ ; Nếu x biến nguyên thì  $x=4$ . Nếu x là biến thực  $x=4.666666667$  (số thực có sai số phần thập phân). Xem thêm so sánh hai số thực  $a=b$  và xuất số thực.

Ví dụ: y=5; y+=10; Hai lệnh này thì y có giá trị 15;

Ví dụ: Chương trình tính tuổi có dùng biến trung gian t.

<pre>#include &lt;bits/stdc++.h&gt; using namespace std; int a, b; main(){     cout&lt;&lt;"Chương trình tính tuổi";     cout&lt;&lt;endl;     cout&lt;&lt;"Nhập nam sinh = ";     cin&gt;&gt;a;     cout&lt;&lt;endl;     cout&lt;&lt;"Nhập nam hiện tại = ";     cin&gt;&gt;b;     cout&lt;&lt;endl;     int t=b-a+1;     cout&lt;&lt;"Tuổi của bạn là: "&lt;&lt;t; }</pre>	<p>//khai báo hai biến toàn cục a, b là số nguyên</p> <p>//Xuống hàng</p> <p>//Nhập một số a</p> <p>Nhập một số b</p> <p>//khai báo biến cục bộ t và tính tuổi</p> <p>//Xuất thông báo và tuổi</p>
---	--

## Chú ý quan trọng: Nếu a, b là số nguyên (int):

- Khi chia a cho b kết quả luôn là số nguyên (int); Để kết quả là số thực thì ta phải khai báo a hoặc b là số thực (double). Để kết quả là số nguyên long long thì ta phải khai báo a hoặc b có kiểu long long. Hoặc dùng công thức chuyển kiểu biến a hoặc biến b lên long long: **1ll\*(a)\*b; hoặc 1ll\*(b)\***. Để chuyển biến x kiểu thực về kiểu nguyên ta viết **int(x)**.

Ví dụ: biến **a** và **b** kiểu nguyên int có giá trị là  $10^6$ . Khi nhân **a** cho **b** thì kết quả là  $10^{12}$  vượt kiểu int.

<pre>#include &lt;bits/stdc++.h&gt; using namespace std; int a, b, c; long long d;</pre>
--

```
main(){
    a=b=1e6;
    c=a*b; // giá trị c sai
    d=1ll*(a)*b; // giá trị d đúng
    cout<<c<<endl;
    cout<<d;
}
```

## BÀI TẬP 2 NHẬP XUẤT DỮ LIỆU

### Bài 1. CẤP SỐ CỘNG.

Trong toán học, một **cấp số cộng** là một dãy số thỏa mãn điều kiện: hai phân tử liên tiếp nhau sai khác nhau một hằng số. Hằng số này gọi là **công sai** của cấp số cộng.

Chẳng hạn: 1; 2; 3; ...; n

**Yêu cầu:** Nhập từ bàn phím số nguyên dương **n**, tính tổng các số từ 1 đến n và xuất ra màn hình.

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
Nhap so nguyen duong n = 4	Tong la: 10

### Bài 2. HÌNH CHỮ NHẬT (Tuyển sinh 10 – 2014)

Hình chữ nhật là loại hình cơ bản trong hình học. Với chu vi không đổi thì hình chữ nhật có diện tích lớn nhất khi nó là hình vuông.

**Yêu cầu:** Nhập từ bàn phím một số thực **p** là chu vi hình chữ nhật. Hãy tính diện tích lớn nhất của hình chữ nhật có thể đạt được với chu vi đã cho. Xuất kết quả ra màn hình (kết quả lấy một chữ số thập phân).

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
Nhap chu vi HCN P = 12	Dien tich lon nhat HCN la: 9.0

*Giải thích :* Với chu vi  $p=12$ , hình chữ nhật có diện tích lớn nhất là hình vuông có cạnh là 3, nên diện tích là 9.0

**Ràng buộc:** Số **p** là số thực dương và có giá trị không lớn hơn 1000.

### Bài 3. HÌNH TAM GIÁC

Hình tam giác (tam giác) là một loại hình cơ bản trong hình học. Diện tích **S** của tam giác được tính theo công thức Heron :

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

Trong đó: a, b, c là độ dài 3 cạnh, p là nửa chu vi của tam giác.

**Yêu cầu:** Nhập từ bàn phím 3 số thực dương a, b, c lần lượt là độ dài ba cạnh của tam giác. Tính chu vi, diện tích của tam giác. Xuất kết quả ra màn hình (kết quả lấy một chữ số thập phân).

**Ví dụ:**



Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu

Nhập từ bàn phím	Xuất ra màn hình
- Nhập do dai canh a = 3.5 - Nhập do dai canh b = 4.5 - Nhập do dai canh c = 5.5	- Chu vi tam giac la: 13.5 - Dien tich tam giac la: 7.9

**Giới hạn:** Số a, b, c là các số thực dương và có giá trị không lớn hơn 1000.

#### **Bài 4: ĐÈN THẮP SÁNG**

Nhà Nam sử dụng bóng đèn để thắp sáng là loại bóng đèn huỳnh quang tiết kiệm điện, mỗi bóng đèn đều có ghi công suất (W). Điện năng tiêu thụ (kWh) = Công suất bóng đèn (kW) x Số giờ sử dụng (h):  $A = P \times T$ ;  $1\text{kW} = 1000\text{W}$ . Em hãy viết chương trình giúp Nam tính tiền điện của một bóng đèn trong một tháng, biết rằng giá điện bình quân là 1500 đồng/kWh.

**Yêu cầu:** Nhập từ bàn phím số nguyên dương P là công suất bóng đèn (W), số nguyên dương T là thời gian (h) thắp sáng của bóng đèn trong một tháng. Tính và xuất ra màn hình số tiền điện (đồng) của bóng đèn đó (kết quả lấy một chữ số thập phân).

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
- Nhập công suất bóng đèn (W) P = 20 - Nhập số giờ thắp sáng (h) T = 100	Số tiền điện: 3000.0
- Nhập công suất (W) P = 45 - Nhập số giờ thắp sáng (h) T = 73	Số tiền điện: 4927.5

**Ràng buộc:**  $0 < P \leq 100$ ;  $0 < T \leq 300$

#### **Bài 5: TÍNH TIỀN KHÁCH SẠN**

Một người ở khách sạn n ngày, khách sạn có hai mệnh giá: Giá tuần là 5 trăm ngàn đồng, giá ngày lẻ là 1 trăm ngàn đồng.

**Yêu cầu:** Nhập từ bàn phím số ngày ở khách sạn n. Xuất ra màn hình số tiền người đó phải thanh toán.

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
Nhập vào số ngày n = 10	Số tiền là: 8

**Ràng buộc:**  $0 < n \leq 10^5$ ;

### CHUYÊN ĐỀ 2. NGÔN NGỮ LẬP TRÌNH C++ Bài 2. CÂU LỆNH Rẽ NHÁNH

Đôi khi chúng ta muốn làm cho sự thực thi một lệnh phụ thuộc vào một điều kiện nào đó cần được thỏa mãn. Lệnh if cung cấp cách để thực hiện công việc này:

#### **1- Câu lệnh if dạng khuyết:**

- Cú pháp:

```
if(biểu thức){
    lệnh 1;
    lệnh 2;
    .....
```



lệnh n

}

- Ý nghĩa: Nếu biểu thức có giá trị **đúng** thì thực hiện lệnh 1, lệnh 2, ..., lệnh n.  
Nếu biểu thức có giá trị sai thì bỏ qua câu lệnh if

## 2- Câu lệnh if dạng đủ:

- Cú pháp:

```
if(biểu thức){  
    lệnh 1.1;  
    lệnh 1.2;  
    .....  
    lệnh 1.n  
}  
else{  
    lệnh 2.1;  
    lệnh 2.2;  
    .....  
    lệnh 2.m  
}
```

- Ý nghĩa: Nếu biểu thức có giá trị **đúng** thì thực hiện lệnh 1.1, 1.2, ...1.n. Nếu biểu thức có giá trị **sai** thì thực hiện lệnh 2.1, 2.2, ..., 2.m.

Ví dụ: Chương trình xét tính chẵn, lẻ một số nguyên nhập từ bàn phím

### Cách 1:

```
#include <bits/stdc++.h>  
using namespace std;  
int x;  
main(){  
    cout<<"Nhập một số nguyên x = ";  
    cin>>x;  
    if (x%2==0){  
        cout<<x<<" là số chẵn";  
    }  
    if (x%2!=0){  
        cout<<x<<" là số lẻ";  
    }  
}
```

### Cách 2:

```
#include <bits/stdc++.h>  
using namespace std;  
int x;  
main(){  
    cout<<"Nhập một số nguyên x = ";  
    cin>>x;  
    if (x%2==0){  
        cout<<x<<" là số chẵn";  
    }  
    else{  
        cout<<x<<" là số lẻ";  
    }  
}
```

## 3- Lệnh Switch: (Lệnh này ít dùng)

```
switch (biểu thức) {  
    case hằng 1 : {  
        các lệnh;  
        break;  
    };  
    .....  
    case hằng n: {  
        các lệnh;  
        break;  
    }  
}
```



```
default: {  
    các lệnh;  
}
```

**Biểu thức** (gọi là thẻ switch) được tính trước và kết quả được so sánh với mỗi hằng số từ trên xuống (gọi là các nhãn), nếu so khớp được tìm thấy, *lệnh* sau nhãn so khớp được thực hiện cho đến khi gặp lệnh break. Nếu so khớp không tìm thấy thì thực hiện lệnh sau nhãn default

**Ví dụ:** Nhập từ bàn phím hai số nguyên a và b. Để tính tổng hoặc hiệu hoặc tích hai số phụ thuộc điều kiện nhập của biến n.

```
#include <bits/stdc++.h>  
using namespace std;  
int a, b, n;  
main(){  
    cout<<"Nhập hai số a và b: ";  
    cin>>a>>b;  
    cout<<endl;  
    cout<<"Nhập phép toán "<<endl;  
    cout<<"1: Phép toán cộng: a+b\n";  
    cout<<"2: Phép toán trừ: a-b\n";  
    cout<<"3: Phép toán nhân: a*b\n";  
    cin>>n;  
    switch(n){  
    case 1:{  
        cout<<"Tổng = "<<a+b;  
        break;  
    }  
    case 2:{  
        cout<<"Hiệu = "<<a-b;  
        break;  
    }  
    default:  
        cout<<"Tích = "<<a*b;  
    }  
}
```



### BÀI TẬP 3- CÂU LỆNH Rẽ NHÁNH

#### **Bài 1: TÍNH TIỀN NƯỚC**

Công ty cung cấp nước sinh hoạt ở vùng ngập mặn khuyến khích các hộ dân tiết kiệm nước bằng cách đưa ra hai mức giá. Nếu hộ nào trong một tháng sử dụng không quá 10 m<sup>3</sup> nước thì đơn giá mỗi m<sup>3</sup> là 4000 đồng. Nếu sử dụng vượt quá 10 m<sup>3</sup> thì mỗi m<sup>3</sup> vượt có giá là 7000 đồng.

**Yêu cầu:** Nhập từ bàn phím số nguyên dương **n** ( $0 < n \leq 100$ ) là số m<sup>3</sup> nước sử dụng trong tháng của một hộ gia đình. Hãy tính số tiền nước phải trả của hộ gia đình trong tháng đó. Kết quả xuất ra màn hình.

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
------------------	------------------

Nhap n = 5	So tien = 20000
Nhap n = 13	So tien = 61000

## Bài 2: KANGAROOS

Một trò chơi trên máy tính giúp các em nhỏ rèn luyện tính toán đơn giản. Trò chơi như sau: có 3 con Kangaroos đứng trên một đường thẳng được đánh dấu bởi dãy số nguyên dương theo thứ tự tăng. Mỗi lượt đi một trong hai con ở hai bên sẽ nhảy vào khoảng trống ở giữa hai con còn lại và chiếm lấy một vị trí tương ứng với một số nguyên đã được đánh dấu. Trò chơi kết thúc khi không còn khoảng trống giữa hai con bất kỳ.

**Yêu cầu:** Nhập từ bàn phím ba số nguyên dương **a, b, c** là vị trí ban đầu của 3 con Kangaroos ( $0 < a < b < c \leq 10^3$ ). Tính và xuất ra màn hình số lượt đi nhiều nhất có thể của trò chơi.

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình	Giải thích
Nhap a = 3 Nhap b = 5 Nhap c = 9	So luot di = 3	Con Kangaroos ở vị trí 3 nhảy vào vị trí 6. Con Kangaroos ở vị trí 5 nhảy vào vị trí 7. Con Kangaroos ở vị trí 6 nhảy vào vị trí 8. Vậy: 3 lần đi trò chơi kết thúc.

## Bài 3: CHỈ SỐ BMI

Chỉ số khối cơ thể BMI (*Body Mass Index*) được dùng để đánh giá mức độ gầy hay béo của một người. Gọi W là khối lượng của một người (tính bằng kg) và H là chiều cao của người đó (tính bằng mét), chỉ số khối cơ thể được tính theo công thức:  $BMI = \frac{W}{(H)^2}$

Theo khuyến nghị của Tổ chức Y tế Thế giới (WHO):

- Nếu  $BMI < 20$  : Người gầy.
- Nếu  $20 \leq BMI \leq 30$  : Người lí tưởng.
- Nếu  $BMI > 30$  : Người béo.

**Yêu cầu:** Nhập từ bàn phím số nguyên dương W (kg) là khối lượng cơ thể, số thực H (m) là chiều cao của một người. Tính và xuất ra màn hình chỉ số BMI và Kết luận theo WHO

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
- Nhap khoi luong (kg) W = 60 - Nhap chieu cao (m) H = 1.63	BMI= 22.6 Nguoi li tuong
- Nhap khoi luong (kg) W = 40 - Nhap chieu cao (m) H = 1.5	BMI= 17.8 Nguoi gay
- Nhap khoi luong (kg) W = 80 - Nhap chieu cao (m) H = 1.6	BMI= 31.3 Nguoi beo

**Giới hạn:** Số nguyên dương W có giá trị không lớn hơn 500, H là số thực dương có giá trị không lớn hơn 3.

## Bài 4. SỐ CHÍNH PHƯƠNG.

**Số chính phương** hay còn gọi là **số hình vuông** là **số tự nhiên** có **căn** bậc 2 là một số tự nhiên, hay số chính phương là **bình phương** (lũy thừa bậc 2) của một số tự nhiên khác.

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

**Yêu cầu:** Nhập một số nguyên dương từ bàn phím. Nếu n là số chính phương xuất ra màn hình Yes, ngược lại xuất No

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
Nhap so n = 9	Yes
Nhap so n = 12	No

## Bài 5. GIẢI PHƯƠNG TRÌNH BẬC HAI

Trong đại số sơ cấp, **phương trình bậc hai** là phương trình có dạng:  $ax^2 + bx + c = 0$  với x là biến, a, b, c là các hệ số đã biết.

**Yêu cầu:** Nhập ba số nguyên a, b, c từ bàn phím. Tính và đưa ra màn hình nghiệm của phương trình.

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
- Nhap a = 1 - Nhap b = 2 - Nhap c = 3	PTVN
- Nhap a = 1 - Nhap b = 2 - Nhap c = -3	x1= 1 x2= -3
- Nhap a = 1 - Nhap b = 2 - Nhap c = 1	x1=x2= -1



## CHUYÊN ĐỀ 2. NGÔN NGỮ LẬP TRÌNH C++

### Bài 3. CÂU LỆNH LẶP

#### 1. Lệnh while:

**Cú pháp:**

```
while (biểu thức) {  
    lệnh 1;  
    lệnh 2;  
    -----;  
    lệnh n;  
}
```

- **Ý nghĩa:** Trong khi biểu thức còn có **giá trị đúng** thì thực hiện lệnh 1, lệnh 2, ..., lệnh n cho đến khi **biểu thức có giá trị sai** thì câu lệnh while kết thúc.

Đôi khi chúng ta có thể gặp vòng lặp **while có thân rỗng** (nghĩa là một câu lệnh null). Ví dụ vòng lặp sau đặt n tới thừa số lẻ lớn nhất của nó.

```
while (n % 2 == 0 && n /= 2) ;
```

#### 2. Lệnh do...while (lệnh này ít dùng):

- **Cú pháp:**

```
do {
```



```
lệnh 1;  
lệnh 2;  
.....  
lệnh n;  
}while (biểu thức);
```

- Ý nghĩa: Lệnh 1, 2, ... lệnh n được thực hiện trước và sau đó mới tính giá trị của biểu thức. Nếu biểu thức có giá trị sai thì kết thúc câu lệnh Do While. Nếu biểu thức có giá trị đúng thì trở về thực hiện lệnh 1, 2, ..., n và cứ như thế.

**Ví dụ:** Tính tổng các số trên mặt đồng hồ: (giá trị lưu vào biến s)

<b>Dùng while:</b> int s=0, n=12; while (n>0){ s=s+n; n--; }	<b>Dùng do while</b> int s, n=12; do{ s=s+n; n--; }while (n>0);
---	--

### 3. Câu lệnh for:

```
for (biểu thức1; biểu thức2; biểu thức3){  
lệnh 1;  
lệnh 2;  
.....  
lệnh n;  
}
```

Biểu thức1 (thường được gọi là biểu thức khởi tạo) được tính trước tiên. Mỗi vòng lặp biểu thức2 được tính. Nếu biểu thức 2 có kết quả đúng thì thực hiện lệnh 1, 2, ..., n được và biểu thức 3 được tính. Nếu biểu thức 2 có kết sai thì câu lệnh for kết thúc.

Vòng lặp for tổng quát thì tương đương với vòng lặp while:

```
biểu thức1;  
while (biểu thức 2) {  
lệnh;  
biểu thức 3;  
}
```

Ví dụ: Chương trình tính tổng 5 số được nhập từ bàn phím

<b>Cách 1:</b> #include <bits/stdc++.h> using namespace std; int s, x; main(){ cout<<"TINH TONG 6 SO "; cout<<endl; for(int i=1; i<=6; i++){ cout<<"Nhap so thu "<<i<<" = "; cin>>x; s=s+x; } cout<<"Tong la: "<<s;	<b>Cách 2:</b> #include <bits/stdc++.h> using namespace std; int s, x; main(){ cout<<"TINH TONG 6 SO "; cout<<endl; int i=1; while(i<=6){ cout<<"Nhap so thu "<<i<<" = "; cin>>x; s=s+x; i=i+1;
---	---



}	} cout<<"Tổng là: "<<s; }
---	---------------------------------

Chú ý: - Câu lệnh lặp có thể lồng nhau. – biểu thức có thể làm câu lệnh lặp vô tận

- Lệnh break gặp lệnh này vòng lặp sẽ kết thúc.

- Lệnh **continue** làm cho chương trình bỏ qua phần còn lại của vòng lặp và nhảy sang lần lặp tiếp theo.

Ví dụ 1: Chương trình in ra màn hình bảng cửu chương.

<b>Cách 1:</b> <pre>#include &lt;bits/stdc++.h&gt; using namespace std; main(){     for(int i=1; i&lt;=10; i++){         for(int j=1; j&lt;=10; j++){             cout&lt;&lt;i&lt;&lt;" x "&lt;&lt;j&lt;&lt;" = "             &lt;&lt;i*j&lt;&lt;endl;             cout&lt;&lt;endl;         }     } }</pre>	<b>Cách 2:</b> <pre>#include &lt;bits/stdc++.h&gt; using namespace std; int s, x; main(){     for(int i=1; i&lt;=10; i++){         int j=1;         while(j&lt;=10){             cout&lt;&lt;i&lt;&lt;" x "&lt;&lt;j&lt;&lt;" = "             &lt;&lt;i*j&lt;&lt; endl;             j=j+1;         }         cout&lt;&lt;endl;     } }</pre>
--	---

## BÀI TẬP 4 CÂU LỆNH LẶP

### Bài 1. GIAI THỪA

Cho **n** là một số tự nhiên dương, "*n* giai thừa", kí hiệu **n!** là tích của **n** số tự nhiên dương đầu tiên.

**Yêu cầu:** Nhập số **n** từ bàn phím, in ra màn hình **n!**

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
Nhap n = 5	N!=120

### Bài 2. DÃY SỐ FIBONACII

Dãy số Fibonacci là dãy vô hạn các số tự nhiên bắt đầu bằng hai phần tử 1 và 1, các phần tử sau đó được thiết lập theo quy tắc mỗi phần tử luôn bằng tổng hai phần tử trước nó, được nhà toán học người Ý công bố vào năm 1202: Bài toán con thỏ và bài toán số các "cụ tổ" của một ong đực.

**Yêu cầu:** Nhập từ bàn phím số nguyên dương **n** ( $n > 2$ ). In ra màn hình số Fibonacci thứ **n** (**F<sub>n</sub>**).

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
Nhap n = 6	F <sub>n</sub> =8
Nhap n = 10	F <sub>n</sub> =55

### Bài 3: SỐ PHONG PHÚ

Trong số học, số phong phú là các số mà tổng các ước số của số đó (không kể chính nó) lớn hơn số đó. Ví dụ, số 12 có tổng các ước số (không kể 12) là  $1 + 2 + 3 + 4 + 6 = 16 > 12$ . Do đó 12 là một số phong phú.

**Yêu cầu:** Hãy lập trình đếm xem có bao nhiêu số phong phú trong đoạn  $[L, R]$ .

**Dữ liệu vào:** Nhập từ bàn phím 2 số nguyên  $L, R$  ( $1 \leq L \leq R \leq 10^5$ )

**Kết quả:** Xuất ra màn hình một số nguyên là số lượng số phong phú trong đoạn  $[L, R]$ .

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình	Giải thích
1 50	9	Từ 1 đến 50 có 9 số phong phú là: 12, 18, 20, 24, 30, 36, 40, 42, 48

### Bài 4. ĐỊNH ĐỀ BERTRAN

Định đề Bertran: “ Với mọi số nguyên  $n \geq 2$  bao giờ cũng tìm thấy số nguyên tố  $p$  thỏa mãn  $n < p < 2n$ ”. Định đề này do nhà toán học Pháp Jojeph Bertran đưa ra năm 1845 sau khi đã kiểm tra với mọi  $n \leq 3000000$ . Điều này đã được Tchebusep chứng minh năm 1850. Năm 1932 Erdoeus đã tìm được cách chứng minh mới đơn giản hơn.

**Yêu cầu:** cho  $n$ , hãy xác định số lượng số nguyên tố  $p$  thỏa mãn điều kiện  $n < p < 2n$ .

**Dữ liệu vào:** Nhập từ bàn phím số nguyên dương  $n$  ( $2 \leq n \leq 10^6$ ).

**Kết quả:** Ghi ra màn hình số lượng số nguyên tố  $p$ .

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
6	2
239	39



### Bài 5: XẾP GIẤY

Một tờ giấy có độ dày  $a$   $\mu\text{m}$  (mi-crô-mét). Hỏi cần phải xếp đôi tờ giấy ít nhất bao nhiêu lần để độ dày nhận được không nhỏ hơn  $b$  cm (xăng-ti-mét). Biết rằng: 1 cm = 10000  $\mu\text{m}$

**Yêu cầu:** Cho số  $a$   $\mu\text{m}$  là độ dày tờ giấy, số  $b$  cm là độ dày cần có. Tính số lần xếp giấy.

**Dữ liệu vào:** Nhập từ bàn phím hai số nguyên dương  $a, b$  ( $0 < a, b \leq 100$ ).

**Kết quả:** Xuất ra màn hình số lần xếp giấy tính được.

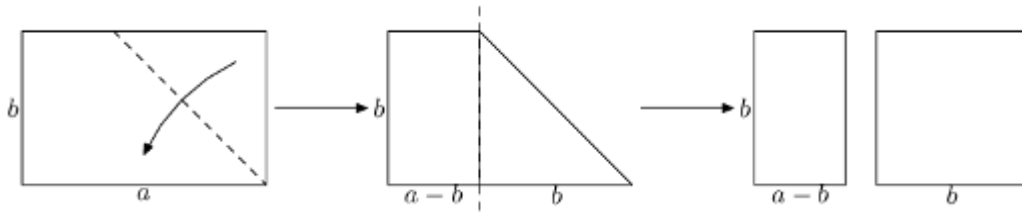
**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
- Nhập do dày tờ giấy $a = 10$ - Nhập do dày cần có $b = 1$	Số lần xếp giấy là : 10

### Bài 6: CẮT GIẤY

Để chuẩn bị cho ngày cắm trại, Nam và các bạn cùng nhau cắt những mảnh giấy màu để trang trí trại. Nhằm làm tăng thêm phần độc đáo, bạn Nam nghĩ ra một cách cắt tờ giấy màu thành các hình vuông.

Tờ giấy màu của Nam có dạng hình chữ nhật kích thước  $a \times b$  ( $a > b$ ). Sau đó Nam sẽ gấp chéo tờ giấy tạo thành đường gấp có góc 45 độ, một mép trùng với một cạnh của tờ giấy (như hình vẽ), sau đó cắt phần giấy thừa không bị gấp đè lên.



Sau khi cắt, Nam nhận được một mảnh giấy hình vuông kích thước  $b \times b$  và một mảnh kích thước  $b \times (a-b)$ . Nam lại tiếp tục thực hiện thao tác như trên với mảnh giấy  $b \times (a-b)$  và cứ thế cho đến khi tất cả các mảnh giấy đều là hình vuông.

**Yêu cầu:** Hãy xác định xem Nam có thể có được bao nhiêu mảnh giấy hình vuông?

**Dữ liệu vào:** Nhập từ bàn phím hai số nguyên  $a$  và  $b$  ( $1 \leq b \leq a \leq 10^9$ ).

**Kết quả:** Xuất ra màn hình một số nguyên là số hình vuông sau khi cắt.

**Ví dụ:**

Nhập từ bàn phím	Xuất ra màn hình
10 7	6

## CHUYÊN ĐỀ 2. NGÔN NGỮ LẬP TRÌNH C++ BÀI 4. KIỂU TẬP – KIỂU MẢNG

### 1. Kiểu tập văn bản:

- Dữ liệu ghi trong tệp văn bản được lưu trữ lâu dài. Dung lượng tệp phụ thuộc vào bộ nhớ ngoài để ghi tệp.

- Tệp văn bản được bố trí thành dòng và cột, khi mở tệp con trỏ tệp ở dòng 1 cột 1. Trong C++ đọc dữ liệu từ tệp hay ghi dữ liệu vào tệp được thực hiện một cách tuần tự. Từ đầu tệp (1,1) đến cuối tệp (eof).

- Trong C++ có nhiều hàm hỗ trợ để đọc/ghi tệp văn bản và cần quan tâm đến cấu trúc tệp để đọc/ghi cho đúng yêu cầu. Hai cách thường dùng là:

**Cách 1: Dùng lệnh chuyển hướng:** (tệp được lưu ở thư mục hiện hành):

- Cho phép đọc dữ liệu từ tệp văn bản: **freopen("Tên tệp", "r", stdin);**

- Cho phép xuất dữ liệu vào tệp văn bản: **freopen("Tên tệp", "w", stdout);**

Lệnh **cin** sẽ đọc dữ liệu từ tệp, mỗi lần đọc con trỏ tệp tự dịch chuyển sang phải một phần tử (mỗi phần tử cách nhau bởi kí tự trống hoặc kí tự xuống dòng).

Lệnh **cout** sẽ ghi dữ liệu vào tệp.

**Cách 2: Dùng hàm trong thư viện stream:**

Khai báo tệp **f** chỉ cho phép đọc dữ liệu : **ifstream f("tên tệp");**

Khai báo tệp **g** chỉ cho phép ghi dữ liệu vào tệp: **ofstream g("tên tệp");**

+ Đọc dữ liệu từ tệp dùng cú pháp: **f>>tên biến;**

+ Xuất dữ liệu vào tệp dùng cú pháp: **g<<biểu thức**



Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

+ Đóng tệp để lưu dữ liệu: f.close(); g.close();

Chú ý: Nếu sử dụng hàm ifstream/ ofstream ta phải đóng tệp và thư viện này có hàm eof() kiểm tra con trỏ tệp ở cuối tệp hay chưa (f.eof() con trỏ ở cuối tệp f)

Ví dụ: - Đọc tệp **DAYSO.INP** có cấu trúc:

Dòng thứ nhất ghi ba số nguyên **x, y, z** ( $0 < x, y, z < 10^5$ );

Dòng thứ hai ghi **hai số nguyên** số a, b ( $0 < a, b < 10^5$ )

- Xuất ra tệp: **DAYSO.OUT**

Dòng thứ nhất: xuất ra giá trị lớn nhất trong ba số x, y, z

Dòng thứ hai ghi số  $c = a * b$

Ví dụ:

DOCSO.INP	DOCSO.OUT
3 5 4	5
2 6	12

**Cách 1:**

```
#include <bits/stdc++.h>
using namespace std;
int a, b;
long long c;
int x, y, z;
main(){
    freopen("DOCSO.INP","r", stdin);
    freopen("DOCSO.OUT","w", stdout);
    cin>>x>>y>>z;
    cin>>a>>b;
    c=a*b;
    cout<<max(max(x,y),z)<<endl;
    cout<<c;
}
```

**Cách 2:**

```
#include <bits/stdc++.h>
using namespace std;
int a, b;
long long c;
int x, y, z;
main(){
    ifstream f("DOCSO.INP");
    ofstream g("DOCSO.OUT");
    f>>x>>y>>z;
    f>>a>>b;
    c=a*b;
    g<<max(max(x,y),z)<<endl;
    g<<c;
    f.close();
    g.close();
}
```

**Chú ý:**

- Cách 1 thường sử dụng, để đọc từ bàn phím và xuất ra màn hình, ta chỉ cần khóa dòng `//freopen("DOCSO.INP","r", stdin);` và dòng `//freopen("DOCSO.OUT","w", stdout);`;

- Thời gian 1 giây lệnh cin/cout trong C/C++ đọc được  $10^5$  số. Để đọc được  $10^6$  số ta cần bổ sung dòng lệnh `ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);` trong hàm main() hoặc dùng lệnh nhập/xuất scanf/ printf

Lệnh	Ý nghĩa
scanf("%d", &n);	Đọc số nguyên n kiểu int
scanf("%lld", &n);	Đọc số nguyên n kiểu long long
scanf("%f", &n);	Đọc số thực n kiểu float/double
scanf("%s", &n);	Đọc chuỗi n kiểu string
scanf("%c", &n);	Đọc ký tự n kiểu char
printf("%d", n);	Xuất giá trị n kiểu int
printf("%lld", n);	Xuất giá trị n kiểu long long
printf("%s", n);	Xuất giá trị chuỗi n kiểu string



<code>printf("%.1f", n);</code>	Xuất giá trị n kiểu thực và lấy 1 chữ số thập phân
---------------------------------	--

**2. Kiểu mảng:** Là tập hợp các phần tử cùng kiểu dữ liệu với độ dài cố định. Để khai báo nhiều biến cùng kiểu dữ liệu ta thường dùng mảng. Chẳng hạn để quản lí điểm trung bình của học sinh trong lớp có 35 học sinh, thay vì ta dùng 35 biến lần lượt là  $x_1, x_2, \dots, x_{35}$  (còn gọi biến đơn trị) ta dùng mảng  $a[1], a[2], \dots, a[35]$  (biến đa trị) thuận lợi hơn.

**a- Khai báo biến mảng:**

- Mảng 1 chiều: <Kiểu dữ liệu> <Tên mảng>[Số lượng phần tử];
- Mảng 2 chiều: <Kiểu dữ liệu> <Tên mảng>[Số lượng dòng][Số lượng cột];

**b- Truy xuất biến mảng:**

- Mảng 1 chiều: <Tên biến mảng>[chỉ số];
- Mảng 2 chiều: <Tên biến mảng>[chỉ số dòng][chỉ số cột]

Ví dụ:

`int a[35];` Mảng một chiều có tên là **a** có 36 phần tử, kiểu dữ liệu các phần tử nhận là số nguyên. Có tên truy xuất là:  $a[0], a[1], \dots, a[35]$ ;

`int b[3][4];` Mảng hai chiều có tên là **b** (còn gọi là ma trận), có 3 hàng, 4 cột, có 20 phần tử, kiểu dữ liệu các phần tử là số nguyên. Có tên truy xuất là:  $a[0][0], a[0][1], a[0][2], a[0][3]; a[1][0], a[1][1], a[1][2], \dots, a[3][4]$

**c- Một số hàm thông dụng trong mảng:**

- Hàm: `sort(a+1, a+1+n);` Sắp xếp tăng các giá trị mảng a từ phần tử 1 đến phần tử n –  $O(n \log(n))$ .
- `sort(a+1, a+1+n, greater<int>());` Sắp xếp giảm  $O(n \log(n))$
- Hàm `*max_element(a+1, a+1+n);` Trả về giá trị lớn nhất của mảng a.
- Hàm `*min_element(a+1, a+1+n);` Trả về giá trị nhỏ nhất của mảng a.
- Hàm `memset(a, 0, sizeof(a));` Khởi tạo mảng a các phần tử có giá trị = 0
- Hàm `memcpy(b, a, sizeof(int)*n);` mảng b chép n phần tử của mảng a (bắt đầu 0)

Ví dụ: Viết chương trình sắp xếp một dãy số có **n** phần tử  $a_1, a_2, \dots, a_n$  các phần tử có kiểu số nguyên ( $1 < n \leq 10^5$ ).

**Dữ liệu vào:** Cho từ tệp văn bản DAYSO.INP gồm:

- Dòng thứ nhất ghi số nguyên n.
- Dòng thứ hai ghi n số nguyên  $a_1, a_2, \dots, a_n$

**Kết quả:** Xuất vào tệp văn bản DAYSO.OUT chỉ có một dòng ghi dãy số đã sắp xếp.

**Ví dụ:**

DAYSO.INP	DAYSO.OUT
5	2 3 3 5 6
3 2 3 6 5	

<code>#include &lt;bits/stdc++.h&gt;</code> <code>using namespace std;</code>	
--	--



Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu

<pre>const int N=1e5+5; int a[N]; int n; main(){     freopen("DAYSO.INP", "r", stdin);     freopen("DAYSO.OUT", "w", stdout);     cin&gt;&gt;n;     for(int i=1; i&lt;=n; i++) cin&gt;&gt;a[i];     sort(a+1, a+n+1);      for(int i=1; i&lt;=n; i++) cout&lt;&lt;a[i]&lt;&lt;" "; }</pre>	<pre>//khai báo hằng số N=10<sup>5</sup>+5 //khai báo mảng a có 10<sup>5</sup>+5 phần tử  //mở tệp chỉ cho phép đọc dữ liệu //mở tệp chỉ cho phép ghi dữ liệu //Đọc giá trị cho biến n //Đọc n giá trị cho mảng a //sắp xếp mảng a  // Xuất giá trị mảng a</pre>
--	--

**Ghi chú quan trọng:**

- Mảng trong C++ được đánh chỉ số bắt đầu từ 0. Mỗi phần tử của mảng lưu một giá trị. Để lưu được nhiều giá trị ta cần gộp kiểu dữ liệu bằng từ khóa **typedef pair**

Ví dụ: **typedef pair <int, int > ii;**

ii a[100]; // mảng a có 101 phần tử, mỗi phần tử lưu hai số nguyên. Có tên truy xuất tổng quát là a[i].first và a[i].second; (i=0..100)

- Số lượng phần tử của mảng 1 chiều không quá  $10^8$ ; Số lượng phần tử của mảng hai chiều: số dòng x số cột cũng không quá  $10^8$

- Để đọc dữ liệu trong tệp thì tệp đó phải có trong thư mục hiện hành.

- Để đọc dữ liệu cho đến khi cuối tệp ta dùng **while( cin>>x)**

- Để đọc xâu có kí tự trắng dùng hàm **getline(f, s)** hoặc **getline(cin, s)**; (s là biến xâu, f là biến tệp).



**BÀI TẬP 5**  
**KIỂU TỆP - KIỂU MẢNG**

**Bài 1. TÌM SỐ LỚN THỨ HAI (Đề thi GV dạy giỏi THPT 2016 – 2017)**

Cho dãy gồm **n** số nguyên **a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>n</sub>** đôi một khác nhau.

**Yêu cầu:** Tìm số lớn thứ hai của dãy số đã cho.

**Dữ liệu vào:** cho từ tệp văn bản MAX2.INP

- dòng thứ nhất ghi số nguyên **n** ( $1 < n \leq 10^6$ );

- dòng thứ hai ghi **n** số nguyên **a<sub>1</sub>, a<sub>2</sub>, ..a<sub>n</sub>** ( $0 < a[i] \leq 10^6$ ;  $i=1..n$ )

**Kết quả:** Ghi ra tệp MAX2.OUT chỉ có 1 dòng ghi kết quả theo yêu cầu

**Ví dụ:**

MAX2.INP	MAX2.OUT
5	7
8 5 7 6 2	

**Ràng buộc:** 30% số test tương ứng 30% số điểm của bài có  $1 < n \leq 10^3$

40% số test tương ứng 40% số điểm của bài có  $10^3 < n \leq 10^5$

30% số test tương ứng 30% số điểm của bài có  $10^5 < n \leq 10^6$

**Bài 2. KHOẢNG CÁCH SỐ**

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

Với dãy số hữu hạn  $a_1, a_2, \dots, a_n$  ta định nghĩa khoảng cách  $k$  giữa hai số  $a_i$  và  $a_j$  là:  
 $k = |a_i - a_j|$  ( $1 < i, j \leq n$ ;  $i \neq j$ ).

**Yêu cầu:** Tính khoảng cách  $k$  lớn nhất giữa hai số trong dãy số đã cho.

**Dữ liệu vào:** cho từ tệp văn bản KC.INP

- dòng thứ nhất ghi số nguyên  $n$  ( $1 < n \leq 10^6$ );

- dòng thứ hai ghi  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $|a[i]| < 100$ ;  $i=1..n$ )

**Kết quả:** Ghi ra tệp KC.OUT chỉ có 1 dòng ghi kết quả theo yêu cầu

**Ví dụ:**

KC.INP	KC.OUT
7	9
1 -1 2 8 5 5 4	

**Giải thích**

Hai số -1, 8 có  $|-1-8| = 9$  là khoảng cách lớn nhất

Ràng buộc: 70% số test tương ứng 70% số điểm của bài có  $1 < n \leq 10^5$

30% số test tương ứng 30% số điểm của bài có  $10^5 < n \leq 10^6$

### Bài 3: MA TRẬN SỐ

Cho ma trận vuông cấp  $n$ , mỗi ô có một số nguyên.

**Yêu cầu:** Hãy tính tổng các số trên đường chéo chính và đường chéo phụ của ma trận

**Input:** MATRAN.INP gồm:

- Dòng thứ nhất: ghi hai số nguyên dương  $n$  ( $n \leq 10^4$ )

- Tiếp theo là  $n$  dòng, mỗi dòng ghi  $n$  số nguyên  $a_{ij}$  ( $|a_{ij}| \leq 10^9$ )

**Output:** MATRAN.OUT gồm:

- Dòng thứ nhất ghi số nguyên là tổng các số trên đường chéo chính.

- Dòng thứ hai ghi số nguyên là tổng các số trên đường chéo phụ.

**Ví dụ:**

MATRAN.OUT	MATRAN.OUT
3	15
1 2 3	15
4 5 6	
7 8 9	

### Bài 4: BẢNG SỐ

Cho một bảng gồm  $n$  hàng và  $n$  cột, được đánh số cột từ 1 đến  $n$  từ trái qua phải và đánh số dòng từ 1 đến  $n$  từ trên xuống dưới. Các ô ở dòng thứ  $i$  và cột thứ  $j$  có giá trị là  $i \times j$ .

**Yêu cầu:** Cho một số nguyên  $x$ . Hãy đếm số lượng ô trong bảng chứa số nguyên  $x$ .

**Dữ liệu vào:** Cho trong tệp văn bản BANGSO.INP chỉ có một dòng chứa hai số nguyên  $n$  và  $x$  ( $1 \leq n \leq 10^6$ ;  $1 \leq x \leq 10^9$ ).

**Kết quả:** Xuất ra tệp văn bản BANGSO.OUT chỉ có một dòng chứa số nguyên là số lần số  $x$  xuất hiện trong bảng.

**Ví dụ:**

BANGSO.INP	BANGSO.OUT	Minh họa
------------	------------	----------

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu

6 12	4	<div><div><div><div><div><div>1</div><div>2</div><div>3</div><div>4</div><div>5</div><div>6</div></div><div><div>2</div><div>4</div><div>6</div><div>8</div><div>10</div><div><u>12</u></div></div><div><div>3</div><div>6</div><div>9</div><div><u>12</u></div><div>15</div><div>18</div></div><div><div>4</div><div>8</div><div><u>12</u></div><div>16</div><div>20</div><div>24</div></div><div><div>5</div><div>10</div><div>15</div><div>20</div><div>25</div><div>30</div></div><div><div>6</div><div><u>12</u></div><div>18</div><div>24</div><div>30</div><div>36</div></div></div></div></div><div>Số 12 xuất hiện 4 lần trong bảng</div></div>
5 13	0	

**Ràng buộc:** - Có 70% số test tương ứng 70% số điểm có  $1 \leq n \leq 10^3$   
 - Có 30% số test tương ứng 30% số điểm có  $10^3 < n \leq 10^6$

## Bài 5. BỐC BÀI

Trò chơi Blackjack (Xì dách) sử dụng bộ bài 52 quân. Các quân bài đánh số từ 2, 3, ..., 9, 10, J, Q, K và A. Với mỗi số đánh dấu có 4 quân bài (Cơ, Rô, Chuồn, Bích). Các quân J, Q, K có giá trị là 10, quân A có giá trị 11 điểm (theo luật quốc tế không tính 1 điểm), các quân còn lại có số điểm bằng số ghi trên quân bài.

Người chơi cần giữ trên tay các quân bài với tổng số điểm gần 21 nhất, nhưng không vượt quá 21 điểm (vượt 21 điểm gọi là quắc). Đến lượt mình đi, người chơi có thể bốc thêm quân bài mới (DRAW) hoặc không bốc thêm quân bài (STOP).

Bé Sen đang giữ  $n$  quân bài trên tay, quân  $i$  có  $p_i$  điểm ( $i=1..n$ ). Nếu khả năng bốc thêm quân bài mới mà tổng điểm các quân bài trên tay vẫn không vượt quá 21 điểm thì bé Sen bốc bài, trong trường hợp ngược lại thì bé không bốc bài.

**Yêu cầu:** Hãy xác định ở nước này bé Sen nên bốc bài hay bỏ qua và đưa ra thông báo tương ứng (DRAW hoặc STOP)

**Dữ liệu vào:** G21.INP

- Dòng đầu tiên chứa số nguyên  $n$  ( $1 < n \leq 52$ )
  - Dòng thứ  $i$  trong  $n$  dòng sau chứa số nguyên  $p_i$  ( $2 \leq p_i \leq 11$ )
- Dữ liệu đảm bảo hợp lệ

**Kết quả:** Đưa ra tệp G21.OUT thông báo xác định được

**Ví dụ:**

G21.INP	G21.OUT
2 6 5	DRAW

## CHUYÊN ĐỀ 2. NGÔN NGỮ LẬP TRÌNH C++ Bài 5. XÂU KÍ TỰ

**1. Xâu kí tự:** Là tập hợp các kí tự trong bảng mã ASCII. Độ dài xâu là số lượng các kí tự trong xâu. (Trong C++ độ dài xâu không quá  $10^9$  kí tự).

**a- Khai báo biến xâu:**

**string <Tên biến chuỗi>;**

Chuỗi là một mảng ký tự, được đánh chỉ số bắt đầu từ 0. Để dễ xử lý ta gán ký trắng trước chuỗi.

Ví dụ: string s; s="TIN HOC"; s=" "+s; thì s[0]=' '; s[1]='T', s[2]='I', ...,s[7]='C';

### **b- Đọc xuất chuỗi từ tệp:**

Nếu chuỗi không có khoảng trắng ta dùng cin/cout bình thường.

Nếu chuỗi có ký tự trắng ta dùng hàm **getline(cin, biến chuỗi)** hoặc **getline(biến tệp, biến chuỗi)**

### **c- Các phép toán trên chuỗi:**

- Cộng(+): "A" + "B" = "AB"

- Bằng (==): Hai chuỗi bằng nhau nếu chúng giống nhau hoàn toàn.

- Lớn (>) Chuỗi A lớn hơn chuỗi B nếu ký tự đầu tiên khác nhau giữa chúng có mã Ascii của chuỗi A lớn hơn mã Ascii của chuỗi B.

- Toán tử: != (khác); >=(Lớn hơn bằng); <= (nhỏ hơn bằng); = (phép gán)

Ví dụ: "HOA"+"HONG"="HOAHONG"; "LOP9">"LOP10"

### **2- Hàm/thủ tục xử lý chuỗi:**

- Hàm s.size() : Độ dài chuỗi s, tương tự hàm s.length();

- Hàm s.substr(x,y): Sao chép chuỗi con của chuỗi s từ vị trí x, số lượng ký tự là y

- Thủ tục: reverse(s.begin(), s.end()): Đảo ngược chuỗi s

- Thủ tục s.insert(x,r): Chèn chuỗi r vào chuỗi s tại vị trí x

- Thủ tục s.erase(x,y): Xóa chuỗi con của chuỗi s tại vị trí x, số ký tự xóa là y.

- s.empty(): kiểm tra chuỗi rỗng

- s.push\_back():chèn ký tự vào sau chuỗi s

- s.replace(x,n,r): thay thế chuỗi s từ vị trí "x", số phần tử thay thế là "n" và thay thế bằng chuỗi r.

- s.swap(r): đổi giá trị 2 chuỗi cho nhau.

**Ví dụ:** Một chuỗi ký tự gọi là đối xứng nếu nó không thay đổi khi ta viết các ký tự trong chuỗi theo thứ tự ngược lại (gọi là chuỗi đảo). Ví dụ các chuỗi 'tam mat', 'madam' là các chuỗi đối xứng, vì khi viết ngược lại ta cũng thu được 'tam mat', 'madam'. Chuỗi 'sa dec' là chuỗi không đối xứng vì khi viết ngược lại ta thu được là: 'ced as'.

**Yêu cầu:** Kiểm tra chuỗi s có phải là chuỗi đối xứng hay không? Nếu chuỗi s là chuỗi đối xứng xuất ra chữ YES ngược lại xuất ra chữ NO

**Input:** XAU.INP chỉ có một dòng ghi chuỗi s có độ dài không quá 200

**Output:** XAUDAO.INP ghi YES nếu chuỗi s là chuỗi đối xứng, ngược lại ghi "NO"

**Ví dụ:**

XAU.INP	XAU.INP
Tam mat	YES
sa dec	NO

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

```
using namespace std;
```

```
string s, r;
```

```
main(){
```

```
freopen("XAU.INP", "r", stdin);
```

**Khai báo hai biến chuỗi s và r**



```
freopen("XAU.OUT", "w", stdout);  
getline(cin, s);  
r=s;  
reverse(s.begin(), s.end());  
if(s==r) cout<<"YES";  
else cout<<"NO";  
}
```

// đọc xâu s có khoảng trắng  
Gán xâu r bằng xâu s  
Tạo xâu s đảo ngược  
So sánh xâu s và xâu r

## BÀI TẬP 6 XÂU KÍ TỰ

### Bài 1. TÁCH HỌ TÊN

Cho họ và tên học sinh là một xâu, có hai từ trở lên, giữa các từ ngăn cách nhau chỉ một kí tự trắng và cũng không có kí tự trắng nào ở đầu xâu và ở cuối xâu. Từ thứ nhất là xâu họ, từ cuối cùng là xâu tên của học sinh.

**Yêu cầu:** Tách họ, tên học sinh.

**Input:** HOTEN.INP chỉ có một dòng ghi xâu họ và tên của học sinh

**Output:** HOTEN.OUT

Dòng thứ nhất ghi xâu họ  
Dòng thứ hai ghi xâu tên

**Ví dụ :**

HOTEN.INP	HOTEN.OUT
Le Van Tam	Le Tam



### Bài 2: HIỆU LỆNH

Nam là học sinh thực hiện tốt các hiệu lệnh đội hình, đội ngũ. Trong giờ học môn Giáo dục Quốc phòng, Nam được tin tưởng giao nhiệm vụ làm mẫu thực hiện các hiệu lệnh của thầy. Hiệu lệnh của thầy như sau: Nam đứng nghiêm, nếu thầy hô “trái” thì Nam dịch chuyển sang trái một mét, nếu thầy hô “phải” thì Nam dịch chuyển sang phải một mét.

**Yêu cầu:** Sau  $n$  lần thầy hô hiệu lệnh Nam cách vị trí ban đầu là bao nhiêu mét?

**Dữ liệu vào:** Cho từ tệp văn bản HIEULENH.INP có một dòng duy nhất ghi xâu  $S$  gồm  $n$  kí tự ( $1 \leq n \leq 255$ ) là xâu biểu diễn hiệu lệnh của thầy, xâu chỉ có các kí tự “T” và “P”. Nếu là kí tự “T” là thầy hô “trái”, nếu là kí tự “P” là thầy hô “phải”.

**Kết quả:** Ghi vào tệp văn bản HIEULENH.OUT có một số nguyên dương duy nhất là khoảng cách tính được.

**Ví dụ:**

HIEULENH.INP	HIEULENH.OUT
TTT	3
TTPT	2
PPTTT	1

### Bài 3. TEEN (HSG Đồng Tháp 2016)

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

Bé Sen học rất giỏi môn tiếng Anh và bé rất thích từ ‘TEEN’. Khi gặp một xâu kí tự, bé Sen luôn tự hỏi từ các kí tự trong xâu đó có thể ghép được nhiều nhất bao nhiêu từ ‘TEEN’.

- **Yêu cầu:** Cho trước xâu *s* trong tệp TEEN.INP. Xâu *s* chỉ gồm các chữ cái in hoa và kí tự trắng, có độ dài tối đa 255 kí tự. Xuất ra tệp TEEN.OUT: số lượng kí tự ‘T’, số lượng kí tự ‘E’, số lượng kí tự ‘N’ và số lượng từ ‘TEEN’ nhiều nhất có thể tạo thành từ các kí tự của xâu *s* đã cho.

- **Ví dụ:**

TEEN.INP	TEEN.OUT
CONNECT CONNECT EEET	3 5 4 2

#### Bài 4: XẾP HÀNG (Đề HSG 2014)

Trong giờ học thể dục, thầy giáo xếp *n* học sinh của lớp thành một hàng và vị trí của các học sinh được đánh số từ 1 đến *n* từ trái sang phải. Ban đầu các học sinh đứng tùy ý trong hàng. Tuy nhiên, để tôn trọng các bạn nữ, thầy muốn các bạn nam không được đứng liền trước bạn nữ nào (đứng liền trước ở đây được hiểu rằng vị trí của các bạn nam là *i* và vị trí của các bạn nữ là *i*+1). Để thực hiện quy định này, thầy bắt đầu đi từ đầu hàng đến cuối hàng, khi gặp bạn nam nào đó đứng liền trước một bạn nữ, thầy sẽ yêu cầu bạn nam này đổi chỗ cho bạn nữ rồi đi tiếp đến các bạn sau đó. Chú ý rằng trong một lượt sắp xếp, thầy chỉ đi theo một chiều và mỗi bạn nam sẽ chỉ được đổi chỗ một lần. Tất nhiên là chỉ đi theo một chiều và mỗi bạn nam chỉ được đổi chỗ một lần. Tất nhiên là chỉ lượt sắp xếp như vậy thì vẫn có thể có nhiều vị trí mà bạn nam đứng trước nữ xuất hiện thêm nên cần phải làm đi làm lại thao tác sắp xếp này nhiều lần.

**Yêu cầu:** Cho hai số nguyên dương *n* với  $0 < n \leq 50$  và một dãy kí tự *G* và *B*, trong đó *G* là kí hiệu bạn nữ và *B* là kí hiệu bạn nam thể hiện vị trí các học sinh của lớp ban đầu. Hỏi sau bao nhiêu lần thao tác thì thầy giáo sẽ hoàn tất việc sắp xếp này?

**Dữ liệu vào:** Trong file XEPHANG.INP gồm có: Dòng thứ nhất chứa số nguyên dương *n*, dòng thứ hai gồm một dãy *n* kí tự ‘*G*’ và ‘*B*’ biểu thị vị trí đứng của các học sinh trong hàng (từ trái qua phải tương ứng với chỉ số vị trí tăng dần).

**Kết quả:** In ra file XEPHANG.OUT chỉ một dòng chứa số nguyên cho biết số lần thầy giáo cần sắp xếp.

**Ví dụ:**

XEPHANG.INP	XEPHANG.OUT
5 BGGBG	3
5 BGGBG	3
4 GGGB	0

#### Bài 5: XẾP HÀNG

Trong buổi lễ đón học sinh đầu cấp ở một trường Trung học phổ thông, các học sinh khối 10 sẽ được xếp thành một hàng dài theo đơn vị lớp để đi diễu hành qua lễ đài. Mỗi em học sinh của lớp được ban tổ chức đánh mã hiệu là một chữ cái Latinh in hoa. Chẳng hạn: lớp 10A mỗi

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

em được ban tổ chức đánh mã hiệu là A, lớp 10B mỗi em được ban tổ chức đánh mã hiệu là B, .... Để buổi diễu hành diễn ra đẹp mắt, ban tổ chức yêu cầu: lớp có số lượng học sinh nhiều hơn sẽ xếp trước, nếu có số lượng học sinh bằng nhau thì xếp mã hiệu theo thứ tự từ A đến Z.

**Yêu cầu:** Cho trước số nguyên  $n$  là số học sinh của khối 10 đi diễu hành, các mã hiệu của ban tổ chức. Hãy thực hiện sắp xếp theo yêu cầu của ban tổ chức.

**Dữ liệu vào:** Cho trong tệp văn bản XEPHANG.INP gồm:

- Dòng thứ nhất chứa số nguyên  $n$  là số học sinh khối 10 ( $1 \leq n \leq 10^3$ )
- Dòng thứ hai chứa mã hiệu của các học sinh khối 10. Biết rằng các mã hiệu chỉ nằm trong tập chữ cái Latinh từ A đến Z

**Kết quả:** Ghi vào tệp văn bản XEPHANG.OUT chỉ có một dòng ghi từ trái sang phải các kí hiệu là thứ tự xếp hàng theo yêu cầu của ban tổ chức.

**Ví dụ:**

XEPHANG.INP	XEPHANG.OUT
8 BBAABBBA	BBBBBAAA
7 AACCCBB	CCCAABB



## CHUYÊN ĐỀ 2. NGÔN NGỮ LẬP TRÌNH C++ BÀI 6. KIỂU BẢN GHI

**1. Bản ghi (record):** là một tập hợp bao gồm các vùng tin (field), còn được gọi là trường, các vùng tin phải có tên khác nhau và cũng có thể thuộc các kiểu dữ liệu khác nhau.

### a. Khai báo kiểu bản ghi:

```
struct <Tên kiểu bản ghi>{  
    <kiểu dữ liệu> tên trường 1;  
    <kiểu dữ liệu> tên trường 2;  
    <kiểu dữ liệu> tên trường n;  
};
```

Sau khi định nghĩa kiểu bản ghi ta có thể khai báo các biến có kiểu bản ghi. nên khai báo struct ở vị trí toàn cục chương trình

### b. Truy xuất bản ghi:

<tên biến bản ghi>.<Tên trường>

Ví dụ: struct hs{  
 string hoten;  
 int namsinh;  
};  
hs a[50]; // mảng bản ghi có 50 phần tử  
a[1].hoten="HTT";  
t.namsinh=1968;  
cout<<t.namsinh;

**Ví dụ:** Đọc tệp hocsinh.inp: dòng thứ nhất ghi số nguyên **n**, tiếp theo **n** dòng, mỗi dòng ghi 2 thông tin: Tên và tuổi học sinh. Sắp xếp theo tên và xuất kết quả vào tệp hocsinh.out

<pre>#include &lt;bits/stdc++.h&gt; using namespace std; const int N=100; struct hocsinh{     string ten;     int tuoi; }; hocsinh a[N]; int n; bool sosanh(hocsinh u, hocsinh v){     return u.ten&lt;v.ten; } main(){     freopen("hocsinh.inp", "r", stdin);     freopen("hocsinh.out", "w", stdout);     cin&gt;&gt;n;     for(int i=1; i&lt;=n; i++)         cin&gt;&gt;a[i].ten&gt;&gt;a[i].tuoi;      sort(a+1, a+1+n, sosanh);</pre>	<p>Khai báo bản ghi có hai trường: ten và tuoi.</p> <p>Hàm so sánh tên học sinh dùng trong sắp xếp.</p> <p>Đọc từ tệp đưa vào mảng bản ghi</p> <p>Sắp xếp theo tên</p>
--	--

<pre>for(int i=1; i&lt;=n; i++){     cout&lt;&lt;a[i].ten;     cout&lt;&lt;" "&lt;&lt;a[i].tuoi&lt;&lt;endl; } }</pre>	Xuất mảng bảng ghi
--	--------------------

## CHUYÊN ĐỀ 2. NGÔN NGỮ LẬP TRÌNH C++ Bài 7- CHƯƠNG TRÌNH CON – ĐỆ QUY

**1- Chương trình con:** là một chương trình thực hiện một công việc hay một chức năng đơn giản và nó có thể được gọi bởi một chương trình khác.

Chương trình con được chia làm hai loại: Hàm và thủ tục.

### a- Khai báo CTC:

#### + Hàm:

```
<kiểu dữ liệu> <Tên hàm> (Tham số hình thức){
    Các lệnh;
    return <giá trị trả về của hàm>;
}
```

#### + Thủ tục:

```
void <Tên thủ tục> (Tham số hình thức){
    Các lệnh;
}
```

### b- Lời gọi CTC:

- Chương trình con có thể gọi từ một chương trình con khác gọi nó hoặc gọi từ chương trình chính.

Lời gọi CTC: <Tên CTC>(Tham số thật sự);

Ví dụ: Chương trình con tính diện tích hình chữ nhật và được gọi từ chương trình chính

```
#include <bits/stdc++.h>
using namespace std;
int dientich(int a, int b){
    return a*b;
}
int a, b;
main(){
    cin>>a>>b;
    cout<<dientich(a, b);
}
```

**Ghi chú:** Lời gọi của hàm luôn có giá trị trả về, lời gọi thủ tục giống như cách viết một câu lệnh.

**2- Đệ quy:** Chương trình con nếu nó gọi chính nó gọi là đệ quy. (Hàm main() không được gọi chính nó. Một hàm có thể gọi đệ quy nếu có:

+ Mệnh đề cơ sở (điểm neo/ điểm để dừng CTC)

+Mệnh đề đệ quy.

- Ví dụ: Chương trình tính  $n!$  được thiết kế bằng kỹ thuật đệ quy.

<pre>#include &lt;bits/stdc++.h&gt; using namespace std; int n, kq; int p(int n){     if(n==1) kq=1;     else kq=n*p(n-1); //de qui return kq; } main(){     cin&gt;&gt;n;     cout&lt;&lt;p(n); }</pre>	<p>//diem neo // lời gọi hàm p(), nó gọi chính(đệ qui)</p> <p>Lời gọi hàm p() ở chương trình chính</p>
--	--

## BÀI TẬP 7 – CHƯƠNG TRÌNH CON – KIỂU BẢN GHI

### Bài 1. DÃY SỐ FIBONACCI (Yêu cầu cài đặt CTC đệ quy)

**Yêu cầu:** Tính số  $F_n$  của dãy Fibonacci bằng kỹ thuật đệ qui. Biết  $F(1)=1$ ;  $F(2)=1$ ;  $F(i)=F(i-1)+F(i-2)$  ( $i=1..n$ )

**Input:** Fibo.INP chỉ có một dòng ghi số nguyên  $n$  ( $1 < n \leq 40$ )

**Output:** Fibo.OUT chỉ có 1 dòng ghi số  $F_n$  tính được.

**Ví dụ:**

FIBO.INP	FIBO.OUT
6	8

### Bài 2. BỘI CHUNG – ƯỚC CHUNG (Yêu cầu cài đặt CTC đệ quy)

Bội chung nhỏ nhất của hai số nguyên  $a$  và  $b$  là số nguyên dương nhỏ nhất chia hết cho cả  $a$  và  $b$ .  $BCNN(a,b) = a.b/UCLN(a,b)$ .

**Yêu cầu:** Cho hai số nguyên dương  $a, b$ . Tính  $UCLN(a, b)$  và  $BCNN(a,b)$

**Input:** UCBC.INP chỉ có 1 dòng ghi hai số nguyên  $a, b$

**Output:** UCBC.OUT dòng 1 ghi  $UCLN(a,b)$ . Dòng 2 ghi  $BCNN(a,b)$

**Ví dụ:**

UCBC.INP	UCBC.OUT
6 8	2 24

### Bài 3: NỘP THUẾ (Yêu cầu cài đặt có CTC)

Ông Nam hiện sống trong một quốc gia có luật thuế hết sức đặc biệt. Tổng số tiền thu nhập của ông Nam là  $n$  đô-la ( $n \geq 2$ ) và ông ta phải nộp thuế thu nhập cá nhân với số tiền là ước số lớn nhất của  $n$  (tất nhiên không thể bằng  $n$ ). Chẳng hạn nếu  $n=6$  thì ông phải nộp thuế là 3 đô-la, trong khi nếu  $n=25$  thì phải nộp 5 đô-la và nếu  $n=2$  thì chỉ nộp 1 đô-la.

Tuy nhiên, ông Nam là một người rất cơ hội, ông ta muốn ăn gian tiền nộp thuế. Chính vì vậy ông Nam đã chia số tiền ban đầu là  $n$  ra thành một số phần  $n_1 + n_2 + \dots + n_k = n$  ( $k$  là tùy ý, thậm chí bằng 1), và nộp thuế cho từng phần riêng biệt. Mặc dù vậy ông ta không thể chia thành những phần bằng 1 vì như vậy sẽ bị lộ bí mật, do đó  $n_i \geq 2$  với  $i=1..k$ .

Vì muốn số tiền thuế phải nộp là ít nhất nên ông Nam đã tính mãi mà chưa tìm được cách tối ưu. Bạn là một nhà lập trình viên hãy lập trình giúp ông Nam tìm một cách chia.





**Yêu cầu:** Tính số tiền thuế phải nộp ít nhất.

**Dữ liệu vào:** TAXES.INP gồm một dòng ghi một số nguyên  $n$  ( $2 \leq n \leq 2 \cdot 10^9$ ).

**Kết quả:** TAXES.OUT một số nguyên là số tiền thuế ít nhất mà ông Nam phải nộp.

**Ví dụ:**

TAXES.INP	TAXES.OUT	TAXES.INP	TAXES.OUT
4	2	27	3

#### Bài 4. Nhiếp ảnh gia (Yêu cầu cài đặt bằng bảng ghi)

Jack là một nhiếp ảnh gia, máy ảnh mới của anh có thể nhớ dung lượng là  $d$  megabytes (MB). Tùy mức độ điều chỉnh máy ảnh mà anh ấy có thể chụp những bức ảnh có chất lượng cao hay thấp, một bức ảnh có chất lượng thấp sẽ chiếm dung lượng là  $a$  MB của không gian trống trong thẻ nhớ, một bức ảnh có chất lượng cao sẽ chiếm dung lượng là  $b$  MB của không gian trống trong thẻ nhớ.

Có  $n$  khách hàng đến chụp ảnh (đánh số thứ tự từ 1 đến  $n$ ), khách hàng thứ  $i$  yêu cầu chụp cho anh ta  $x_i$  bức ảnh chất lượng thấp và  $y_i$  bức ảnh chất lượng cao.

**Yêu cầu:** Hãy giúp nhà nhiếp ảnh gia của chúng ta đưa ra phương án hợp lý để có thể phục vụ được nhiều khách hàng nhất.

**INPUT: PHOTO.INP**

- Dòng 1 chứa hai số nguyên dương  $n$  và  $d$  ( $1 \leq n \leq 10^5$ ,  $1 \leq d \leq 10^9$ )
- Dòng 2 chứa hai số nguyên dương  $a$  và  $b$  ( $1 \leq a \leq b \leq 10^4$ )
- $n$  dòng tiếp theo, dòng thứ  $i$  chứa hai số  $x_i$  và  $y_i$  ( $0 \leq x_i, y_i \leq 10^5$ ) – Số lượng bức ảnh chất lượng thấp và số lượng bức ảnh chất lượng cao của khách hàng thứ  $i$  yêu cầu.

**OUTPUT: PHOTO.OUT** Chỉ có một dòng chứa số  $k$  – số lượng khách hàng được phục vụ nhiều nhất

**Ví dụ:**

PHOTO.INP	PHOTO.OUT
3 10 2 3 1 4 2 1 1 0	2

#### BÀI 5. MUA BÌ (Yêu cầu cài đặt bằng bảng ghi)

Bé Sen đi mua bì ở siêu thị. Siêu thị có  $m$  loại bì với màu khác nhau, loại màu  $i$  siêu thị có  $a_i$  hộp, mỗi hộp chứa  $b_i$  viên bì. Giá mỗi hộp là như nhau và Sen đủ tiền mua  $n$  hộp. Sen muốn mua nhiều viên bì nhất có thể.

**Yêu cầu:** Hãy xác định số viên bì nhiều nhất Sen có thể mua được.

**Dữ liệu vào:** COLBALL.INP

Dòng thứ nhất: ghi hai số nguyên  $n, m$  ( $1 \leq n \leq 2 \times 10^9$ ;  $1 \leq m \leq 20$ )

Tiếp theo là  $m$  dòng, dòng thứ  $i$  ghi hai số nguyên  $a_i, b_i$  ( $a_i \leq 10^9$ ;  $1 \leq b_i \leq 10$ )

**Kết quả:** Ghi ra tệp COLBALL.OUT chỉ có một dòng ghi số nguyên theo yêu cầu.

**Ví dụ:**

COLBALL.INP	COLBALL.OUT
-------------	-------------

7 3 5 10 2 5 3 6	62
---------------------------	----

### CHUYÊN ĐỀ 3. CÁC PHƯƠNG PHÁP THIẾT KẾ CÀI ĐẶT THUẬT TOÁN

#### 1. DUYỆT QUAY LUI

Trong một số bài toán việc tìm nghiệm có thể qui về việc tìm vector hữu hạn  $(x_1, x_2, \dots, x_n)$ . Trong đó  $n$  có thể xác định trước hoặc không xác định trước. Vector cần tìm phải thỏa mãn điều kiện nào đó tùy thuộc vào từng bài toán cụ thể. Trong đó các thành phần  $x_i$  được chọn ra từ tập  $A_i$ .

##### Phương pháp:

- Lần lượt liệt kê tất cả các khả năng của nghiệm (Gọi là phần thử - duyệt).
- Với mỗi vector nghiệm ta **kiểm tra** xem nghiệm đó có thỏa mãn điều kiện của đầu bài hay không,.... (Phân sai).

#### 2. DUYỆT NHÁNH CẬN:

- + Sử dụng phương pháp quay lui nhưng tại mỗi bước đưa thêm thao tác đánh giá giá trị phương án hiện có.
- + Nếu đó là phương án tối ưu hoặc có hy vọng trở thành phương án tối ưu (tức là tốt hơn phương án hiện có) thì cập nhật lại phương án tối ưu hoặc đi tiếp theo hướng đó.
- + Trong trường hợp ngược lại thì bỏ qua hướng đang xét.

**3-Phương pháp tham lam:** Sử dụng phương pháp quay lui nhưng được tiến hành duyệt theo một tiêu chí định trước.

Chú ý: Phương pháp tham lam có thể cho ra nghiệm chưa tối ưu, nhưng cài đặt dễ và độ phức tạp thấp.

#### Ví dụ 1: Bài toán Tính số gà và số chó

Vừa gà, vừa chó.  
Bó lại cho tròn  
36 con, 100 chân

Cài bằng quay lui	Cài bằng nhánh cận
<pre>#include &lt;bits/stdc++.h&gt; using namespace std; int x, y; main(){     for(int x=1; x&lt;=36; x++)         for(int y=1; y&lt;=36; y++)             if(2*x+4*y==100 &amp; x+y==36)                 cout&lt;&lt;x&lt;&lt;" "&lt;&lt;y; }</pre>	<pre>#include &lt;bits/stdc++.h&gt; using namespace std; int x, y; main(){     for(int x=1; x&lt;=36; x++)         if(2*x+4*(36-x)==100)             cout&lt;&lt;x&lt;&lt;" "&lt;&lt;36-x; }</pre>

#### Ví dụ 2: Bài toán Máy rút tiền ATM:

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

Máy ATM hiện có  $n$  ( $n < 200$ ) tờ tiền có mệnh giá  $t_1, t_2, \dots, t_n$ . Hãy tìm cách trả ít tờ nhất với số tiền đúng bằng  $S$ .

Input ATM.INP

Dòng đầu là hai số  $n$  và  $S$

Dòng thứ hai gồm  $n$  số  $t_1, t_2, \dots, t_n$ .

Output: ATM.OUT có dạng: Nếu trả tiền đúng bằng  $S$  thì đưa ra số tờ ít nhất, nếu không đưa ra -1.

**Ví dụ:**

ATM.INP	ATM.OUT
5 10 2 5 3 4 7	2
5 13 2 5 3 4 7	-1

Ý tưởng: Tiêu chí lấy tiền mệnh giá lớn trả trước (tham lam). Có hai trường hợp (vì đề chưa nói rõ): Trường hợp 1: mỗi loại có một tờ, Trường hợp 2 mỗi loại có số tờ không hạn chế.

Trường hợp 1: mỗi loại 1 tờ	Trường hợp 2: mỗi loại nhiều tờ
<pre>#include &lt;bits/stdc++.h&gt; using namespace std; const int N=201; int a[N], n,s, res; bool ss(int x, int y){     return x&gt;y; } main(){     freopen("ATM.INP","r",stdin);     freopen("ATM.OUT","w", stdout);     cin&gt;&gt;n&gt;&gt;s;     for(int i=1; i&lt;=n; i++) cin&gt;&gt;a[i];     sort(a+1, a+n+1, ss);      for(int i=1; i&lt;=n; i++)         if (a[i]&lt;=s){             res=res+1;             s=s-a[i];         }     if(s==0) cout&lt;&lt;res;     else cout&lt;&lt;-1; }</pre>	<pre>#include &lt;bits/stdc++.h&gt; using namespace std; const int N=201; int a[N], n,s, res; bool ss(int x, int y){     return x&gt;y; } main(){     freopen("ATM.INP","r",stdin);     freopen("ATM.OUT","w", stdout);     cin&gt;&gt;n&gt;&gt;s;     for(int i=1; i&lt;=n; i++) cin&gt;&gt;a[i];     sort(a+1, a+n+1, ss);      for(int i=1; i&lt;=n; i++)         if (a[i]&lt;=s){             res=res+s/a[i];             s=s%a[i];         }     if(s==0) cout&lt;&lt;res;     else cout&lt;&lt;-1; }</pre>

## BÀI TẬP 10 PHƯƠNG PHÁP THIẾT KẾ THUẬT TOÁN

### **Bài 1: NHỮNG VIÊN KẸO**

Cô HM là cô giáo trẻ chủ nhiệm lớp chuyên Tin học. Hôm nay, cô có một chuyến đi dã ngoại cùng với các học sinh của lớp mình. Cô chuẩn bị các hộp kẹo để phát cho các em học sinh. Thật không may, số viên kẹo trong mỗi hộp lại không bằng nhau. Cô lo ngại rằng các em học

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

sinh sẽ tranh nhau những hộp kẹo có nhiều viên kẹo nhất. Để tránh điều này xảy ra, cô đã quyết định mở tất cả các hộp kẹo và đếm số viên kẹo trong mỗi hộp, sau đó cô chuyển một số viên kẹo từ hộp kẹo có nhiều viên sang hộp có ít viên hơn để số kẹo trong mỗi hộp kẹo là như nhau.

**Yêu cầu:** Hãy cho biết có ít nhất bao nhiêu viên kẹo phải được chuyển từ hộp kẹo này sang hộp kẹo khác.

**Dữ liệu vào:** Cho từ tệp văn bản VIENKEO.INP có dạng:

- Dòng thứ nhất ghi số nguyên dương  $n$  là số hộp kẹo ( $1 \leq n \leq 10^5$ )
- Dòng thứ hai ghi  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  là số viên kẹo trong mỗi hộp kẹo ( $1 \leq a_i \leq 10^5, i=1..n$ ).

Các số trên cùng một dòng ghi cách nhau ít nhất một khoảng trắng.

**Kết quả:** Ghi vào tệp văn bản VIENKEO.OUT một số nguyên duy nhất là số ít nhất các viên kẹo được chuyển từ hộp kẹo này sang hộp kẹo khác. Nếu không thể chuyển được tất cả các hộp kẹo có số viên kẹo bằng nhau thì ghi -1.

**Ví dụ:**

VIENKEO.INP	VIENKEO.OUT
5 1 1 6 1 1	4

VIENKEO.INP	VIENKEO.OUT
2 2 3 6	-1

## Bài 2: ĐĨA NHẠC

Mùa Xuân, một trong những đề tài từ trước đến nay có rất nhiều người ca ngợi và ca ngợi khá thành công trong lĩnh vực Âm nhạc. Để chuẩn bị vui xuân đón Tết Bính Thân 2016. Bé Sen đến cửa hàng ghi một đĩa CD nhạc xuân, cửa hàng ghi nhạc xuân cho Bé Sen có  $n$  bài hát với dung lượng không ít hơn 650 MB nhưng không quá 100 GB. Mỗi bài hát có dung lượng là một số nguyên dương  $a_i$  MB. Bé Sen yêu cầu chủ cửa hàng ghi vào đĩa CD càng nhiều bài hát càng tốt. Biết rằng đĩa CD trắng của Bé Sen có dung lượng ghi tối đa 650 MB. Em hãy viết chương trình giúp chủ cửa hàng thực hiện ghi đĩa nhạc xuân theo yêu cầu của Bé Sen.

**Dữ liệu vào:** Cho từ tệp văn bản CD.INP gồm hai dòng:

- Dòng thứ nhất ghi số nguyên dương  $n$  ( $1 \leq n \leq 10^5$ );
- Dòng thứ hai ghi  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^3; i=1..n$ )

Các số trên cùng một dòng cách nhau một ký tự trắng.

**Kết quả:** Ghi vào tệp văn bản CD.OUT gồm hai dòng:

- Dòng thứ nhất ghi số lượng bài hát có thể ghi vào đĩa CD.
- Dòng thứ hai ghi tổng dung lượng các bài hát có thể ghi vào đĩa CD.

**Ví dụ:**

CD.INP	CD.OUT
5 100 400 50 200 200	4 550

**Ràng buộc:**

- Có 70% số test tương ứng 70% số điểm của bài có  $1 \leq n \leq 10^4$
- Có 30% số test tương ứng 30% số điểm của bài có  $10^4 < n \leq 10^5$

## BÀI 3. VẮT SỮA BÒ:

Vào mỗi buổi sáng, anh Sen sắp đàn bò gồm  $n$  con để vắt sữa, anh dự kiến buổi sáng hôm đó, con bò thứ  $i$  có khả năng vắt được  $a_i$  lít sữa. Tuy nhiên đàn bò của anh có đặc tính là cứ mỗi lần vắt sữa một con, những con còn lại trông thấy sợ quá nên sẽ bị giảm sản lượng sữa mỗi con

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

một lít. Chẳng hạn, nếu vắt sữa con bò thứ nhất,  $n-1$  con còn lại sẽ bị giảm sản lượng sữa, sau đó Sen vắt con bò thứ hai thì  $n-2$  con bò còn lại bị giảm sản lượng sữa,....

**Yêu cầu:** Tính số lượng sữa mà Sen có thể vắt được nhiều nhất.

Input: VATSUA.INP

- Dòng thứ nhất ghi số nguyên  $n$  ( $1 < n \leq 105$ ) – là số lượng con bò  
- Dòng thứ hai ghi  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $a_i \leq 1000; i=1..n$ ) – là sản lượng sữa của các con bò.

**Ví dụ:**

VATSUA.INP	VATSUA.OUT
4 4 4 4 4	10
4 2 1 4 3	6

#### **BÀI 4:** (7,0 điểm) **MUA ĐỒ ĂN**

Cô HM tổ chức chuyến đi dã ngoại trong  $n$  ngày và việc ăn uống của lớp cũng khá tốn kém khiến cô phải tính toán cẩn thận. Ngày thứ  $i$  cần có đúng  $a_i$  kg gạo, giá bán trong ngày thứ  $i$  là  $p_i$  đồng cho mỗi kg. Mỗi ngày cô HM có thể mua số lượng gạo không hạn chế, nếu thừa có thể để dành sang những ngày hôm sau.

**Yêu cầu:** Giúp cô HM quyết định lượng mua gạo mỗi ngày để đáp ứng yêu cầu của lớp với tổng số tiền phải chi là ít nhất có thể.

**Dữ liệu vào:** Cho từ tệp văn bản FEEDING.INP

- Dòng 1: Chứa số nguyên dương  $n$  ( $n \leq 10^5$ )
- Dòng 2: Chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $\forall i: a_i \leq 10^5, i=1..n$ )
- Dòng 3: Chứa  $n$  số nguyên dương  $p_1, p_2, \dots, p_n$  ( $\forall i: p_i \leq 10^5, i=1..n$ )

Các số trên một dòng được ghi cách nhau bởi khoảng trắng.

**Kết quả:** Ghi ra tệp văn bản FEEDING.OUT một số nguyên duy nhất là tổng số tiền phải chi (tính bằng đồng) để mua gạo theo phương án tìm được.

**Ví dụ:**

FEEDING.INP	FEEDING.OUT
3 1 2 3 3 2 1	10
3 1 2 3 1 2 3	6

#### **Giải thích**

Ngày 1 mua 1 kg (3đ), ngày 2 mua 2 kg (4đ), ngày 3 mua 3 kg (3đ)

Ngày 1 mua luôn 6 kg (6đ) và dùng số đồ ăn đó cho cả hai ngày tiếp theo

**Ràng buộc kĩ thuật:** 50% số điểm ứng với các test có  $n \leq 2000$

#### **Bài 5. ĐUA NGỰA**

Thời Xuân Thu, một lần Tôn Tần tổ chức đua ngựa với vua Tề. Tôn Tần và vua Tề mỗi người có đúng  $N$  con ngựa, ngựa thứ  $i$  của Tôn Tần có tốc độ chạy là  $a_i$ , ngựa thứ  $j$  của vua Tề có tốc độ chạy là  $b_j$ . Luật đua như sau:

+ Có tất cả  $N$  cặp đua, mỗi cặp đua có một ngựa của Tôn Tần và một ngựa của vua Tề.

+ Con ngựa nào cũng phải tham gia đúng một cặp đua

+ Trong một cặp đua, con ngựa nào tốc độ cao hơn sẽ thắng, nếu hai con ngựa có cùng tốc độ thì kết quả của cặp đua đó sẽ hoà.

- **Yêu cầu:** Hãy giúp Tôn Tẫn chọn ngựa ra đấu cặp đua với ngựa vua Tề sao cho có được nhiều trận thắng nhất có thể.

- **Dữ liệu vào:** Cho từ tệp văn bản DUANGUA.INP có cấu trúc:

- Dòng thứ nhất ghi số nguyên dương  $N$  ( $1 \leq N \leq 10^4$ ).
- Dòng thứ hai ghi  $N$  số nguyên dương, số thứ  $i$  là tốc độ chạy của con ngựa thứ  $i$  của Tôn Tẫn ( $1 \leq a_i \leq 10^4, i=1..N$ ).
- Dòng thứ ba ghi  $N$  số nguyên dương, số thứ  $j$  là tốc độ chạy của con ngựa thứ  $j$  của vua Tề ( $1 \leq b_j \leq 10^4, j=1..N$ ).

Các số trên một dòng được ghi cách nhau bởi kí tự trắng.

- **Kết quả:** Ghi vào tệp văn bản DUANGUA.OUT chỉ có một dòng chứa một số nguyên duy nhất theo yêu cầu bài toán.

- **Ví dụ:**

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

**Ràng buộc dữ liệu:**

- 60% số test tương ứng với  $1 < n \leq 10^3$
- 40% số test tương ứng với  $10^3 < n \leq 10^4$



## BÀI TẬP 11 DUYỆT TOÀN BỘ && KỸ THUẬT LẬP TRÌNH

### BÀI 1: CẮT DÂY RUY BĂNG

Để chuẩn bị cho cắm trại xuân sắp đến, các bạn học sinh lớp TH dự định kết nơ bằng các dây ruy băng để trang trí trại. Các nơ trang trí có 3 loại: Loại thứ nhất cần đoạn dây ruy băng có chiều dài **a**, loại thứ hai cần đoạn dây ruy băng có chiều dài **b** và loại thứ ba cần đoạn dây ruy băng có độ dài **c**. Các bạn đã mua một cuộn dây ruy băng có chiều dài **n** và dự định dùng hết cuộn dây này để kết nơ.

**Yêu cầu:** Hãy giúp các bạn lớp 9A cắt cuộn dây ruy băng thành nhiều đoạn nhất để kết thành ba loại nơ (số lượng mỗi loại không nhất thiết phải bằng nhau) sao cho không có phần thừa nào sau khi cắt. Dữ liệu luôn luôn đảm bảo sau khi cắt hợp lý không có phần ru băng thừa.

**Dữ liệu:** RUYBANG.INP chỉ có một dòng chứa bốn số nguyên **n, a, b, c**

(  $1 < n, a, b, c \leq 10000; a \leq b \leq c$  )

**Kết quả:** Ghi ra tệp văn bản RUYBANG.OUT chứa số nguyên là số lượng đoạn ruy băng cắt được theo yêu cầu.

**Ví dụ:**



Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

RUYBANG.INP	RUYBANG.OUT	Giải thích
5 2 3 5	2	Cắt 2 đoạn có chiều dài 2 và 3
10 3 5 7	2	Cắt 2 đoạn có chiều dài 3 và 7 hoặc cắt thành 2 đoạn mỗi đoạn có chiều dài là 5

**Ràng buộc:**

Sub1: 60% số test tương ứng 60% số điểm của bài có  $n \leq 1000$ ;

Sub2: 40% số test tương ứng 60% số điểm của bài có  $100 < n \leq 10000$

**Bài 2. (6,0 điểm) XÓA DÃY (Chọn đội quốc gia 2019)**

Cho dãy số nguyên  $A = (a_0, a_1, \dots, a_{n-1})$ . Bạn được phép xóa đi một phần tử có giá trị nhỏ nhất hoặc xóa đi một phần tử có giá trị lớn nhất trong dãy.

**Yêu cầu:** Tìm cách dùng ít nhất các phép xóa theo luật trên để thu được một dãy mới có tổng các phần tử trong dãy bằng 0 (dãy rỗng cũng được coi là dãy có tổng các phần tử bằng 0).

**Dữ liệu:** Vào từ file văn bản DELARR.INP

- Dòng 1 chứa số nguyên dương  $n \leq 10^5$ .
- Dòng 2 chứa  $n$  số nguyên  $a_0, a_1, \dots, a_{n-1}$  cách nhau bởi dấu cách ( $\forall i: |a_i| \leq 10^9$ ).

**Kết quả:** Ghi ra file văn bản DELARR.OUT một số nguyên duy nhất là số phép xóa cần thực hiện.

**Ví dụ:**

DELARR.INP	DELARR.OUT	Giải thích
6 -4 -5 1 2 -3 10	3	Xóa số -5, 10 và -4 theo thứ tự đó
8 -3 -3 -3 0 0 0 4 4	5	
9 1 2 3 4 5 6 7 8 9	9	

Bộ test chia làm 2 subtasks:

- Subtask 1: (50% số điểm) gồm các test có  $n \leq 2000$ .
- Subtask 2: (50% số điểm) không có ràng buộc bổ sung.

**Bài 3: (7,0 điểm) ĐỔI CHỖ (HS Giải thành phố SaDec 2018)**

Theo tâm lí học thì những học sinh thích học Sử thường không thích học Tin, những học sinh thích học Tin thì chắc chắn không thích học Sử. Một lần, nhà trường muốn kiểm tra năng lực của học sinh. Ban giám hiệu quyết định chọn ra  $n$  em thích học Sử và  $n$  em thích học Tin tham gia xếp thành một hàng. Vì tính nhanh nhẹn nên những em thích học Tin đã nhanh chân đứng xếp hàng trước cả  $n$  em thích học Sử. Thấy vậy, thầy giáo phụ trách ra lệnh thực hiện đổi chỗ  $k$  lần. Lần thứ  $i$  thầy cho đổi chỗ 2 em ở vị trí  $a_i$  và  $b_i$ . Sau mỗi lần đổi chỗ, thầy muốn biết còn bao nhiêu em thích học Tin vẫn đứng ở nửa đầu của hàng.

**Yêu cầu:** Viết chương trình giúp thầy giáo trả lời được ngay câu hỏi trên.

**Dữ liệu vào:** Cho từ tệp văn bản SWAP.INP gồm:

- Dòng thứ nhất chứa số nguyên dương  $n$  ( $1 \leq n \leq 10^5$ )
- Dòng thứ hai chứa số  $k$  - số lần đổi chỗ ( $1 \leq k \leq n$ )
- Dòng thứ  $i$  trong  $k$  dòng tiếp theo, mỗi dòng ghi hai số  $a_i$  và  $b_i$  ( $1 \leq a_i, b_i \leq 2n; i=1..n$ )

**Kết quả:** Ghi ra tệp văn bản SWAP.OUT gồm  $k$  dòng, mỗi dòng một số nguyên là số lượng học sinh thích học Tin đứng ở nửa đầu của hàng.



Ví dụ:

SWAP.INP	SWAP.OUT
2	1
4	1
1 3	2
3 4	1
4 1	
2 3	

Ràng buộc:

- Có 60% số test ứng với 60% số điểm có  $1 \leq n \leq 10^3$
- Có 40% số test ứng với 40% số điểm có  $10^3 < n \leq 10^5$

#### Bài 4: Sắp xếp sách

Steve có một thư viện nhỏ gồm  $n$  cuốn sách. Do phòng hẹp, sách phải một chồng cao. Sách được đánh số từ 1 đến  $n$  theo thứ tự từ điển của tên sách. Ban đầu, các cuốn sách nằm đúng vị trí của mình từ trên xuống dưới theo đúng trình tự từ 1 đến  $n$ .

Khi cần một cuốn sách nào đó, Steve dễ dàng rút ra, nhưng sau đó không thể nhét lại vào vị trí cũ mà phải đặt nó lên trên cùng. Thời gian trôi qua và các sách không còn nằm đúng vị trí ban đầu làm cho việc tìm cuốn sách cần thiết trở nên khó khăn. Steve muốn khôi phục lại vị trí ban đầu của các cuốn sách. Để làm việc đó Steve rút ra một cuốn sách, đặt nó lên trên cùng, rồi lại rút ra cuốn khác, đặt lên trên . . . cứ như thế cho đến khi sách được sắp xếp theo thứ tự từ điển từ trên xuống dưới. Ví dụ, với  $n = 4$  và trật tự các cuốn sách từ trên xuống dưới là (1, 3, 4, 2), Steve cần rút cuốn 2 đặt lên trên, sau đó rút tiếp cuốn 1 đặt lên trên, tổng cộng là 2 lần rút sách ra đặt lại.



**Yêu cầu:** Cho  $n$  và trình tự hiện tại từ trên xuống dưới của các cuốn sách. Hãy xác định số lần tối thiểu cần rút sách ra để sắp xếp lại sách theo đúng trình tự.

**Dữ liệu:** Vào từ file văn bản BOOKSORT.INP:

- Dòng đầu tiên chứa số nguyên  $n$  ( $n \leq 300\,000$ ),
- Dòng thứ  $i$  trong  $n$  dòng sau chứa một số nguyên xác định số thứ tự của cuốn sách ở vị trí  $i$  tính từ trên xuống.

**Kết quả:** Đưa ra file văn bản BOOKSORT.OUT một số nguyên – số lần tối thiểu rút sách cần thực hiện.

Ví dụ:

BOOKSORT.INP	BOOKSORT.OUT
4	2
1	
3	
4	
2	

### Bài 5. QUẢNG ĐƯỜNG ĐI BỘ (Chọn đội quốc gia 2018)

Lớp chuyên tin có  $n$  bạn ở trong ký túc xá, những bạn này được đánh số từ 1 tới  $n$ . Hàng ngày các bạn đi bộ từ ký túc xá đến khu giảng đường trên quảng đường độ dài  $L$ . Bạn thứ  $i$  xuất phát tại thời điểm  $a_i$  và đi với vận tốc không đổi: mất  $t_i$  đơn vị thời gian để di chuyển qua 1 đơn vị độ dài.

Tuy nhiên, hôm nay là ngày thi học kỳ và các bạn muốn tranh thủ thời gian trao đổi với nhau trước giờ thi. Vì vậy, cứ mỗi khi một nhóm người (từ hai người trở lên) gặp nhau trên đường, họ sẽ tiếp tục đi cùng nhau đến trường theo tốc độ của người đi chậm nhất trong nhóm. Tất nhiên nếu có nhiều bạn xuất phát vào cùng thời điểm thì họ sẽ đi cùng nhau trong suốt quảng đường theo tốc độ của người chậm nhất trong số đó.

**Yêu cầu:** Với mỗi bạn, hãy cho biết thời điểm bạn đó đến trường trong ngày thi học kỳ.

**Dữ liệu:** Vào từ file văn bản WALKING.INP

- Dòng thứ nhất chứa hai số nguyên dương  $n$  và  $L$  ( $n \leq 10^5$ ;  $L \leq 10^9$ )
- $n$  dòng tiếp theo, dòng thứ  $i$  chứa hai số nguyên dương  $a_i$  và  $t_i$  ( $a_i, t_i \leq 10^9$ ,  $i=1..n$ )

*Các số trên cùng một dòng được ghi cách nhau bởi dấu cách.*

**Kết quả:** Ghi ra file văn bản WALKING.OUT gồm  $n$  dòng, dòng thứ  $i$  ghi thời điểm bạn thứ  $i$  đến trường.

**Ví dụ:**

WALKING.INP	WALKING.OUT
6 10	102
5 8	102
2 10	102
9 1	31
1 3	31
1 2	31
1 1	

**Chú ý:** 50% số điểm ứng với các test có  $n \leq 1000$

-----



#### CHUYÊN ĐỀ 4. SẮP XẾP & TÌM KIẾM

**Bài toán:** Cho dãy A là dãy tăng gồm N số nguyên khác nhau  $a_1, a_2 \dots, a_n$  và số nguyên K. Hỏi trong dãy A có phần tử nào có giá trị bằng K không? Nếu tìm thấy đưa ra vị trí số k trong dãy A, nếu không tìm thấy đưa ra -1.

**Ví dụ:**

TKNP.INP	TKNP.OUT
5 7 2 3 5 7 11	4
5 4 2 3 5 7 11	-1

Để giải quyết bài toán trên ngoài thuật toán tìm kiếm tuần tự có độ phức tạp  $O(N)$ , còn có thuật toán tìm kiếm nhị phân có độ phức tạp  $O(\log N)$  với ý tưởng như sau:

Vì dãy A là dãy tăng nên ta tìm cách thu hẹp phạm vi tìm kiếm sau mỗi lần so sánh khóa với số hạng được chọn. Để làm được điều đó, ta chọn số hạng  $A_{giua}$  ở “giữa dãy” để so sánh với k, trong đó  $Giua = [(N+1)/2]$ . Khi đó xảy ra một trong ba trường hợp:

- Nếu  $A_{giua} = k$  thì **giua** là là phần tử cần tìm, thông báo có phần tử bằng K rồi kết thúc thuật toán.
- Nếu  $A_{giua} > k$  thì việc tìm kiếm tiếp theo chỉ xét trên dãy  $a_1, a_2 \dots, a_{giua-1}$
- Nếu  $A_{giua} < k$  thì việc tìm kiếm tiếp theo chỉ xét trên dãy  $a_{giua+1}, a_{giua+2} \dots, a_N$

Quá trình trên sẽ được lặp đi lặp lại một số lần cho tới khi hoặc đã tìm thấy khóa K trong dãy A hoặc phạm vi tìm kiếm bằng rỗng.

**Code:**

```
#include <bits/stdc++.h>
using namespace std;
const int N=1e6+6;
int a[N], k, n;
int tknp(int k){
    int dau=1, cuoi=n, giua;
    while(dau<=cuoi){
        giua=(dau+cuoi)/2;
        if (a[giua]==k) return giua;
        if (a[giua]>k) cuoi=giua-1;
        else dau=giua+1;
    }
    return -1;
}
main(){
    freopen("TKNP.INP","r", stdin);
    freopen("TKNP.OUT","w", stdout);
    cin>>n>>k;
    for(int i=1; i<=n; i++) cin>>a[i];
    cout<<tknp(k);
}
```



**Ghi chú:**

- Tư tưởng chia bài toán ban đầu thành các bài toán nhỏ hơn gọi là tư tưởng chia để trị.
- Trong trường hợp tìm số  $a_i \leq k$  mà lớn nhất. Nếu  $a[\text{giua}] \leq k$  đó là kết quả tạm  $\text{res} = a[\text{giua}]$  ta tiếp tục chặt lên cho đến khi  $a[\text{giua}] > k$

**Ví dụ:**

TKNP.INP	TKNP.OUT
5 4	3
2 3 7 11 13	

- Trong trường hợp tìm số  $a_i \geq k$  mà nhỏ nhất. Nếu  $a[\text{giua}] \geq k$  đó là kết quả tạm  $\text{res} = a[\text{giua}]$  ta tiếp tục chặt xuống cho đến khi  $a[\text{giua}] < k$

**Ví dụ:**

TKNP.INP	TKNP.OUT
5 4	7
2 3 7 11 13	

Tìm $a_i \leq k$ mà lớn nhất	Tìm $a_i \geq k$ mà nhỏ nhất
<pre>#include &lt;bits/stdc++.h&gt; using namespace std; const int N=1e6+6; int a[N], k, n;  int tknp(int k){     int dau=1, cuoi=n, giua, res;     while(dau&lt;=cuoi){         giua=(dau+cuoi)/2;         if (a[giua]&lt;=k){             res= a[giua];             dau=giua+1;         }         else cuoi=giua-1;     }     return res; }  main(){     freopen("TKNP.INP","r", stdin);     //freopen("TKNP.OUT","w", stdout);     cin&gt;&gt;n&gt;&gt;k;     for(int i=1; i&lt;=n; i++) cin&gt;&gt;a[i];     cout&lt;&lt;tknp(k); }</pre>	<pre>#include &lt;bits/stdc++.h&gt; using namespace std; const int N=1e6+6; int a[N], k, n;  int tknp(int k){     int dau=1, cuoi=n, giua, res;     while(dau&lt;=cuoi){         giua=(dau+cuoi)/2;         if (a[giua]&gt;=k){             res= a[giua];             cuoi=giua-1;         }         else dau=giua+1;     }     return res; }  main(){     freopen("TKNP.INP","r", stdin);     //freopen("TKNP.OUT","w", stdout);     cin&gt;&gt;n&gt;&gt;k;     for(int i=1; i&lt;=n; i++) cin&gt;&gt;a[i];     cout&lt;&lt;tknp(k); }</pre>

Chú ý các hàm sau đây chỉ hoạt động khi dãy số được sắp xếp tăng dần:

+ Hàm `lower_bound(first, last, value)` : trả về vị trí đầu tiên lớn hơn hoặc bằng value trong dãy.

+ Hàm `upper_bound(first, last, value)` : trả về vị trí đầu tiên lớn hơn `value` trong dãy. Ví dụ: `q = lower_bound(a+1, a+n+1, p) - a; // a[1..n], a[q]>=p`    `q = upper_bound(a+1, a+n+1, p) - a; // a[1..n], a[q]>p`

+ Hàm `binary_search(a, a+n, k);` // trả về giá trị 0 nếu không tìm thấy `k` và ngược lại trả về 1

+ Hàm `max_element(a, a+n);` // trả về địa chỉ đặt giá trị lớn nhất trong mảng `a`.

Nếu muốn lấy giá trị lớn nhất thì ta có thể viết `*max_element(a, a+n)`; Ngoài những hàm trên trong C++ còn có rất nhiều các hàm khác.

Ta còn có thuật toán tìm kiếm tam phân với ý tưởng tương tự thuật toán tìm kiếm nhị phân và thời gian chạy chương trình nhanh hơn thuật toán tìm kiếm nhị phân, nhưng không đáng kể.

## BÀI TẬP 12 SẮP XẾP – TÌM KIẾM NHỊ PHÂN

### Bài 1: ĐOÁN SỐ

Hai người chơi như sau: Người thứ nhất sẽ nghĩ ra một số nguyên dương  $X$  trong khoảng từ 1 đến  $N$  ( $N=10^6$ ). Người thứ hai sẽ lần lượt đưa ra các số dự đoán. Với mỗi số dự đoán này, người thứ hai sẽ nhận được câu trả lời cho biết số mình vừa nêu ra lớn hơn, nhỏ hơn, hay bằng với số mà người thứ nhất đã nghĩ. Em hãy giúp người thứ hai chọn đúng số cần tìm với số lần đoán càng ít nhất.

**Input:** DOANSO.INP chỉ có 1 dòng ghi số nguyên  $x$  ( $1 \leq x \leq 10^6$ )

**Output:** DOANSO.INP Số lần đoán số ít nhất tính được.

**Ví dụ:**

DOANSO.INP	DOANSO.OUT
2017	18

### Bài 2: XE ĐẠP ĐÔI

Đến với Làng hoa Sa Đéc du khách có thể thuê xe đạp đôi để đi tham quan những loài hoa tuyệt đẹp ở nơi đây. Biết rằng, đoàn tham quan có  $n$  học sinh được đánh số thứ tự từ 1 đến  $n$ , học sinh thứ  $i$  có trọng lượng là  $a_i$ . Đoàn tham quan chỉ thuê được duy nhất một xe đạp đôi với tải trọng là  $m$ , hai học sinh chỉ có thể cùng lên xe nếu tổng trọng lượng của hai học sinh đó không vượt quá  $m$ .

**Yêu cầu:** Hãy đếm số cặp hai học sinh có thể sử dụng được xe đạp đôi (đếm số lượng  $a_i + a_j \leq m$ ,  $i \neq j$ ). Chẳng hạn: có ba học sinh  $a, b, c$  mà trọng lượng:  $a+b \leq m$  và  $a+c \leq m$  mà  $b+c > m$  được tính là 2 cặp.

**Dữ liệu vào:** Cho từ tệp văn bản XEDAP.INP gồm:

- Dòng thứ nhất ghi hai số nguyên dương  $n, m$  ( $1 \leq n, m \leq 10^6$ )
  - Dòng hai ghi  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^6; i=1..n$ ).
- Các số ghi cùng dòng cách nhau bởi dấu cách

**Kết quả:** Ghi ra tệp văn bản XEDAP.OUT chỉ có một số nguyên duy nhất là đáp số theo yêu cầu.

**Ví dụ:**



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

**Ràng buộc:** Có 60% số test tương ứng 60% số điểm có  $n \leq 10^3$

### **Bài 3: HÀNG CÂY PHI LAO**

Cây phi lao là loại cây được trồng nhiều ở ven biển nước ta. Ngoài công dụng chắn gió, giữ đất, cây phi lao còn là nguyên liệu để làm giấy. Ở ven biển, ông Năm trồng cây phi lao thành một hàng dài gồm có  $n$  cây, các cây có độ cao lần lượt là  $a_1, a_2, \dots, a_n$  mét. Để đảm bảo chắn gió và giữ đất, ông Năm thu hoạch phi lao từ trên ngọn xuống để giữ lại phần gốc. Chẳng hạn để thu hoạch  $m$  mét gỗ phi lao, ông Năm cưa hàng cây ở độ cao  $h$  mét nào đó (dĩ nhiên những cây có độ cao không lớn hơn  $h$  thì không bị cưa) sao cho thu hoạch phần ngọn đủ  $m$  mét gỗ và số gỗ phi lao dư ra là ít nhất. Ví dụ, hàng cây có các cây với độ cao tương ứng là: 20; 15; 10 và 18 mét, cần lấy 7 mét gỗ, ông Năm cưa hàng cây ở độ cao  $h = 15$  là hợp lí nhất vì ông Năm thu hoạch được 8 mét phần ngọn (dư 1 mét), phần gốc còn lại lần lượt là: 15; 15; 10; 15 mét.

#### **Yêu cầu:**

- Tính tổng số lượng mét gỗ thu hoạch được nếu chọn cưa hàng cây ở độ cao bằng với cây phi lao thấp nhất.
- Hãy tìm vị trí  $h$  mét hợp lí nhất ( $h$  là số nguyên) sao cho thu hoạch được  $m$  mét gỗ và số mét gỗ dư ra là ít nhất.

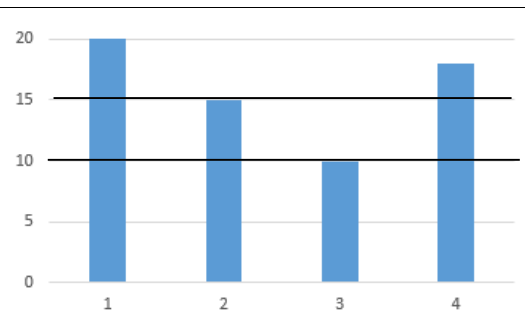
**Dữ liệu vào:** Cho từ tệp văn bản PHILAO.INP gồm hai dòng:

- Dòng thứ nhất ghi hai số nguyên dương  $n$  và  $m$  ( $1 \leq n \leq 10^6$ ;  $1 \leq m \leq 10^7$ )
  - Dòng thứ hai ghi  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  với  $a_i$  là độ cao của cây thứ  $i$  ( $1 \leq a_i \leq 10^9$ ;  $i=1 \dots n$ ).
- Các số ghi trên cùng một dòng cách nhau một kí tự trắng.

**Kết quả:** Ghi vào tệp văn bản PHILAO.OUT gồm hai dòng:

- Dòng thứ nhất ghi số nguyên  $s$  là tổng số mét gỗ thu hoạch được nếu cưa hàng cây ở độ cao bằng với cây phi lao thấp nhất.
  - Dòng thứ hai ghi số nguyên  $h$  tìm được để thu hoạch đủ  $m$  mét gỗ và số gỗ dư ra là ít nhất. Nếu không tìm được  $h$  ghi ra số -1.
- Các số ghi trên cùng một dòng cách nhau một kí tự trắng.

#### **Ví dụ:**

PHILAO.INP	PHILAO.OUT	Mô phỏng
4 7 20 15 10 18	23 15	

**Ràng buộc dữ liệu:**

- Có 70% số test tương ứng 70% số điểm của bài có  $n \leq 10^4$
- Có 30% số test tương ứng 30% số điểm của bài có  $n \leq 10^6$

#### Bài 4: TÌNH YÊU TRONG CHIẾN TRANH

Nếu như nói đến chiến tranh là nói đến sự khốc liệt, sự mất mát, những đau khổ, những hy sinh lớn lao thì khi nói đến tình yêu ta nghĩ tới sự êm dịu, ngọt ngào, hạnh phúc. Chiến tranh có thể hủy diệt mọi thứ trên đường nó đi qua, những con người có thể bị bao thương tật, biến dạng hình thể trong chiến tranh, song tình yêu như một sức mạnh không thể giập vùi, tình yêu ấy đã giúp bao anh lính bộ đội cụ Hồ đã chắc tay súng để bảo vệ tổ quốc và có niềm tin vào một ngày mai đất nước độc lập thống nhất, tình yêu bị kìm nén ấy sẽ lại bùng cháy và thăng hoa cùng dân tộc. Những con người đầy phần đông là thanh niên, khi đó tại các trường đại học khi có lệnh tổng động viên nhiều sinh viên cả nam nữ gác bút nghiên đi theo tiếng gọi của lịch sử. Trường đại học hồi đó quản lý sinh viên bằng mã sinh viên, hai sinh viên khác nhau sẽ có 2 mã khác nhau. Trong khi học tập các cặp nam nữ sinh viên đã nảy sinh những tình cảm đẹp giành cho nhau. Các cặp nam nữ sinh viên nói trên ghi lại mã sinh viên của nhau trước khi lên đường nhập ngũ để ngày giải phóng họ dễ tìm lại nhau trong trường đại học.

Năm 1975, khi chiến tranh kết thúc, đất nước hoàn toàn giải phóng, những sinh viên năm nào người đã ngã xuống trên mặt trận, một số may mắn còn lại và có điều kiện quay về trường tiếp tục học tập. Biết rằng có  $n$  sinh viên nam và  $m$  sinh viên nữ đã trở về trường.  $n$  sinh viên nam, trong hoàn cảnh chiến tranh khốc liệt, họ đã để thất lạc mã số của những người sinh viên nữ và chỉ nhớ được mã sinh viên của chính họ:  $b_1, b_2, b_3, \dots, b_n$ ;  $m$  sinh viên nữ vẫn giữ lại được mã sinh viên của mình:  $g_1, g_2, g_3, \dots, g_m$  và của các sinh viên nam mà họ đã dành tình cảm thời sinh viên trước đó:  $y_1, y_2, y_3, \dots, y_m$  ( $g_i$  giữ  $y_i$ ). Cặp sinh viên  $b_i$  và  $g_j$  sẽ tìm được nhau nếu  $y_j = b_i$ .

**Yêu cầu:** Em hãy thống kê xem có bao nhiêu cặp sinh viên nam nữ trong số trên có thể tìm được nhau và chỉ ra các cặp sinh viên đó.

**Dữ liệu vào :** LIW.INP gồm:

Dòng 1:  $n \ m$

Dòng 2: mã sinh viên nam  $b_1, b_2, b_3, \dots, b_n$

Dòng 3: mã sinh viên nữ  $g_1, g_2, g_3, \dots, g_m$

Dòng 4: mã các sinh viên nam mà sinh viên nữ đã dành tình cảm thời sinh viên trước đó:  $y_1, y_2, y_3, \dots, y_m$  ( $g_i$  giữ  $y_i$ ).

**Dữ liệu ra:** tệp LIW.OUT ghi số cặp sinh viên nam nữ tìm được nhau.

**Ví dụ:**

LIW.INP	LIW.OUT
4 5	3
8 6 2 4	
1 7 5 3 11	
4 2 10 12 6	

**Giới hạn:**  $1 \leq n, m \leq 10^4$ .  $1 \leq b_i, g_i, y_i \leq 10^{16}$ .

**Bài này có thể yêu cầu thêm:** chỉ ra các cặp sinh viên đã tìm được nhau.

#### Bài 5. XẾP TRÚNG

Cho  $n$  quả trứng được đưa vào dây chuyền theo thứ tự, quả trứng thứ  $i$  có thể tích là  $a_i$ , ở cuối dây có  $m$  thùng để chứa trứng khi đầy thì chuyển sang thùng khác, các thùng này có sức chứa như nhau.

**Yêu cầu:** Hãy tìm sức chứa **k** tối thiểu của mỗi thùng để đựng hết số quả trứng

**Dữ liệu vào:** Cho từ tệp XEPTRUNG.INP

- Dòng thứ nhất ghi hai số nguyên dương **n, m** ( $n, m \leq 10^5$ ).

- Dòng thứ hai ghi n số **a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>n</sub>** ( $a_i \leq 10^5$ ;  $i=1..n$ );

**Kết quả:** Xuất ra tệp XEPTRUNG.OUT chỉ có một dòng ghi số nguyên k tính được.

**Ví dụ:**

XEPTRUNG.INP	XEPTRUNG.OUT
5 3	12
6 5 4 8 9	

**Ràng buộc:** có 50% số test tương ứng 50% số điểm có  $n, m < 100$



## CHUYÊN ĐỀ 5. QUY HOẠCH ĐỘNG

### 1. Một số khái niệm:

Phương pháp quy hoạch động dùng để giải bài toán tối ưu (max/min) có bản chất đệ quy, với tư tưởng chia để trị kết hợp lưu nghiệm các bài toán con.

Tư tưởng chia để trị (divide and conquer): Để giải quyết một bài toán lớn, ta chia nó thành nhiều bài toán con cùng dạng với nó để có thể giải quyết độc lập.

Bài toán giải theo phương pháp quy hoạch động gọi là bài toán quy hoạch động.

Công thức phối hợp nghiệm của các bài toán con để có nghiệm của bài toán lớn gọi là công thức truy hồi.

Tập các bài toán con có ngay lời giải để từ đó giải quyết các bài toán lớn hơn gọi là cơ sở quy hoạch động.

Không gian lưu trữ lời giải các bài toán con để tìm cách phối hợp chúng gọi là bảng phương án.

- Trước khi áp dụng phương pháp quy hoạch động ta phải xét xem bài toán đó có thể chia được thành các bài toán con hay không? Bài toán con nhỏ nhất là gì?

- Hạn chế của phương pháp quy hoạch động: Tốn nhiều bộ nhớ, khó tìm công thức truy hồi, truy vết tìm nghiệm bài toán khó.

### 2. Các bước cơ bản để giải một bài toán quy hoạch động:

B1. Xác định cơ sở quy hoạch động (xác định bài toán con nhỏ nhất)

B2. Lập bảng phương án lưu nghiệm các bài toán con: Dựa vào nghiệm bài toán con nhỏ nhất tính nghiệm bài toán con tiếp theo.

B3. Xây dựng công thức truy hồi: Dựa vào bảng phương án để lập công thức truy hồi. Công thức truy hồi là công thức để tính nghiệm bài toán con dạng tổng quát.

B4. Truy vết: Dựa vào nghiệm các bài toán con để truy vết trả lời nghiệm bài toán ban đầu.

### 3. Một số bài toán quy hoạch động điển hình:

**Bài toán 1:** Tính số Fibonacci thứ  $n$

**Input:** Fibo.inp chỉ có 1 dòng ghi số  $n$  ( $n < 50$ )

**Output:** Fibo.inp chỉ có 1 dòng ghi số nguyên theo yêu cầu.

**Ví dụ:**

FIBO.INP	FIBO.OUT
6	8

Phân tích bài toán							Cài đặt
Gọi f[i] là số Fibo của bài toán con thứ i. (i=1..n)							#include <bits/stdc++.h> using namespace std; const int N=1e6+6; long long f[N]; int n; main(){
B1. Cơ sở qhd:							
- Ta có bài toán con nhỏ nhất là n=1 -> F[1]=1							
- Ta có bài toán con tiếp theo n=2 -> F[2]=1							
B2. Lập bảng phương án:							
n	1	2	3	4	5	6	}

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

F[n]	1	1	2	3	5	8
<p>B3: Công thức truy hồi:  <math>F[i]=F[i-1]+F[i-2]</math> (<math>i=3..n</math>)</p> <p>B4: Truy vết: F[n]</p> <p>* Bài toán này nếu cài bằng đệ quy hay kỹ thuật 3 biến cũng có <math>O(n)</math>, nhưng ít tốn bộ nhớ hơn.</p>						
<pre>freopen("Fibo.inp","r",stdin); freopen("Fibo.out","w",stdout); cin&gt;&gt;n; f[1]=1; f[2]=1; for(int i=3; i&lt;=n; i++)     f[i]=f[i-1]+f[i-2]; cout&lt;&lt;f[n]; }</pre>						

## Bài toán 2. TỔNG LỚN NHẤT CỦA ĐOẠN CON

Cho một dãy gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n$ . Hãy tìm một đoạn con (dãy gồm các phần tử liên tiếp nhau) có tổng lớn nhất.

**Dữ liệu vào:** từ tệp MAXSUB.INP

- Dòng đầu ghi số  $n$  ( $1 \leq n \leq 10^6$ )

- Dòng hai ghi  $n$  số nguyên  $a_1, a_2, \dots, a_n$ , mỗi số có giá trị tuyệt đối không vượt quá  $10^9$ .

**Kết quả:** ghi vào tệp MAXSUB.OUT ghi ra 1 số là tổng lớn nhất tìm được.

**Ví dụ:**

MAXSUB.INP	MAXSUB.OUT
10 1 3 -5 2 7 6 -2 4 -3 1	17

Phân tích bài toán	Cài đặt																																	
<p>Gọi <math>f[i-1]</math> là tổng dãy con có <math>i-1</math> phần tử bắt đầu từ 1 kết thúc tại <math>i-1</math>.</p> <p>Ta có: bài toán con nhỏ nhất: <math>n=1</math> dãy có 1 phần tử: <math>f[1]=a[1] \rightarrow</math> CSQHĐ.</p> <table><tr><td>i</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr><tr><td>a[i]</td><td>1</td><td>3</td><td>-5</td><td>2</td><td>7</td><td>6</td><td>-2</td><td>4</td><td>-3</td><td>1</td></tr><tr><td>f[i]</td><td>1</td><td>4</td><td>-1</td><td>2</td><td>9</td><td>15</td><td>13</td><td>17</td><td>14</td><td>15</td></tr></table> <p>- Xét phần tử <math>i</math>:</p> <p>+ Nếu <math>f[i-1]+a[i]&gt;f[i-1]</math> thì <math>f[i]=f[i-1]+a[i]</math>.</p> <p>+ Nếu <math>f[i-1]+a[i]&lt;f[i-1]</math> thì <math>f[i]=a[i]</math> (bắt đầu tổng mới).</p> <p>-&gt; Công thức truy hồi:</p> <p><math>f[i]=\max(a[i], f[i-1]+a[i]) \ (i=2..n)</math></p> <p>Số lớn nhất trong mảng <math>f</math> là kết quả cần tính.</p> <p>--&gt; Truy vết.</p> <p>Một cách tiếp cận khác là dùng thuật toán Kadane cũng độ phức tạp <math>O(n)</math>.</p>	i	1	2	3	4	5	6	7	8	9	10	a[i]	1	3	-5	2	7	6	-2	4	-3	1	f[i]	1	4	-1	2	9	15	13	17	14	15	<pre>#include &lt;bits/stdc++.h&gt; using namespace std; const int N=1e6+1; const int oo=INT_MAX; int n; long long f[N], a[N]; long long res=-oo; main(){     freopen("MAXSUB.INP","r", stdin);     freopen("MAXSUB.OUT","w", stdout);     cin&gt;&gt;n;     for(int i=1; i&lt;=n; i++) cin&gt;&gt;a[i];      f[1]=a[1];     for (int i=2; i&lt;=n; i++)         f[i]=max(a[i], f[i-1]+a[i]);      for (int i=1; i&lt;=n; i++)         res=max(res, f[i]);     cout&lt;&lt;res; }</pre>
i	1	2	3	4	5	6	7	8	9	10																								
a[i]	1	3	-5	2	7	6	-2	4	-3	1																								
f[i]	1	4	-1	2	9	15	13	17	14	15																								

## Bài 2. TAM GIÁC SỐ

Cho một tam giác gồm các số nguyên không âm.

**Yêu cầu:** Hãy viết chương trình tính tổng lớn nhất của các số nằm trên lộ trình: Tại mỗi bước đi, lộ trình có thể đi xuống phía bên trái hoặc xuống phía bên phải.

**Input :** TAMGIAC.INP gồm:

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

- Dòng đầu tiên chứa số **n** là số hàng trong tam giác ( $1 \leq n \leq 10^3$ ).
- Tiếp theo là **n** dòng, dòng thứ **i** ghi **i** số nguyên  $c_{ij}$ . ( $0 < c_{ij} \leq 10^5$ ).

**Output** : TAMGIAC.OUT có một dòng ghi tổng lớn nhất của các số nằm trên lộ trình.

**Ví dụ :**

TAMGIAC.INP	TAMGIAC.OUT
5 7 3 8 8 1 0 2 7 4 4 4 5 2 6 5	30

Trong ví dụ trên, lộ trình đi qua các điểm 7, 3, 8, 7, 5 có tổng các số là lớn nhất bằng 30.

### Phân tích bài toán

Gọi  $f[i][j]$  là mảng lưu các giá trị tổng của các bước đi, xuất phát từ  $a[1][1]$  và kết thúc tại  $a[i][j]$ . Ta có:

- Cơ sở quy hoạch động  $f[1][1]=a[1][1]$
- Bảng giá trị:

A						F					
i/j	1	2	3	4	5	i/j	1	2	3	4	5
1	7					1	7				
2	3	8				2	10	15			
3	8	1	0			3	18	16	15		
4	2	7	4	4		4	20	25	20	19	
5	4	5	2	6	5	5	24	30	27	26	24

- Công thức truy hồi:  $f[i][j]=\max(f[i-1][j], f[i-1][j-1]) + a[i][j]$  ( $i=2..n; j=1..i$ )
- Truy vết: Ta cộng dồn trên đường đi, do đó số lớn nhất trong dòng  $n$  của mảng  $f$  là kết quả bài toán:  $\text{res}=\max(\text{res}, f[n][j])$  ( $j=1..n$ )

### Code bài Tam giác số

```
#include<bits/stdc++.h>
using namespace std;
const int N=1e3+3;
int a[N][N], f[N][N], n, res;
main(){
    freopen("TAMGIAC.INP", "r", stdin);
    freopen("TAMGIAC.OUT", "w", stdout);
    cin>>n;
    for(int i=1; i<=n; i++){
        for(int j=1; j<=i; j++){
            cin>>a[i][j];
            f[i][j]=max(f[i-1][j-1], f[i-1][j])+a[i][j];
        }
        for(int j=1; j<=n; j++) res=max(res, f[n][j]);
    }
    cout<<res;
```





### Bài 3: DÃY CON ĐƠN ĐIỀU TĂNG DÀI NHẤT (LIQ)

Cho dãy số A có n phần tử:  $a_1, a_2, \dots, a_n$ . Dãy con của A là cách chọn trong A một số phần tử (có thể không liên tiếp) và giữ nguyên thứ tự xuất hiện giữ chúng.

**Yêu cầu:** Tìm dãy con của dãy A là dãy con đơn điệu tăng và có số lượng phần tử nhiều nhất.

**Dữ liệu vào từ tệp daytang.inp.**

- Dòng đầu ghi số n ( $n < 10000$ )
- Dòng thứ hai ghi các số:  $a_1, a_2, \dots, a_n$ . ( $a_i < 1000, i = 1..n$ )

**Dữ liệu ra: ghi vào tệp daytang.out**

- Dòng thứ nhất ghi độ dài dãy con
- Dòng thứ hai ghi dãy con tìm được (có nhiều nghiệm chỉ cần ghi một nghiệm).

**Ví dụ:**

DAYTANG.INP	DAYTANG.OUT
10	5
20 4 3 6 1 10 8 7 18 19	4 6 10 18 19

#### Phân tích bài toán

Gọi  $f[i]$  là số lượng phần tử của dãy con tăng khi xét đến  $a[i]$  (bài toán con có i phần tử).  $tr[i]$  là mảng lưu vết dãy con.

Bổ sung dãy a phần tử  $-oo$  vào đầu dãy ( $a[0] = -oo$ ), phần tử  $+oo$  vào cuối dãy  $a[n+1] = oo$ . Dãy con tăng là dãy con bắt đầu từ  $-oo$  và kết thúc  $+oo$  (hiển nhiên)

- Bài toán con nhỏ nhất chỉ có 1 phần tử vậy có độ dài là 1. Ta chọn  $f[n+1] = 1 \rightarrow$  CS QHĐ

- Xét bài toán con có i phần tử: Ta cần tìm trong số các dãy con có i-1 phần tử có  $a[i-1] > a[i]$  và độ dài lớn nhất ( $f[j_{max}]$ ). Ta có  $f[i] = f[j_{max}] + 1$ ; lưu vết dãy con  $tr[j] = j_{max}$ . Ta có công thức truy hồi:  $f[i] = f[j_{max}] + 1$ ; ( $i = n+1..0$ ;  $j_{max}$  là độ dài lớn nhất của bài toán con nhỏ có i-1 phần tử.

Chẳng hạn: Xét bài toán con có 3 phần tử ta tìm tất cả các bài toán con có 2 phần tử, trong các bài toán con có 2 phần tử ta chọn bài toán con có độ dài lớn nhất và  $a_2 < a_3$ .

- Truy vết: Độ dài dãy con tăng dài nhất là:  $f[0] - 2$  (vì đã thêm hai phần tử:  $-oo, oo$ )

Dự vào mảng tr để truy vết dãy con.

i	0	1	2	3	4	5	6	7	8	9	10	n+1
a[i]	-oo	20	4	3	6	1	10	8	7	18	19	oo
f[i]	7	2	6	6	5	5	4	4	4	3	2	1
tr[i]	2	11	4	4	6	6	9	9	9	10	11	0

Cài đặt thuật toán có nhiều cách cài đặt như qui hoạch động ngược/xuôi.

#### Code quy hoạch động ngược

```
#include <bits/stdc++.h>
using namespace std;
const int N=1e3+5;
```

#### Code quy hoạch động xuôi

```
//code này không truy vết để dễ hiểu.
include <bits/stdc++.h>
using namespace std;
```

<pre> const int oo=INT_MAX; int a[N], tr[N], f[N], n; void doctep(){     cin&gt;&gt;n;     for(int i=1; i&lt;=n; i++) cin&gt;&gt;a[i]; } void qhd(){     a[0]=-oo;     a[n+1]=oo;     f[n+1]=1; // co so qhd     for (int i=n+1; i&gt;=0; i--){         int jmax=0;         for(int j=i+1; j&lt;=n+1; j++){             if (a[j]&gt;=a[i]&amp;&amp; f[j]&gt;f[jmax])                 jmax=j;             f[i]=f[jmax]+1; //cong thuc truy hoi             tr[i]=jmax; //luu vet         }     } } void truyvet(){     cout&lt;&lt;f[0]-2&lt;&lt;endl;     int t=tr[0];     while(t!=n+1){         cout&lt;&lt;a[t]&lt;&lt;" ";         t=tr[t];     } } main(){     freopen("DAYTANG.INP","r",stdin);     freopen("DAYTANG.OUT","w",stdout);     doctep();     qhd();     truyvet(); } </pre>	<pre> const int N=1e3+3; int a[N], f[N], n, res; main(){     freopen("DAYTANG.INP","r", stdin);     freopen("DAYTANG.OUT","w", stdout);     cin&gt;&gt;n;     for(int i=1; i&lt;=n; i++) cin&gt;&gt;a[i];     for(int i=1; i&lt;=n; i++){         f[i]=1;         for(int j=0; j&lt;i; j++)             if(a[i]&gt;a[j])                 f[i]=max(f[i], f[j]+1);         res=max(res, f[i]);     }     cout&lt;&lt;res&lt;&lt;endl; } </pre>
---	--



\* Bài toán LIQ giải bằng QHĐ có độ phức tạp  $O(n^2)$ , còn có cách giải bằng TKNP có độ phức tạp  $O(n \log(n))$ .

#### Bài 4. XÂU CON CHUNG DÀI NHẤT

Xâu ký tự T gọi là xâu con của xâu ký tự S nếu có thể xóa bớt một số ký tự trong xâu S để được xâu T (giữ nguyên thứ tự xuất hiện trong xâu S).

**Yêu cầu:** Cho hai xâu a, b hãy tìm xâu con chung dài nhất (LCS - Longest common subsequence) của hai xâu a và b.

**Dữ liệu:** vào từ tệp văn bản **SUBSTR.INP** có dạng:

- Dòng 1 ghi xâu ký tự a
- Dòng 2 ghi xâu ký tự b.

Các xâu có độ dài không quá 500 và chỉ chứa chữ cái tiếng Anh thường.

**Kết quả:** Đưa ra tệp văn bản **SUBSTR.OUT** chỉ có một dòng ghi một số nguyên là độ dài xâu con chung dài nhất của xâu a và xâu b

**Ví dụ:**

SUBSTR.INP	SUBSTR.OUT
banana oana	3

SUBSTR.INP	SUBSTR.OUT
abcabcaa acbacba	5

### Phân tích bài toán

Gọi  $F[i][j]$  là độ dài xâu con chung của xâu a, xâu b khi xét đến i phần tử đầu tiên của xâu a và j phần tử đầu tiên của xâu b

- CS QHĐ:  $f[0][0]=0$  vì bài toán con nhỏ nhất là xâu a và xâu b không có ký tự nào.

- Bảng giá trị:

i/j	0	1	2	3	4	5	6
0	0	b	a	n	a	n	a
1	o	0	0	0	0	0	0
2	a	0	1	1	1	1	1
3	n	0	1	2	2	3	3
4	a	0	1	2	3	3	3

- Công thức quy hoạch động: ( $n=\text{len}(a)$ ;  $m=\text{len}(b)$ )

+ Nếu  $a[i] \neq b[j]$  thì  $f[i][j] = \max(f[i-1][j], f[i][j-1])$

+ Nếu  $a[i] = b[j]$  thì  $f[i][j] = f[i-1][j-1] + 1$

Với  $i=1..n$ ;  $j=1..m$ ;

- Truy vết:  $f[n][m]$

- Độ phức tạp  $O(n^2)$

### Code bài Xâu con chung dài nhất

```
#include <bits/stdc++.h>
using namespace std;
const int N=1e3+3;
string a, b;
int c[N][N]; // mang qhd
int n, m;
int main() {
    getline(cin, a);
    getline(cin, b);
    n=a.size(); m=b.size();
    a=" "+a; b=" "+b;
    c[0][0]=0;
    for(int i=1; i<=n; i++)
        for(int j=1; j<=m; j++)
            if (a[i]==b[j]) c[i][j]=c[i-1][j-1]+1;
            else c[i][j]=max(c[i-1][j], c[i][j-1]);
    cout<<c[n][m];
}
```

## Bài 5. Dãy con có tổng bằng S

Cho dãy số A có  $n$  phần tử:  $a_1, a_2, \dots, a_n$ . Dãy con của A là cách chọn trong A một số phần tử (có thể không liên tiếp) và giữ nguyên thứ tự xuất hiện giữ chúng.

**Yêu cầu:** Tìm ra một dãy con của dãy A có tổng bằng S cho trước.

**Input:** TONGS.INP gồm:

Dòng đầu tiên ghi hai số nguyên dương  $n$  và  $s$  ( $0 < n \leq 200$ ;  $0 < S \leq 40000$ )

Dòng thứ hai ghi  $n$  số:  $a_1, a_2, \dots, a_n$  ( $0 < a_i \leq 200$ ;  $i=1..n$ )

**Output:** TONGS.INP

- Nếu bài toán vô nghiệm thì ghi duy nhất số 0

- Nếu bài toán có nghiệm thì dòng thứ nhất ghi số độ dài dãy con. Dòng thứ hai ghi dãy con tìm được. Các số ghi cách nhau một dấu cách. Nếu có nhiều phương án thì chỉ cần ghi ra một phương án.

**Ví dụ:**

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

### Phân tích bài toán

Ta dùng kĩ thuật đánh dấu tại các vị trí có tổng sinh ra (tổng các số trong dãy con)

Gọi  $f$  là mảng đánh dấu có  $s$  phần tử

+ CS QHĐ:  $f[0]=1$  vì bài toán con nhỏ nhất  $n=0$ , không có phần tử nào thì có tổng = 0;

+ Công thức truy hồi:  $f[j]=1$  nếu  $f[j]=0$  và  $f[s-a[i]]=1$  ( $i=1..n$ ;  $j=s..a[i]$ )

Khi xét đến  $a[i]$ ; giả sử có tổng sinh ra là  $s$  mà  $s-a[i]$  có tổng sinh ra thì các vị trí  $f[j]$  là số để sinh ra tổng  $s$ .

+ Truy vết: Nếu  $f[s]=1$  thì có tổng sinh ra =  $s$ . Xuất phát từ đây để truy vết bài toán.

### Code bài dãy con có tổng bằng s

```
#include <bits/stdc++.h>
using namespace std;
const int N=200+2;
int a[N], tr[N];
bool f[40002]; // mảng danh dau
int s, n;
void doctep(){
    cin>>n>>s;
    for(int i=1; i<=n; i++) cin>>a[i];
}
void qhd(){
    f[0]=1;
    for (int i=1; i<=n; i++){
        for(int j=s; j>=a[i]; j--){
            if (f[j]==0 && f[j-a[i]]==1){
                f[j]=1;
                tr[j]=i;
            }
        }
        if(f[s]==1) break;
    }
}
```



```
}  
void truyvet(){  
    int p[40002], t, dem=0;  
    if (f[s]==0) cout<<0;  
    else{  
        t=tr[s];  
        while(t>0){  
            dem=dem+1;  
            p[dem]=a[t];  
            s=s-a[t];  
            t=tr[s];  
        }  
        cout<<dem<<endl;  
        for(int i=dem; i>=1; i--) cout<<p[i]<<" ";  
    }  
}  
}  
main(){  
    freopen("TONGS.INP","r",stdin);  
    freopen("TONGS.OUT","w",stdout);  
    doctep();  
    qhd();  
    truyvet();  
}
```



## BÀI 6. BÀI TOÁN CÁI TÚI (CÁI VALY)

Trong siêu thị có  $n$  đồ vật ( $n \leq 1000$ ), đồ vật thứ  $i$  có trọng lượng là  $W[i] \leq 1000$  và giá trị  $V[i] \leq 1000$ . Một tên trộm đột nhập vào siêu thị, tên trộm mang theo một cái túi có thể mang được tối đa trọng lượng  $M$  ( $M \leq 1000$ ).

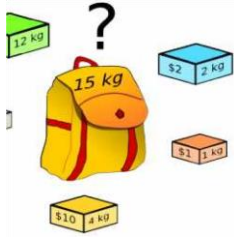
**Yêu cầu:** Hỏi tên trộm sẽ lấy đi những đồ vật nào để được tổng giá trị lớn nhất.

**Input:** file văn bản **VALY.inp**

- Dòng 1:  $n, M$  cách nhau ít nhất một dấu cách
- $n$  dòng tiếp theo: Mỗi dòng gồm 2 số  $V_i, W_i$  là chi phí và giá trị đồ vật thứ  $i$ .

**Output:** file văn bản **VALY.out**: Ghi giá trị lớn nhất tên trộm có thể lấy được.

**Ví dụ: (Mỗi vật chỉ chọn một lần)**

VALY.INP	VALY.OUT	Mô phỏng
5 15 12 4 2 2 1 1 1 2 4 10	15	

**Ví dụ: (Mỗi vật được chọn nhiều lần)**

VALY.INP	VALY.OUT
5 15 12 4 2 2 1 1 1 2 4 10	36

**Phân tích bài toán:** Giá trị của cái túi phụ thuộc vào 2 yếu tố: khối lượng và giá trị, do đó ta dùng mảng hai chiều làm mảng qui hoạch động.

**Trường hợp mỗi vật được chọn một lần:**

- Gọi  $F[i][j]$  là tổng giá trị lớn nhất của cái túi khi xét từ vật 1 đến vật  $i$  và trọng của cái túi chưa vượt quá  $j$ . Với giới hạn  $j$ , việc chọn tối ưu trong số các vật  $\{1, 2, \dots, i-1, i\}$  để có giá trị lớn nhất sẽ có hai khả năng:

+ Nếu không chọn vật thứ  $i$  thì  $F[i][j]$  là giá trị lớn nhất (có thể chọn trong số các vật  $\{1, 2, \dots, i-1\}$ ) với giới hạn trọng lượng là  $j$ , tức là:

$$F[i][j] = F[i-1][j]$$

+ Nếu có chọn vật thứ  $i$  (phải thỏa điều kiện  $W[i] \leq j$ ) thì  $F[i, j]$  bằng giá trị vật thứ  $i$  là  $V[i]$  cộng với giá trị lớn nhất có thể có được bằng cách chọn trong số các vật  $\{1, 2, \dots, i-1\}$  với giới hạn trọng lượng  $j - W[i]$  tức là về mặt giá trị thu được:

$$F[i][j] = V[i] + F[i-1][j - W[i]]$$

Vậy chúng ta phải xem xét xem nếu chọn vật  $i$  hay không chọn vật  $i$  thì sẽ tốt hơn. Từ đó chúng ta có công thức truy hồi như sau.

- $F[0, j] = 0$  (hiển nhiên) – Bài toán con nhỏ nhất.
- $F[i][j] = \max(F[i-1][j], V[i] + F[i-1][j - W[i]])$

**Trường hợp mỗi vật được chọn nhiều lần:** Tương tự như suy luận ở trên ta xét:

Nếu không chọn vật thứ  $i$  thì  $F[i, j]$  là giá trị lớn nhất có thể chọn trong số các vật  $\{1, 2, \dots, i-1\}$  với giới hạn trọng lượng là  $j$ , tức là:

$$F[i, j] = F[i-1, j]$$

Nếu có chọn vật thứ  $i$  (phải thỏa điều kiện  $W[i] \leq j$ ) thì  $F[i, j]$  bằng giá trị vật thứ  $i$  là  $V[i]$  cộng với giá trị lớn nhất có thể có được bằng cách chọn trong số các vật  $\{1, 2, \dots, i\}$  (vì vật  $i$  vẫn có thể được chọn tiếp) với giới hạn trọng lượng  $j - W[i]$  tức là về mặt giá trị thu được:



$$F[i,j]=V[i]+F[i,j-W[i]]$$

Ta có công thức truy hồi như sau:

- $F[0,j] = 0$  (hiển nhiên) – Bài toán con nhỏ nhất.
- $F[i,j] = \max(F[i-1,j], V[i]+F[i,j-W[i]])$

**Truy vết**

**Trường hợp 1:** Trong bảng phương án  $F[n,m]$  chính là giá trị lớn nhất thu được khi chọn trong cả  $n$  vật với giới hạn trọng lượng là  $M$ .

Nếu  $f[n,M]=f[n-1,M]$  thì tức là không chọn vật thứ  $n$ , ta truy về  $f[n-1,M]$ . Còn nếu  $f[n,M] \neq f[n-1,M]$  thì ta thông báo rằng phép chọn tối ưu có chọn vật thứ  $n$  và truy về  $f[n-1, M-W_n]$ .

**Trường hợp 2:** Trong bảng phương án  $F[n,m]$  chính là giá trị lớn nhất thu được khi chọn trong cả  $n$  vật với giới hạn trọng lượng là  $M$ .

Nếu  $f[n,M]=f[n-1,M]$  thì tức là không chọn vật thứ  $n$ , ta truy về  $f[n-1,M]$ . Còn nếu  $f[n,M] \neq f[n-1,M]$  thì ta thông báo rằng phép chọn tối ưu có chọn vật thứ  $n$  và truy về  $f[n, M-W_n]$ .

**Code mẫu:**

Trường hợp mỗi vật chọn một lần	Trường hợp mỗi vật chọn nhiều lần
<pre>#include &lt;bits/stdc++.h&gt; using namespace std; const int N=1e4+4; int m, n, w[N], v[N], f[N][N]; main(){     freopen("VALY.INP", "r", stdin);     freopen("VALY.OUT", "w", stdout);     cin&gt;&gt;n&gt;&gt;m;     for(int i=1; i&lt;=n; i++)         cin&gt;&gt;w[i]&gt;&gt;v[i];      for(int i=1; i&lt;=n; i++)         for(int j=1; j&lt;=m; j++)             if(w[i]&lt;=j) f[i][j]=max(f[i-1][j], v[i]+f[i-1][j-w[i]]);             else f[i][j]=f[i-1][j];      cout&lt;&lt;f[n][m]; }</pre>	<pre>#include &lt;bits/stdc++.h&gt; using namespace std; const int N=1e4+4; int m, n, w[N], v[N], f[N][N]; main(){     freopen("VALY.INP", "r", stdin);     freopen("VALY.OUT", "w", stdout);     cin&gt;&gt;n&gt;&gt;m;     for(int i=1; i&lt;=n; i++)         cin&gt;&gt;w[i]&gt;&gt;v[i];      for(int i=1; i&lt;=n; i++)         for(int j=1; j&lt;=m; j++)             if(w[i]&lt;=j) f[i][j]=max(f[i-1][j], v[i]+f[i][j-w[i]]);             else f[i][j]=f[i-1][j];      cout&lt;&lt;f[n][m]; }</pre>

**BÀI TẬP 12**  
**QUY HOẠCH ĐỘNG**

**Bài 1: LÁT GẠCH**

Cho một hình chữ nhật kích thước  $2 \times N$ , có các viên gạch nhỏ kích thước  $1 \times 2$  và  $2 \times 2$ . Gọi  $K$  là số cách lát các viên gạch vào hình trên sao cho không có phần nào của các viên gạch thừa ra ngoài, cũng không có vùng diện tích nào của hình chữ nhật không được lát.

**Yêu cầu:** Cho trước số nguyên  $N$ . Hãy tính phần dư của  $K$  chia cho 2019

**Dữ liệu:** Vào từ tệp LATGACH.INP gồm:

- Dòng thứ nhất ghi số nguyên  $T$  là số bộ test ( $0 < T \leq 10$ );
- Tiếp theo là  $T$  dòng, mỗi dòng ghi số nguyên dương  $N$  ( $1 \leq N \leq 10^5$ )

**Kết quả:** Ghi ra tệp LATGACH.OUT gồm  $T$  dòng, mỗi dòng chứa số nguyên theo yêu cầu.

**Ví dụ:**

LATGACH.INP	LATGACH.OUT
2	3
3	13
6	

**Ràng buộc:**

- 70% số test tương ứng 70% số điểm có  $N \leq 40$
- 30% số test tương ứng 30% số điểm có  $40 < N \leq 10^5$

**Bài 2: CON KIẾN**

Trên một sân hình chữ nhật  $M \times N$ , được chia thành các ô vuông đơn vị, mỗi ô chứa một lượng thức ăn. Một con kiến xuất phát từ ô  $(1,1)$  muốn đi qua sân để **đến dòng thứ  $M$** . Con kiến chỉ có thể đi theo một dòng chia nhỏ trên sân ứng với một dòng của bảng chữ nhật hoặc đi theo trên một cột của sân.

**Yêu cầu:** Hãy chỉ ra đường đi giúp con kiến có được nhiều thức ăn nhất.

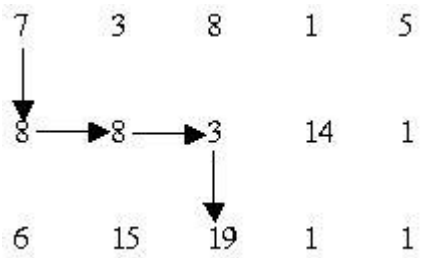
**Dữ liệu vào:** FOOD.INP

Dòng thứ nhất ghi số nguyên  $m, n$  ( $1 < m, n \leq 10^3$ )

Tiếp theo là  $m$  dòng, mỗi dòng ghi  $n$  số nguyên  $a_{ij}$  là lượng thức ăn ở ô  $i, j$  ( $i=1..m; j=1..n$ )

**Kết quả:** FOOD.OUT chỉ có một số nguyên theo yêu cầu.

**Ví dụ:**

FOOD.INP	FOOD.OUT	Giải thích
3 5 7 3 8 1 5 8 8 3 14 1 6 15 19 1 1	45	

**Bài 3: BỐ TRÍ PHÒNG HỌP**

Có  $n$  cuộc họp được đánh số từ 1 đến  $n$  đăng kí làm việc tại một phòng hội thảo. Cuộc họp  $i$  cần bắt đầu vào thời điểm  $a_i$  và thời gian kết thúc là  $b_i$  ( $i = 1..n$ ). Hai cuộc họp có thể

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

nhận phục vụ nếu các khoảng thời gian làm việc tương ứng chỉ có thể giao nhau tại **đầu mút** hoặc **tách rời nhau**.

**Yêu cầu:** Hãy tìm một lịch cho phòng hội thảo để có thể phục vụ nhiều cuộc họp nhất.

**Dữ liệu vào:** cho trong tệp văn bản HOP.INP gồm:

- Dòng đầu tiên là giá trị **n** ( $n \leq 10000$ ).
- Dòng thứ **i** trong **n** dòng tiếp theo ghi 2 số **a<sub>i</sub>** và **b<sub>i</sub>** cách nhau ít nhất một dấu cách. ( $a_i, b_i \leq 32000$  và là các số nguyên dương).

**Kết quả:** Ghi ra tệp văn bản HOP.OUT chỉ có một dòng ghi số **k** là số cuộc họp tối đa có thể bố trí được.

**Ví dụ:**

HOP.INP	HOP.OUT
5	3
1 3	
2 4	
1 6	
3 5	
7 9	

**Bài toán 4: CHỢ CÁ** (olympic BalKan 2000)

Người đánh cá Clement bắt được **n** con cá, khối lượng mỗi con là  $a_i$ , đem bán ngoài chợ. Ở chợ cá, người ta không mua cá theo từng con mà mua theo một lượng nào đó. Chẳng hạn 4kg, 6kg... Ví dụ: có 3 con cá, khối lượng lần lượt là: 3, 2, 4. Mua lượng 6 kg sẽ phải lấy con cá thứ 2 và thứ 3. Mua lượng 3 kg thì lấy con thứ nhất. Không thể mua lượng 8 kg. Nếu bạn là người đầu tiên mua cá, có bao nhiêu lượng bạn có thể chọn?

**Inputdata:** file văn bản **market.inp**

- Dòng 1 : 1 số nguyên dương **N** duy nhất ( $n \leq 10^3$ )
- Dòng 2 : ghi **n** số nguyên  $a_1, a_2, \dots, a_n$  là khối lượng từng con cá ( $0 < a_i \leq 10^3$ )

**Outputdata:** file văn bản **market.out**: Cho biết tổng lượng bạn có thể mua

Market.inp	Market.out
3 2 3 4	7

**Bài 5: SIÊU THỊ 0 ĐỒNG** (HSG 2020)

Siêu thị phục vụ các mặt hàng thực phẩm thiết yếu cho những người có hoàn cảnh kinh tế khó khăn, với khối lượng cho mỗi người không vượt quá **m** kg thì không phải trả tiền. Trong siêu thị có **n** mặt hàng (mỗi mặt hàng có số lượng không hạn chế), mặt hàng thứ **i** có khối lượng **w<sub>i</sub>** kg và có giá trị **v<sub>i</sub>** đồng. Ông Năm đi siêu thị dự định chọn một hoặc một số mặt hàng sao cho tổng khối lượng không vượt quá **m** kg để được hưởng chính sách miễn phí của siêu thị. Tuy nhiên, vì có nhiều cách lựa chọn khác nhau nên ông Năm đang phân vân không biết chọn như thế nào để tổng giá trị các mặt hàng được chọn là lớn nhất nhưng vẫn đảm bảo tổng khối lượng không vượt quá **m** kg.

**Yêu cầu:** Hãy lập trình giúp ông Năm tìm một phương án lựa chọn các mặt hàng để có tổng giá trị lớn nhất và tổng khối lượng không vượt quá **m** kg.



Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

**Dữ liệu vào:** Cho từ tệp văn bản SIEUTHI.INP gồm :

- Dòng thứ nhất chứa hai số nguyên  $n, m$  ( $1 \leq n \leq 1000, 1 < m \leq 10^5$ ).
- Tiếp theo là  $n$  dòng, dòng thứ  $i$  chứa hai số nguyên  $w_i$  và  $v_i$  là khối lượng và giá trị của mặt hàng thứ  $i$  ( $1 \leq i \leq n, 1 \leq w_i, v_i \leq 10^5$ ).

**Kết quả:** Ghi vào tệp văn bản SIEUTHI.OUT gồm một dòng ghi một số nguyên là tổng giá trị các mặt hàng đã chọn.

**Ví dụ:**

SIEUTHI.INP	SIEUTHI.OUT
4 10	108
5 4	
1 9	
8 90	
2 16	

**Giải thích:** Ông Năm chọn mặt hàng thứ hai với số lượng là 2 và mặt hàng thứ ba với số lượng là 1. Tổng khối lượng =  $2 \times 1 + 1 \times 8 = 10$ , tổng giá trị =  $2 \times 9 + 1 \times 90 = 108$ .

**Ràng buộc:**

- Có 20% số test ứng với 20% số điểm có giá trị  $n = 2, m \leq 10^3$ .
- Có 60% số test ứng với 60% số điểm có giá trị  $1 < n, m \leq 10^3$ .
- Có 20% số test ứng với 20% số điểm có giá trị  $1 < n \leq 10^3, 10^3 < m \leq 10^5$ .



**BÀI TẬP 13**  
**ÔN TẬP QUYỀN 1**

**KỲ THI CHỌN HỌC SINH GIỎI LỚP 12 CẤP TỈNH**  
**NĂM HỌC 2016 - 2017**

**Bài 1:** (6,0 điểm) **CÁC ĐIỂM ẢNH**

Trong máy tính, một bức ảnh được mã hóa thành các điểm ảnh, mỗi điểm ảnh được biểu thị bởi một màu, mỗi màu được quy định bằng một mã màu, mã màu là một số nguyên dương nằm trong dãy từ 1 đến  $10^6$ .

**Yêu cầu:** Cho trước một bức ảnh được mã hóa thành  $n$  điểm ảnh và mã màu của từng điểm ảnh.

- a) Hãy cho biết có bao nhiêu mã màu khác nhau xuất hiện trong bức ảnh.
- b) Số lượng điểm ảnh có mã màu giống nhau nhiều nhất là bao nhiêu?

**Dữ liệu vào:** Cho từ tệp văn bản PIXEL.INP gồm hai dòng:

- Dòng thứ nhất ghi số nguyên dương  $n$  ( $1 \leq n \leq 10^6$ )
- Dòng thứ hai ghi  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  với  $a_i$  là mã màu của điểm ảnh thứ  $i$  ( $1 \leq a_i \leq 10^6; i=1..n$ )

Các số ghi trên cùng một dòng cách nhau một kí tự trắng.

**Kết quả:** Ghi ra tệp văn bản PIXEL.OUT gồm hai dòng:

- Dòng thứ nhất ghi một số nguyên là số lượng mã màu khác nhau xuất hiện trong bức ảnh.

- Dòng thứ hai ghi số nguyên là số lượng điểm ảnh có mã màu giống nhau nhiều nhất.

**Ví dụ:**

PIXEL.INP	PIXEL.OUT
6	4
1 9 2 3 1 1	3

**Ràng buộc dữ liệu:**

- Có 70% số test tương ứng 70% số điểm có  $2 \leq n \leq 10^3$
- Có 30% số test tương ứng 30% số điểm có  $10^3 \leq n \leq 10^6$

## **Bài 2:** (7,0 điểm) **CÁC HỘP SỮA**

Một cửa hàng bán sữa hiện có  $n$  hộp sữa trong kho. Mỗi ngày cửa hàng phải đem giao  $k$  hộp sữa cho khách hàng nếu trong kho còn từ  $k$  hộp trở lên, và nếu còn ít hơn thì sẽ giao hết các hộp sữa. Nhưng có một vấn đề với các hộp sữa là ngày hết hạn của chúng. Mỗi hộp sữa có một ngày hết hạn mà sau ngày đó thì hộp sữa không thể dùng được và phải bỏ đi. Cửa hàng không muốn những hộp sữa bị quá hạn, vì thế khi giao một hộp sữa cho khách hàng, cửa hàng sẽ chọn hộp sữa có hạn sử dụng còn ít hơn để giao trước. Chiến lược này sẽ giúp cửa hàng hạn chế tối đa các hộp sữa không giao kịp và bị quá hạn.

Biết rằng, cửa hàng đã biết hạn sử dụng của mỗi hộp sữa. Hạn sử dụng được thể hiện bằng một con số cho biết số ngày còn lại mà hộp sữa còn dùng được. Chẳng hạn, hạn sử dụng là **1** cho biết hộp sữa phải giao trong ngày hôm nay, số **2** là không thể để trễ hơn ngày mai, ...

**Yêu cầu:**

- Hãy cho biết có ít nhất bao nhiêu hộp sữa bị quá hạn sử dụng phải bỏ đi vì cửa hàng không giao kịp.
- Để các hộp sữa không bị quá hạn sử dụng thì mỗi ngày cửa hàng cần phải giao được ít nhất bao nhiêu hộp sữa?

**Dữ liệu vào:** Cho từ tệp văn bản MILK.INP gồm hai dòng:

- Dòng thứ nhất ghi hai số nguyên dương  $n, k$  là số hộp sữa hiện có và số hộp sữa mà cửa hàng phải giao mỗi ngày ( $1 \leq k, n \leq 10^6$ ).
- Dòng thứ hai ghi  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  trong đó  $a_i$  là hạn sử dụng của hộp sữa thứ  $i$  ( $0 < a_i \leq 10^7, i=1..n$ ).

Các số ghi trên cùng một dòng cách nhau một kí tự trắng.

**Kết quả:** Ghi ra tệp văn bản MILK.OUT gồm hai dòng:

- Dòng thứ nhất ghi số hộp sữa bị quá hạn sử dụng ít nhất nếu mỗi ngày cửa hàng giao được  $k$  hộp sữa.
- Dòng thứ hai ghi số hộp sữa ít nhất mà mỗi ngày cửa hàng cần phải giao được để không có hộp sữa nào bị quá hạn sử dụng.

**Ví dụ:**

MILK.INP	MILK.OUT
6 2	1
2 1 1 2 3 2	3

*Giải thích:* Ngày thứ nhất (hôm nay) giao hai hộp sữa thứ 2 và thứ 3 vì hết hạn trong ngày, ngày thứ hai giao hộp sữa thứ 1 và thứ 4, ngày thứ ba chỉ giao một hộp thứ 5 và phải bỏ

Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

hộp sữa thứ 6 đã hết hạn sử dụng. Để các hộp sữa không bị quá hạn sử dụng thì mỗi ngày cửa hàng cần phải giao được ít nhất 3 hộp sữa.

**Bài 3:** (7,0 điểm) **GẤP GIẤY**

Trong một buổi sinh hoạt ngoại khóa, để tăng thêm phần hào hứng, ban tổ chức đã đưa ra trò chơi gấp giấy. Trò chơi bao gồm một tờ giấy hình vuông kích thước  $n \times n$  được kẻ thành lưới các ô vuông đơn vị, mỗi ô vuông được ban tổ chức ghi sẵn một trong hai số 0 hoặc 1. Mỗi người tham gia trò chơi được ban tổ chức yêu cầu gấp tờ giấy một lần theo đường kẻ ngang và một lần theo đường kẻ dọc. Theo nếp gấp đó sẽ chia tờ giấy thành 4 phần. Người thắng cuộc là người tìm được cách gấp tờ giấy mà phần có ít số 1 nhất (trong 4 phần của tờ giấy sau khi gấp) là nhiều nhất so với các cách gấp khác.

**Yêu cầu:** Bạn hãy tìm một cách gấp giấy thỏa mãn yêu cầu của ban tổ chức sao cho bạn là người thắng cuộc.

**Dữ liệu vào:** Cho từ tệp văn bản GAPGIAY.INP có dạng:

- Dòng thứ nhất ghi số nguyên dương  $n$  ( $2 \leq n \leq 250$ )
- Tiếp theo là  $n$  dòng, mỗi dòng ghi  $n$  số: số 0 hoặc số 1, tương ứng với các số được ghi trên tờ giấy.

Các số ghi trên cùng một dòng cách nhau một kí tự trắng.

**Kết quả:** Ghi vào tệp văn bản GAPGIAY.OUT gồm một dòng ghi một số nguyên là số lượng số 1 trên phần giấy có ít số 1 nhất của cách gấp tìm được.

**Ví dụ:**

GAPGIAY.INP	GAPGIAY.OUT	Minh họa
6 1 0 1 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 0 1 0 0 0	3	

**KỲ THI CHỌN HỌC SINH GIỎI LỚP 12 CẤP TỈNH  
NĂM HỌC 2017 - 2018**

**Tổng quan đề thi:**

Tên bài	tệp chương trình	tệp dữ liệu vào	tệp dữ liệu ra
Bài 1. Đếm ước số	BL1.*	UOCSO.INP	UOCSO.OUT
Bài 2. Đua Robot	BL2.*	ROBOT.INP	ROBOT.OUT
Bài 3. Tuyển dò ngang	BL3.*	TUYENDO.INP	TUYENDO.OUT

**Bài 1.** (6,0 điểm) **ĐẾM ƯỚC SỐ**



Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

Bạn Nam rất thích học toán, đặc biệt là các bài toán về tổ hợp. Các công thức tổ hợp như đếm số hoán vị, số chỉnh hợp hay số tổ hợp đều cần phải tính các giá trị giai thừa. Như ta đã biết, giai thừa của một số nguyên dương  $N$  (ký hiệu là  $N!$ ) bằng tích của các số tự nhiên từ 1 đến  $N$ , chẳng hạn  $4! = 1 \times 2 \times 3 \times 4 = 24$ , hay  $6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 = 720$ . Khi tính toán các giá trị giai thừa như vậy, bạn Nam nhận thấy kết quả tính được có số lượng ước số nhiều hơn số thừa số tạo ra kết quả đó. Ví dụ,  $4!$  là tích của bốn thừa số 1, 2, 3 và 4 nhưng kết quả là 24 thì lại có tới 8 ước số là 1, 2, 3, 4, 6, 8, 12, 24.

**Yêu cầu:** Với một số nguyên dương  $N$ , bạn hãy lập trình giúp bạn Nam tính xem giá trị  $N!$  có bao nhiêu ước số.

**Dữ liệu vào:** Cho từ tệp văn bản UOCSO.INP gồm một dòng ghi số nguyên dương  $N$ .

**Kết quả:** Ghi vào tệp văn bản UOCSO.OUT gồm một dòng ghi kết quả tính được.

**Ví dụ:**

UOCSO.INP	UOCSO.OUT
4	8
6	30

**Giới hạn dữ liệu:**

- Có 70% số điểm ứng với giá trị  $N \leq 10$
- Có 20% số điểm ứng với giá trị  $N \leq 15$
- Có 10% số điểm ứng với giá trị  $N \leq 20$

## Bài 2. (7,0 điểm) ĐUA ROBOT

Câu lạc bộ Robocon của trường vừa tổ chức cuộc đua Robot. Có  $N$  đội tham gia cuộc đua, mỗi đội sẽ lập trình cho Robot di chuyển trên một đường thẳng. Ban đầu tất cả Robot của các đội đặt tại vạch xuất phát được đánh tọa độ là 0. Robot có thể được lập trình để di chuyển về phía trước hoặc đi ngược về phía sau. Dọc theo đường đi, ban tổ chức đặt  $M$  trạm kiểm soát, trạm kiểm soát thứ  $i$  đặt tại vị trí có tọa độ  $V_i$  ( $|V_i| \leq 10^9, i=1..M$ ). Sau thời gian di chuyển, Robot thứ  $j$  dừng lại tại vị trí có tọa độ  $R_j$  ( $|R_j| \leq 10^9, j=1..N$ ) và điểm được tính bằng tổng khoảng cách từ Robot đến tất cả các trạm kiểm soát.

**Yêu cầu:** Hãy lập trình tính điểm cao nhất của Robot tại vị trí dừng lại.

**Dữ liệu vào:** Cho từ tệp văn bản ROBOT.INP có dạng:

- Dòng đầu ghi hai số nguyên dương  $N$  và  $M$ .
- Dòng thứ hai ghi  $N$  số nguyên  $R_1, R_2, \dots, R_N$
- Dòng thứ ba ghi  $M$  số nguyên  $V_1, V_2, \dots, V_M$

Các số trên cùng một dòng ghi cách nhau ít nhất một khoảng cách.

**Kết quả:** Ghi vào tệp văn bản ROBOT.OUT gồm một dòng ghi điểm số cao nhất tính được.

**Ví dụ:**

ROBOT.INP	ROBOT. OUT
3 3 -2 3 2 2 -3 -1	11

**Giải thích:** Robot thứ 2 có điểm số cao nhất là 11



Trường THPT chuyên Nguyễn Quang Diêu, tỉnh Đồng Tháp  
**Tài liệu môn Tin học – Chương trình Nâng cao và Chuyên sâu**

- Robot thứ 1 ở vị trí -2 cách trạm 1 là **4**, cách trạm 2 là **1**, cách trạm 3 là **1**, tổng khoảng cách **6**.
- Robot thứ 2 ở vị trí 3 cách trạm 1 là **1**, cách trạm 2 là **6**, cách trạm 3 là **4**, tổng khoảng cách **11**.
- Robot thứ 3 ở vị trí 2 cách trạm 1 là **0**, cách trạm 2 là **5**, cách trạm 3 là **3**, tổng khoảng cách **8**.

**Giới hạn dữ liệu:**

- Có 60% số điểm ứng với giá trị  $N, M \leq 10^4$
- Có 40% số điểm ứng với giá trị  $N, M \leq 10^6$

**Bài 3. (7,0 điểm) TUYẾN ĐÒ NGANG**

Sông Tiền là một trong hai nhánh sông lớn của dòng sông MeKong. Trước đây, việc giao thông qua lại giữa hai bờ sông chủ yếu bằng phà và các tuyến đò ngang. Từ năm 2000, cầu Mỹ Thuận đã được đưa vào sử dụng và sắp tới đây sẽ tiếp tục khánh thành cầu Cao Lãnh nối liền hai bờ sông Tiền. Có thể nói, những công trình này cùng với cầu Cần Thơ và cầu Vàm Cống (cũng sắp hoàn thành) đã góp phần rất lớn trong việc phát triển kinh tế của vùng ĐBSCL, kết nối vùng ĐBSCL gần hơn với TP.HCM và các tỉnh miền Đông Nam bộ. Mặc dù vậy, chúng ta cũng không thể phủ nhận vai trò của những tuyến đò ngang hiện nay vì sự tiện lợi của nó thay vì phải đi vòng những đoạn đường xa để qua sông bằng cầu.

Hiện tại, dọc theo sông Tiền có  $N$  tuyến đò ngang đang hoạt động, mỗi tuyến đò đi từ một bến đò ở bờ Bắc sông Tiền sang một bến ở bờ Nam sông Tiền và ngược lại. Các bến đò ở mỗi bờ được đánh số từ 1 đến  $N$  theo thứ tự từ thượng nguồn xuống hạ nguồn. Biết rằng, nếu tuyến đò xuất phát từ bến đò  $i$  ở bờ Bắc thì sẽ đi sang bến đò  $a_i$  ở bờ Nam ( $i=1..N, a_i=1..N, a_i \neq a_j \forall i \neq j$ ). Vấn đề phát sinh là giữa một số tuyến đò có hành trình khi sang sông bị giao cắt nhau và như vậy có thể không an toàn cho hành khách vì có thể xảy ra tai nạn va chạm trên sông. Chính vì vậy chính quyền địa phương dự định sẽ cho dừng hoạt động một số tuyến đò, chỉ giữ lại những tuyến đò mà hành trình của nó không giao cắt với hành trình của những tuyến khác được giữ lại.

**Yêu cầu:** Hãy tìm một phương án dừng hoạt động một số ít nhất các tuyến đò sao cho những tuyến đò hoạt động còn lại có hành trình không giao cắt nhau.

**Dữ liệu vào:** Cho từ tệp văn bản TUYENDO.INP có dạng:

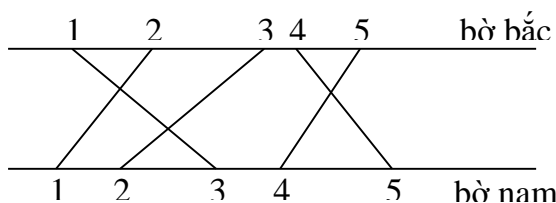
- Dòng thứ nhất ghi số nguyên dương  $N$
- Dòng thứ hai ghi  $N$  số nguyên dương  $a_1, a_2, \dots, a_N$ . Giữa các số cách nhau một khoảng cách.

**Kết quả:** Ghi vào tệp văn bản TUYENDO.OUT gồm một dòng ghi một số nguyên là số tuyến đò ít nhất cần phải dừng hoạt động.

**Ví dụ:**

TUYENDO.INP	TUYENDO. OUT
5 3 1 2 5 4	2

**Giải thích:** Có ít nhất 2 tuyến đò phải dừng hoạt động là tuyến 1 và 4 hoặc tuyến 1 và 5.



**Giới hạn dữ liệu:**

- Có 20% số điểm ứng với giá trị  $N \leq 20$

- Có 50% số điểm ứng với giá trị  $N \leq 10^3$
- Có 30% số điểm ứng với giá trị  $N \leq 10^5$

## PHỤ LỤC

- [1] Phần mềm chấm bài tự động môn Tin học:  
<https://dsapblog.wordpress.com/2013/12/24/themis/>
- [2] Code mẫu các thuật toán:  
<https://sites.google.com/site/kc97ble/home>
- [3] Trang giải bài trực tuyến của ĐH Nha Trang:  
<http://ntucoder.net/Home>
- [4] Trang giải bài trực tuyến của nhóm học sinh giỏi quốc gia thành lập:  
<http://vn.spoj.com/problems/main/>
- [5] fanpage hỏi bài:  
<https://www.facebook.com/groups/163215593699283/>
- [6] Trang giải bài trực tuyến (ngoài nước):  
<http://codeforces.com>

