

Chủ biên: Đỗ Thị Hồng Ngát

Programming for newbies



CODE DREAM

Đoàn Gia Nhu



PROGRAMMING FOR NEWBIES

TRUNG TÂM ĐÀO TẠO LẬP TRÌNH CODEDREAM

PROGRAMMING FOR NEWBIES

(Lưu hành nội bộ)

Năm 2023

LỜI NÓI ĐẦU

Ngôn ngữ lập trình C⁺⁺ được phát triển bởi Bjarne Stroustrup, là môn cơ sở ngành ở hầu hết các trường đào tạo Công nghệ thông tin ở Việt Nam, giúp người học có nền tảng kỹ thuật lập trình, các kiến thức cơ bản và tư duy lập trình. Ngôn ngữ C⁺⁺ có nhiều ứng dụng lập trình phổ biến như: game, giao diện người dùng, Web, đồ họa, phần mềm quản trị cơ sở dữ liệu, trình biên dịch...

Từ tháng 9/2020 các trường THCS, THPT có thể chọn một trong hai ngôn ngữ để dạy lập trình cho học sinh là: Python hoặc C/C⁺⁺. Ở đa số các học viện, trường đại học, cao đẳng, trung cấp hiện cũng tập trung đào tạo ngôn ngữ C/C⁺⁺ hoặc Python. So với Python thì C/C⁺⁺ có từ khóa dài, code phức tạp hơn, nên khó học và khó dùng hơn, nhưng ngôn ngữ C/C⁺⁺ có tốc độ nhanh, kiểu dữ liệu rõ ràng, nếu như làm chủ được ngôn ngữ nền tảng như ngôn ngữ C/C⁺⁺ thì việc học các ngôn ngữ khác sẽ trở nên dễ dàng hơn. Hơn nữa, ngôn ngữ C/C⁺⁺ cũng hỗ trợ rất nhiều thư viện giúp bạn đọc dễ dàng cài đặt các chương trình liên quan đến thuật toán.

Trước nhu cầu thiết thực của bạn đọc, tác giả Đỗ Thị Hồng Ngát đã xuất bản cuốn sách "*Programming for newbies*" nhằm đáp ứng một cách tối ưu cho mỗi học sinh, sinh viên, bạn đọc quan tâm đến ngành Công nghệ thông tin.

“*Programming for newbies*” tập hợp hơn 200 bài tập lập trình cơ bản, được chia thành 6 phần: Nhập - Xuất, Cấu trúc rẽ nhánh, Cấu trúc lặp, Xử lý chuỗi, Dữ liệu kiểu mảng, Đệ quy và bài tập nâng cao nhằm cung cấp cho các bạn mới tìm hiểu về lập trình có một lộ trình tổng quan và môi trường học tập tốt nhất. Thông qua việc làm bài tập trực quan, các bạn mới bắt đầu học lập trình sẽ hiểu được các khái niệm lập trình cơ bản cũng như hệ thống câu lệnh.

Với kinh nghiệm nhiều năm giảng dạy tại các trường THPT, các đội tuyển học sinh giỏi của quận, thành phố Hà Nội, tác giả đã biên soạn một cách khoa học, trình bày tóm tắt những kiến thức cơ bản kết hợp với những bài tập từ dễ đến khó, từ cơ bản đến nâng cao để bạn đọc dễ dàng thực hành theo. Các lý thuyết và ví dụ trong tài liệu được thể hiện bằng ngôn ngữ C++, tuy nhiên người sử dụng hoàn toàn có thể luyện tập bằng các ngôn ngữ khác như Python, Java... Bên cạnh đó, bạn đọc có thể liên hệ với tác giả để có tài khoản và gửi mã nguồn bài thực hành lên website chấm bài tự động của Code Dream qua đường link <https://oj.codedream.edu.vn>, bạn đọc cũng có thể tham gia cộng đồng lập trình cùng Code Dream qua đường link <https://www.facebook.com/groups/hoc.lap.trinh.codedream> để giao lưu, trao đổi về những thông tin hữu ích về lập trình, công nghệ, thuật toán... hỏi đáp các bài tập thực hành trong sách cũng như đóng góp những ý kiến góp ý cho tác giả về cuốn sách.

Cuốn sách thực sự là tài liệu bổ ích dành cho đông đảo đối tượng độc giả học sinh, sinh viên mới làm quen cùng những bạn đọc yêu thích công nghệ thông tin có nhu cầu tìm hiểu và học tập.

Mặc dù đã có nhiều cố gắng trong quá trình biên soạn, song cuốn sách sẽ khó tránh khỏi những thiếu sót. Rất mong nhận được ý kiến đóng góp của quý độc giả để cuốn sách được hoàn thiện hơn trong những lần xuất bản sau. Mọi ý kiến góp ý của quý độc giả xin gửi về địa chỉ email: info@codedream.edu.vn; điện thoại: 0971 730 550.

Xin trân trọng giới thiệu cùng bạn đọc./.

Tác giả

Đỗ Thị Hồng Ngát

KIẾN THỨC CHUNG

Chương trình C++ đơn giản:

```
#include <iostream>
using namespace std;
int main(){
    cout << "Welcome to Code Dream!"
    return 0;
}
```

1. Kiểu dữ liệu (Data type)

Trong lập trình C/C++ (hoặc các ngôn ngữ khác), kiểu dữ liệu chính là phần xác định các giá trị mà một biến có thể nhận hay giá trị mà một hàm có thể trả về.

Mỗi kiểu dữ liệu trong C/C++ bắt buộc phải có số lượng bộ nhớ nhất định và có các hoạt động cụ thể có thể thực hiện trên biến số tương ứng.

Một số kiểu dữ liệu thông dụng trong C/C++:

Lưu ý: các khoảng giá trị biểu diễn của các kiểu dữ liệu đều là các số liệu tương đối.

Kiểu dữ liệu	Trong C++	Kích thước (bytes)	Khoảng giá trị biểu diễn được
Số nguyên	int	4	$[-2 \times 10^9; 2 \times 10^9]$
	long long	8	$[-2^{63}; 2^{63}-1]$
Số thực	float	4	$[3,4 \times 10^{-38}; 3,4 \times 10^{38}]$
	double	8	$[1,7 \times 10^{-308}; 1,7 \times 10^{308}]$
Boolean	bool	1	$[0;1]$ tương ứng False/True
Ký tự	char	1	Dùng để lưu các ký tự trong bảng mã ASCII

2. Biến số (Variables)

Một biến số trong ngôn ngữ lập trình là một vùng trong bộ nhớ được cấp phát để lưu trữ một giá trị hoặc dữ liệu cụ thể. Mỗi biến có một tên định danh duy nhất để chúng ta có thể truy cập và thay đổi giá trị của nó trong quá trình thực thi chương trình.

Biến cho phép chúng ta làm việc với dữ liệu, tính toán và thực hiện các tác vụ phức tạp trong chương trình. Giá trị của biến có thể thay đổi trong quá trình chạy.

Quy tắc đặt tên biến:

- Tên biến bao gồm các chữ cái, chữ số và dấu gạch dưới “_”
- Tên biến phải được bắt đầu bởi chữ cái hoặc dấu gạch dưới “_”

Ví dụ: Tên biến đúng là “_a”, “a_1”, “a1”, tên biến sai là “1a”, “a b”.

- Không đặt tên biến trùng với các từ khoá: *const*, *char*, *int*...
- Hai biến trong cùng phạm vi không được trùng tên nhau
- Đặt tên có ý nghĩa, ngắn gọn, dễ hiểu, có thể viết thường và dùng dấu “_” để ngăn cách giữa các từ hoặc viết hoa chữ đầu tiên của mỗi từ.

Ví dụ: *full_name*, *fullName*...

Cú pháp:

kiểu_dữ_liệu_tên_biến;

Trong đó:

- *kiểu_dữ_liệu*: Là kiểu dữ liệu của biến (ví dụ: *int*, *double*, *char*, ...).
- *tên_biến*: Là tên của biến.

Ví dụ:

- Khai báo biến *a* thuộc kiểu số nguyên: *int a;*

- Khai báo biến c thuộc kiểu ký tự: *char c*;
- Khai báo biến ck thuộc kiểu boolean: *bool ck*;
- Khai báo ba biến a, b, c thuộc kiểu số nguyên: *int a, b, c*;

3. Hằng số (Constant)

Hằng số là một giá trị không thay đổi trong suốt quá trình thực thi của chương trình. Nó là một giá trị cố định và không thể được thay đổi sau khi đã được định nghĩa.

Hằng số thường được sử dụng để đặt tên cho các giá trị cố định, như các hằng số toán học, cài đặt chương trình hoặc các giá trị không đổi khác mà chương trình cần sử dụng.

Cú pháp:

const kiểu_dữ_liệu tên_hằng = giá_trị;

Ví dụ: Khai báo hằng số PI có giá trị bằng 3,14 như sau:

const double PI = 3.14;

4. Phép toán (Operation)

Phép toán là một hành động hoặc quá trình được thực hiện trên các giá trị để tạo ra kết quả mới. Các phép toán có thể thực hiện trên các số, chuỗi ký tự, danh sách, và nhiều kiểu dữ liệu khác nhau, nhằm thực hiện các phép tính, so sánh hoặc thay đổi dữ liệu.

Dưới đây là bảng chuyển đổi một số phép toán trong toán học sang phép toán trong C++:

Phép toán	Trong toán học	Trong C++
<i>Phép toán số học</i>		
Phép cộng	+	+
Phép trừ	-	-
Phép nhân	x hoặc .	*
Phép chia	:	/
Phép chia lấy phần dư	mod	%

<i>Phép toán so sánh</i>		
Lớn hơn	$>$	$>$
Lớn hơn hoặc bằng	\geq	$>=$
Nhỏ hơn	$<$	$<$
Nhỏ hơn hoặc bằng	\leq	$<=$
Bằng	$=$	$==$
Không bằng	\neq	$!=$
<i>Phép toán logic</i>		
and	\wedge hoặc \cap	$\&\&$
or	\vee hoặc \cup	$\ \ $
not	\neg hoặc \sim	$!$

5. Biểu thức (Expression)

Biểu thức là một dãy các giá trị, biến, hằng số, phép toán và các thành phần khác được kết hợp lại để tạo thành một giá trị mới hoặc thực hiện một phép tính.

Cú pháp:

Biểu_thức_1 phép_toán Biểu_thức_2

Một số loại biểu thức thông dụng:

- Biểu thức số học là các biểu thức liên quan đến các phép tính số học như cộng, trừ, nhân, chia, lũy thừa, căn bậc hai và các phép toán khác.

Ví dụ: $x + y$

$a - b$

$3 * x$

$1 / k$

- Biểu thức so sánh là biểu thức dùng để so sánh giữa các giá trị như lớn hơn, nhỏ hơn, bằng nhau, lớn hơn hoặc bằng, nhỏ hơn hoặc bằng, khác nhau. Giá trị của biểu thức sẽ trả về kết quả là True hoặc False.

Ví dụ: $1 > 2$ (biểu thức này sẽ cho ra kết quả là *False*)

$1 \leq 2$ (biểu thức này sẽ cho ra kết quả là *True*)

$x == y$ (biểu thức này sẽ cho ra kết quả là *True* hay *False* phụ thuộc vào x có bằng y hay không)

- Biểu thức logic là các biểu thức liên quan đến logic như phép and, or, not để kiểm tra điều kiện và quản lý logic trong chương trình.

Ví dụ: $(1 > 2) \parallel (3 <= 4)$

$(x + y < 10) \&\& (x > 0)$

6. Phép gán (Assignment)

Phép gán là quá trình gán một giá trị hoặc một biểu thức cho một biến, cho phép lưu trữ và quản lý dữ liệu trong chương trình. Phép gán thường được thực hiện bằng cách sử dụng toán tử gán (=).

Cú pháp:

tên_biến = biểu_thức ;

Ví dụ: $x = 4$ (sau câu lệnh này, x sẽ có giá trị bằng 4)

$x = a + b$ (sau câu lệnh này, x sẽ có giá trị bằng giá trị của biểu thức $a + b$)

- Khi một biểu thức chứa nhiều phép toán được thực hiện, thứ tự thực hiện của các phép toán sẽ tuân theo một số quy tắc và ưu tiên. Các quy tắc này giúp định rõ thứ tự mà các phép toán sẽ được thực hiện để tính toán đúng kết quả của biểu thức.

Thứ tự thực hiện của các phép toán trong ngôn ngữ C/ C++ của một biểu thức thường tuân theo các nguyên tắc sau đây:

- Dấu ngoặc đơn: Các phép toán trong dấu ngoặc đơn () sẽ được thực hiện trước. Các dấu ngoặc này cho phép chương trình xác định thứ tự ưu tiên của các phép toán.

- Phép nhân (*), phép chia (/) và phép chia lấy phần dư (%) được tính theo thứ tự từ trái qua phải.

- Phép cộng (+) và phép trừ (-) được tính theo thứ tự từ trái qua phải.

- Phép so sánh và logic: Các phép so sánh (<, <=, >, >=, ==, !=) và các phép logic (&&, ||, !) được tính sau các phép toán số học.

Lưu ý: Trong biểu thức phức tạp có nhiều phép toán khác nhau, sử dụng dấu ngoặc để xác định rõ thứ tự thực hiện của các phép toán là rất quan trọng.

7. Một số hàm số học chuẩn thông dụng

Ngôn ngữ C/C++ cung cấp một số hàm số học chuẩn thông dụng trong thư viện `<cmath>` của C++ (hoặc `<math.h>` trong C) để thực hiện các phép toán số học phức tạp.

Ngoài ra, trong ngôn ngữ C++ cũng có thể sử dụng thư viện `<bits/stdc++.h>` để thay thế cho `<cmath>`, nó sẽ tự động bao gồm mọi thư viện chuẩn của C++ để sử dụng hầu hết các tính năng của ngôn ngữ mà không cần phải xem xét về việc khai báo riêng từng thư viện.

Dưới đây là một số hàm số học quan trọng và thông dụng trong C++:

Hàm	Cú pháp	Ví dụ	
		Lệnh	Kết quả
Căn bậc hai	<code>sqrt (n)</code>	<code>sqrt (9)</code>	3.0
Lũy thừa	<code>pow (x,y)</code>	<code>pow (2,3)</code>	8.0
Trị tuyệt đối	<code>abs (n)</code>	<code>abs (-2)</code>	2
Lấy cận trên	<code>ceil (n)</code>	<code>ceil (43.0001)</code>	44
		<code>ceil (43.000)</code>	43
Lấy cận dưới	<code>floor (n)</code>	<code>floor (43.999)</code>	43
	<code>atoi (s.c_str())</code>	<code>atoi ("123".c_str())</code>	123

Kiến thức chung

Chuyển chuỗi thành số nguyên kiểu int		atoi ("1.23".c_str())	1
Chuyển chuỗi thành số nguyên kiểu long long	stoll (s)	stoll ("123")	123
		stoll ("1.23")	1
Chuyển chuỗi thành số thực	atof (s)	atof ("1.23")	1.23
		atof ("1.2.3")	1.2

CHƯƠNG 1. NHẬP - XUẤT

1.1. Câu lệnh nhập - xuất

Việc nhập - xuất (input/output) trong ngôn ngữ lập trình rất quan trọng vì nó cho phép chương trình tương tác với người dùng và thế giới bên ngoài.

Chương trình cần tương tác với người dùng thông qua việc đọc dữ liệu từ bàn phím (input) và hiển thị kết quả, thông báo cho người dùng (output). Qua đó chương trình sẽ gửi thông tin cho người dùng và yêu cầu họ cung cấp dữ liệu hoặc thực hiện các thao tác.

1.1.1. Nhập dữ liệu

Để nhập dữ liệu đầu vào, ta sử dụng câu lệnh *cin* và toán tử *>>*.

Ví dụ: Để nhập dữ liệu cho biến, ta sử dụng lệnh *cin* như sau:

```
cin >> a;  
cin >> a >> b >> c;
```

Câu lệnh *cin* sẽ đọc giá trị từ đầu vào cho đến khi gặp dấu cách, dấu xuống dòng (trừ trường hợp đọc kiểu dữ liệu là ký tự), sau đó sẽ gán giá trị vừa đọc được vào biến tương ứng. Có thể đọc nhiều biến cùng một lúc, khi đó tên của mỗi biến được ngăn cách nhau bởi toán tử *>>*.

1.1.2. Xuất dữ liệu

Để hiển thị dữ liệu đầu ra, ta sử dụng câu lệnh *cout* và toán tử *<<*.

Ví dụ: Để hiển thị dữ liệu, ta sử dụng lệnh *cout* như sau:

```
cout << "Welcome to Code Dream";  
cout << "Hello, my name is " << name;
```

Câu lệnh *cout* có thể bao gồm nhiều thành phần, các thành phần được ngăn cách nhau bởi toán tử <<.

- Nếu thành phần đó là chuỗi ký tự (được đặt trong cặp ngoặc kép) thì chương trình sẽ hiển thị ra đúng chuỗi ký tự đó.

- Nếu thành phần đó là biến/biểu thức (không đặt trong cặp ngoặc kép) thì chương trình sẽ hiển thị ra giá trị của biến/biểu thức đó.

1.1.3. Ví dụ

Viết chương trình nhập vào tuổi của bạn và hiển thị ra kết quả "*You are ... year(s) old*". Dấu ... chính là số tuổi bạn vừa nhập.

Chương trình C++ sẽ được viết như sau:

```
#include <iostream>
using namespace std;
int main() {
    int age;
    cout << "Enter your age:" << endl;
    cin >> age;
    cout << "You are " << age << " year(s) old!";
    return 0;
}
```

Giải thích ví dụ:

- Khi người dùng nhập dữ liệu vào, ta cần khai báo một biến để lưu lại giá trị người dùng vừa nhập. Biến *age* được khai báo kiểu *int* (số nguyên) dùng để lưu tuổi mà người dùng vừa nhập.

- Câu lệnh *cin* dùng để đọc giá trị vừa nhập, sau đó biến *age* sẽ được gán giá trị tương ứng.

- Câu lệnh *cout* sẽ in ra màn hình theo yêu cầu đề bài:

- + Những dữ liệu được đặt trong cặp ngoặc kép (") sẽ được hiểu là chuỗi và hiển thị như trong code.

+ Những dữ liệu không đặt trong cặp ngoặc kép được hiểu là biến/biểu thức và chương trình sẽ hiển thị ra giá trị của biến/biểu thức đó.

1.2. Ép kiểu dữ liệu trong lập trình C/C++

- Ép kiểu (casting) là hành động thay đổi kiểu dữ liệu của một biến (hoặc biểu thức) bất kỳ sang một kiểu dữ liệu khác.

- Ép kiểu dữ liệu được sử dụng khi:

+ Đưa dữ liệu về định dạng mà mình mong muốn giúp xử lý cho thuật toán.

+ Khởi tạo hoặc gán một biến với giá trị của một kiểu dữ liệu khác.

+ Kiểu dữ liệu của biến trả về kết quả khác với kiểu dữ liệu trả về của hàm đó.

Ví dụ: $x = 15.3456$ (x là số thực) và ta muốn ép kiểu x thành một số nguyên. Sau khi ép kiểu thì $x = 15$ (bỏ phần sau dấu phẩy)

$y = 978$ (y là số nguyên) và bạn muốn ép kiểu y thành một số thực. Sau khi ép kiểu thì $y = 987.000...$ (thêm phần sau dấu phẩy)

- Các loại ép kiểu dữ liệu:

+ Ép kiểu ngầm định: Trong quá trình biên dịch, nếu phát hiện một lệnh gán giá trị khác loại cho một biến, trình biên dịch sẽ tự động chuyển kiểu.

Khi tính toán 1 biểu thức, ví dụ như các phép toán số học, các hạng tử cần phải có cùng một kiểu thì việc tính toán mới diễn ra, nếu các hạng tử không cùng kiểu, sẽ xảy ra ép kiểu ngầm định, kiểu nhỏ hơn sẽ chuyển sang kiểu lớn hơn. Có đây cấp độ như sau:

$int \rightarrow unsigned\ int \rightarrow long \rightarrow unsigned\ long \rightarrow long\ long \rightarrow unsigned\ long\ long \rightarrow float \rightarrow double \rightarrow long\ double$

Lưu ý: Khi tính toán, kiểu dữ liệu *short* và *char* sẽ tự động được chuyển sang *int*. Các ký tự thuộc kiểu dữ liệu *char* sẽ

được tự động chuyển sang số nguyên tương ứng với số ở hệ cơ số 10 trong bảng mã ASCII (xem phụ lục).

Ví dụ 1: *double d = 3; // giá trị của d là 3.0*

int n = 1.9; // giá trị của n là 1

Ví dụ 2: *int n = 10 + 16.95;*

Biểu thức bên phải là phép toán cộng giữa kiểu *int* (10) và kiểu *double* (16.95), khi đó 10 sẽ được chuyển kiểu thành 10.0 và phép toán trở thành $10.0 + 16.95$, kết quả là 26.95, được gán cho *n* là một biến thuộc kiểu *int* nên được chuyển kiểu và $n = 26$.

Ví dụ 3: *int n = 'a' + 1;*

Biểu thức bên phải là phép toán cộng giữa kiểu *char* ('a') và kiểu *int* (1), khi đó 'a' sẽ được chuyển kiểu thành số nguyên 97 và phép toán trở thành $97 + 1$, kết quả là 98, được gán cho *n* là một biến thuộc kiểu *int* nên được chuyển kiểu và $n = 98$.

Ví dụ 4: *char ch = 1 + 'a';*

Biểu thức bên phải là phép toán cộng giữa kiểu *int* (1) và kiểu *char* ('a'), khi đó 'a' sẽ được chuyển kiểu thành số nguyên 97 và phép toán trở thành $1 + 97$, kết quả là 98, được gán cho *ch* là một biến thuộc kiểu *char* nên được chuyển kiểu và $ch = 'b'$.

+ Ép kiểu tường minh: Là việc chuyển 1 dữ liệu sang kiểu dữ liệu khác theo ý muốn của lập trình viên. Kiểu biến đổi này có thể làm mất thông tin.

Cú pháp chuyển kiểu như sau:

(kiểu_dữ_liệu) dữ_liệu_cần_chuyển

Ví dụ 1: Xét hai câu lệnh sau:

int n = 10; float f = (float) n;

Giá trị của biến *f* là 10.0

Ví dụ 2: $\text{float } f = 10 / 4;$

Ở ví dụ này, ta mong muốn kết quả $f = 2.5$. Nhưng quy ước trong C/C++, 10 và 4 được hiểu là hai số nguyên và khi thực hiện phép chia giữa hai số nguyên, kết quả sẽ trả về một số nguyên, do đó giá trị biểu thức $10/4$ sẽ là 2, dẫn tới $f = 2$.

Ta có thể khắc phục bằng cách ép kiểu tường minh như sau:
 $\text{float } f = ((\text{float})10) / 4;$

Khi đó, số nguyên 10 sẽ được chuyển thành số thực 10.0. Biểu thức trở thành phép chia giữa số thực và số nguyên, cho ra kết quả sẽ là một số thực 2,5. Vậy $f = 2.5$.

1.3. Kiểu ký tự và chuỗi ký tự

Kiểu chuỗi ký tự (string) trong lập trình là một loại dữ liệu dùng để lưu trữ và xử lý chuỗi các ký tự. Chuỗi ký tự thường được sử dụng để đại diện cho văn bản, từ, cụm từ và thông tin có thứ tự của các ký tự.

1.3.1. Ký tự (char)

Trong hầu hết các ngôn ngữ lập trình, chuỗi ký tự được biểu diễn bằng một dãy các ký tự nối liền nhau. Ví dụ, chuỗi "Hello, World!" có thể được biểu diễn bằng chuỗi các ký tự 'H', 'e', 'l', 'l', 'o', ',', ' ', 'W', 'o', 'r', 'l', 'd', '!'.

Trong ngôn ngữ lập trình C++, các ký tự được biểu diễn bằng mã ASCII. Bảng mã ASCII chứa 256 ký tự bao gồm các ký tự thông thường như chữ cái, số, dấu chấm, dấu phẩy, dấu cách, dấu chấm than (!), dấu hỏi, dấu cộng, dấu trừ và nhiều ký tự khác.

Ví dụ: Ký tự số: '0', '1', '2', '3', '4'...

Ký tự chữ: 'a', 'b', 'c', 'A', 'B', 'C'...

Ký tự đặc biệt: '!', '@', '#'...

Cú pháp khai báo ký tự:

char tên_ký_tự ;

1.3.2. Chuỗi ký tự (string)

Chuỗi ký tự là một tập hợp các ký tự.

Cú pháp khai báo:

string tên_chuỗi_ký_tự ;

Ví dụ: - Khai báo chuỗi *s*: *string s*;

- Khai báo chuỗi *s* có giá trị là "ABCD":

string s = "ABCD";

- Một số đặc trưng của chuỗi ký tự:

+ Các toán tử "+", "+=" được dùng để cộng chuỗi hoặc ghép hai chuỗi.

Ví dụ: Nếu chuỗi *s1* là "abcd", chuỗi *s2* là "efgh" thì:

Kết quả của phép cộng hai chuỗi *s1* + *s2* là "abcdefgh"

Sau câu lệnh ghép hai chuỗi *s2* += *s1*, có *s2* là "efghabcd"

+ Vị trí các ký tự trong chuỗi được đánh số bắt đầu từ 0.

Để truy cập vào ký tự thứ *i* trong chuỗi *s* ta gọi *s[i]*:

s1[0] = 'a'; s2[1] = 'f'

+ Để lấy ra độ dài chuỗi *s* ta gọi lệnh: *s.size()*

- Để đổi ký tự chữ thường thành chữ hoa:

char ch2 = (char) toupper(ch1);

- Để đổi ký tự chữ hoa thành chữ thường:

Cách 1: Sử dụng hàm có sẵn trong thư viện

char ch2 = (char) tolower(ch1);

Cách 2: Sử dụng bảng mã ASCII

+ Bước 1: Đổi ký tự sang mã thập phân tương ứng trong bảng mã ASCII (xem phụ lục)

Nhận xét: Các chữ cái in hoa từ A-Z có mã thập phân từ 65 đến 90. Các chữ cái in thường từ a-z có mã thập phân từ 97 đến 122. Mã thập phân của chữ cái in hoa và chữ cái in thường tương ứng hơn kém nhau 32 đơn vị.

+ *Bước 2*: Lấy mã thập phân vừa đổi cộng thêm 32, ta được mã thập phân tương ứng của chữ cái thường.

+ *Bước 3*: Đổi mã thập phân ở bước 2 sang ký tự, ta thu được kết quả.

Ta có code đầy đủ như sau:

```
char ch2 = char(int(ch1)+32)
```

Giải thích: Câu lệnh `int(ch1)` dùng để ép kiểu một biến thuộc kiểu ký tự (`char`) thành một biến thuộc kiểu số nguyên (`int`). Kết quả trả về của phép ép kiểu sẽ là mã thập phân của ký tự đó trong bảng mã ASCII.

Lưu ý: Phép cộng `ch1` với 32 sẽ cho ra kết quả là số nguyên nên trong phép tính này, `ch1` đã được tự động chuyển thành kiểu `int`.

Ta có code rút gọn như sau:

```
char ch2 = char(ch1+32)
```

1.4. Bài tập thực hành

1. NHẬP XUẤT #1

Mã bài: NBI01

Bạn An có a quả bóng, bạn Trang có số bóng gấp năm lần bạn An, hỏi bạn Trang có bao nhiêu quả bóng?

INPUT

Nhập số nguyên a ($0 \leq a \leq 100$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
12	60

Code mẫu:

```
#include <iostream>
using namespace std;
int main(){
    int a;
    cin >> a;
    cout << a * 5;
    return 0;
}
```

2. NHẬP XUẤT #2

Mã bài: NBIO2

Bạn An có a quả bóng, bạn Trang có b quả bóng, bạn Sa có c quả bóng. Hỏi ba bạn có tổng bao nhiêu quả bóng?

INPUT

Nhập ba số nguyên a, b, c ($0 \leq a, b, c \leq 100$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
2 1 10	13

Code mẫu:

```
#include <iostream>
using namespace std;
int main(){
    int a, b, c;
    cin >> a >> b >> c;
    cout << a + b + c;
    return 0;
}
```

3. NHẬP XUẤT #3

Mã bài: NBIO3

Bạn An có a quả bóng, bạn Trang có b quả bóng, bạn Sa có số bóng bằng tích số bóng của hai bạn An và Trang. Hỏi cả ba bạn có bao nhiêu quả bóng?

INPUT

Nhập hai số nguyên a, b ($0 \leq a, b \leq 1000$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
5 10	65
1 2	5

Code mẫu:

```
#include <iostream>
using namespace std;
int main(){
    int a, b;
    cin >> a >> b;
    cout << a + b + a * b;
    return 0;
}
```

4. DIỆN TÍCH TAM GIÁC

Mã bài: NBTRI

Hôm nay ở Trường Mầm non Bách Khoa An được học về hình tam giác. Về nhà bố hỏi An câu hỏi sau: Biết độ dài 3 cạnh của một tam giác, hỏi diện tích của tam giác đó là bao nhiêu?

INPUT

Nhập ba số nguyên dương a, b, c ($1 \leq a, b, c \leq 100$) mô tả độ dài ba cạnh của một tam giác.

OUTPUT

In ra kết quả bài toán lấy đến hai chữ số sau phần thập phân.

INPUT	OUTPUT
67 37 96	911.92

Gợi ý:

- Sử dụng công thức Heron để tính diện tích tam giác khi biết 3 cạnh.
- Sử dụng câu lệnh sau để in ra hai chữ số sau phần thập phân của biến s .

`cout << fixed << setprecision(2) << s;`

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int a, b, c;
    cin >> a >> b >> c ;
    double p = (double) (a + b + c) / 2;
    double s = sqrt(p*(p - a)*(p - b)*(p - c));
    cout<< fixed << setprecision(2) << s;
    return 0;
}
```

5. HÌNH CHỮ NHẬT

Mã bài: NBRECT

Cho độ dài hai cạnh của một hình chữ nhật. Tính chu vi và diện tích của hình chữ nhật đó.

INPUT

Nhập hai số thực a, b là độ dài hai cạnh của hình chữ nhật ($0 < a, b \leq 10^4$), cách nhau bởi một dấu cách.

OUTPUT

In ra chu vi và diện tích của hình chữ nhật lấy chính xác đến hai chữ số sau phần thập phân, cách nhau bởi một dấu cách.

INPUT	OUTPUT
2.12 3.0	10.24 6.36

6. BÌNH PHƯƠNG

Mã bài: NBSQR

Cho một số nguyên n . In ra bình phương của số đó.

INPUT

Nhập một số nguyên n ($|n| \leq 10^9$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
3	9
-5	25

7. TỔNG LẬP PHƯƠNG

Mã bài: NBCUBE

An rất thích khám phá những con số, với mỗi cặp số bất kì, cậu đều biến chúng thành những phép tính đa dạng. Lần

này, An nhận được 2 số a, b nhưng cậu muốn tính tổng lập phương của 2 số đó. Hãy giúp An nhé!

INPUT

Nhập hai số thực a, b ($-100 \leq a, b \leq 100$)

OUTPUT

In ra kết quả bài toán lấy đến hai chữ số sau phần thập phân.

INPUT	OUTPUT
1.23 2.07	10.73

8. LUỸ THỪA

Mã bài: NBPOW

An rất thích khám phá những con số, nhưng có vẻ An chán các kết quả bé tí rồi nên lần này An sẽ chơi trội hơn. Hãy in ra a^b cho An nhé!

INPUT

Nhập hai số nguyên a, b ($|a^b| \leq 10^{18}$).

OUTPUT

In ra kết quả của bài toán.

INPUT	OUTPUT
2 5	32

Gợi ý: Sử dụng hàm $\text{pow}(a, b)$ để tính lũy thừa.

- Tuy nhiên hàm pow trả về kết quả là một số thực, khi số thực đó lớn, nó sẽ được in ra màn hình dưới dạng rút gọn.

Ví dụ: Số đó có giá trị là 10^{18} thì chương trình sẽ in ra màn hình giá trị $1\text{e}+18$.

- Vậy, ta phải thực hiện thêm phép ép kiểu kết quả sang số nguyên để được đáp án chính xác.

Code ví dụ: *(long long) pow(a, b)*

9. CĂN BẬC HAI

Mã bài: NBSQRT

Cho một số nguyên không âm n . Tính căn bậc hai của n .

INPUT

Nhập số n ($0 \leq n \leq 10^9$).

OUTPUT

In ra căn bậc hai của n , lấy đến 2 chữ số sau phần thập phân.

INPUT	OUTPUT
2	1.41

Gợi ý: Sử dụng hàm $\text{sqrt}(n)$ để tính căn bậc hai của số n .

10. ĐƯỜNG CHÉO

Mã bài: NBDIAG

Cho độ dài 1 cạnh của hình vuông. Tính độ dài đường chéo hình vuông đó.

INPUT

Nhập một số thực a mô tả cạnh hình vuông ($0 < a \leq 10^4$).

OUTPUT

In ra độ dài đường chéo hình vuông lấy đến hai chữ số sau phần thập phân.

INPUT	OUTPUT
5	7.07

11. PHÉP TÍNH #1

Mã bài: NBEXP1

Hôm nay An nhận được 2 số thực a, b . Cậu muốn tính tổng, hiệu, tích và thương của chúng. Hãy giúp An nhé!

INPUT

Nhập hai số thực a, b ($-100 \leq a, b \leq 100$).

OUTPUT

In ra 4 số thực là tổng, hiệu, tích, thương của a và b , mỗi số trên 1 dòng, lấy đến hai chữ số sau phần thập phân.

INPUT	OUTPUT
10.2 3	13.20 7.20 30.60 3.40

Gợi ý: Để in ra ký tự xuống dòng, ta sử dụng `endl` hoặc `"\n"`. Code ví dụ:

```
cout << fixed << setprecision(2) << a + b << endl;
```

```
cout << a + b << "\n";
```

12. PHÉP TÍNH #2

Mã bài: NBEXP2

Hôm nay An nhận được 5 số nguyên a, b, c, d, mod . Cậu muốn tính tích các số a, b, c, d sau đó chia dư cho số mod . Hãy giúp An nhé!

INPUT

Nhập năm số nguyên a, b, c, d, mod ($|a|, |b|, |c|, |d| < 100, 0 < \text{mod} \leq 100$).

OUTPUT

In ra kết quả của bài toán.

Lưu ý: phép chia dư phải ra kết quả là một số không âm.

INPUT	OUTPUT
2 8 6 1 11	8

Gợi ý:

- Trong C/C++, khi muốn tính $(a \% c)$ trong đó $a < 0$, thì chương trình sẽ trả về kết quả là một số âm.
- Để kết quả là một số không âm, ta sử dụng tính chất của phép chia dư: $a \% c = (a \% c + c) \% c$ (cộng thêm c để kết quả thành số không âm, nhưng số dư không thay đổi).

13. PHÉP TÍNH #3

Mã bài: NBEXP3

Hôm nay An nhận được 5 số nguyên a, b, c, d, mod . Cậu muốn tính $(a + b) * (c + d)$ sau đó chia dư cho số mod . Hãy giúp An nhé!

INPUT

Nhập năm số nguyên a, b, c, d, mod ($|a|, |b|, |c|, |d| < 100, 0 < \text{mod} \leq 100$).

OUTPUT

In ra kết quả của bài toán.

Lưu ý: phép chia dư phải ra kết quả là một số không âm.

INPUT	OUTPUT
2 8 6 1 11	4

14. PHÉP TÍNH #4

Mã bài: NBEXP4

An nhận được bốn số nguyên a, b, c, d .

Cậu muốn tính $a^b + c^d$. Hãy giúp An nhé!

INPUT

Nhập bốn số nguyên a, b, c, d ($|a^b + c^d| \leq 10^9$).

OUTPUT

In ra kết quả của bài toán.

INPUT	OUTPUT
2 8 6 1	262

15. TÍCH LỚN NHẤT

Mã bài: NBMAXMUL

Cho số nguyên dương n . In ra cách phân tích số n thành tổng hai số nguyên dương a, b sao cho tích của chúng là lớn nhất.

INPUT

Nhập số n ($2 \leq n \leq 10^9$).

OUTPUT

In ra hai số nguyên dương a và b thỏa mãn đề bài ($a \leq b$), hai số ngăn cách nhau bởi một dấu cách.

INPUT	OUTPUT
6	3 3
7	3 4

16. GIAO ĐIỂM HAI ĐƯỜNG THẲNG

Mã bài: NBCROSS

Cho 2 đường thẳng:

$$y = a_1x + b_1$$

$$y = a_2x + b_2$$

Hãy tìm tọa độ giao điểm của chúng.

INPUT

Nhập bốn số nguyên a_1 , b_1 , a_2 , b_2 có giá trị tuyệt đối không vượt quá 1000.

OUTPUT

In ra tọa độ giao điểm x , y của hai đường thẳng cách nhau bởi một dấu cách. Đáp án lấy chính xác đến 6 chữ số sau phần thập phân.

INPUT	OUTPUT
1 0 2 -2	2.000000 2.000000

17. CHỮ CÁI #1

Mã bài: NBCHAR1

Cho một ký tự chữ cái in hoa trong bảng chữ cái tiếng Anh, in ra chữ cái in thường tương ứng.

INPUT

Nhập một ký tự là chữ cái in hoa.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
A	a
Z	z

18.CHỮ CÁI #2

Mã bài: NBCHAR2

Cho một ký tự chữ cái in thường trong bảng chữ cái tiếng Anh, in ra chữ cái in hoa tương ứng.

INPUT

Nhập một ký tự là chữ cái thường.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
a	A
z	Z

19.CHỮ CÁI #3

Mã bài: NBCHAR3

Cho một ký tự chữ cái in hoa trong bảng chữ cái tiếng Anh, in ra chữ cái liền trước, in thường của chữ cái đó.

INPUT

Nhập một ký tự là một chữ cái in hoa.

OUTPUT

Kết quả bài toán.

Lưu ý: Coi chữ cái liền trước chữ cái 'A' là chữ cái 'Z'.

INPUT	OUTPUT
C	b
A	z

20.CHỮ CÁI #4

Mã bài: NBCHAR4

Bạn An có một cái đĩa trên đó có ghi các chữ cái tiếng Anh in thường từ 'a' tới 'z' theo thứ tự vòng tròn. Trang chọn một chữ cái rồi đổ An chữ cái đứng sau chữ cái đó n vị trí là chữ cái nào? (Chữ 'a' đứng sau chữ 'z' khi các chữ cái được xếp theo vòng tròn.)

INPUT

Nhập một ký tự chữ cái tiếng Anh in thường và một số nguyên không âm n có giá trị không vượt quá 10^9 , cách nhau bởi một dấu cách.

OUTPUT

In ra chữ cái đứng sau n vị trí so với chữ cái đã cho dưới dạng chữ in hoa.

INPUT	OUTPUT
a 1	B
y 5	D

21.CHỮ CÁI #5

Mã bài: NBCHAR5

Cho 2 ký tự chữ cái a, b in thường trong bảng chữ cái tiếng Anh. In ra số lượng các chữ cái nằm giữa a và b.

INPUT

Nhập hai chữ cái in thường, cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT	Giải thích
a e	3	Các chữ cái nằm giữa là b, c, d.
d a	2	Các chữ cái nằm giữa là b, c.

Gợi ý:

- Xét hai trường hợp a đứng trước b và a đứng sau b trong bảng chữ cái. Dùng hiệu hai mã thập phân, sau đó trừ đi 1 để tìm số lượng ký tự nằm giữa a và b. Nếu kết quả ra số âm thì có nghĩa là a nằm sau b trong bảng chữ cái và ngược lại.

- Dùng hàm $\max(a, b)$ để lấy giá trị lớn nhất trong hai số a và b.

- Code ví dụ: $\max(a - b - 1, b - a - 1)$

22.CHUỖI KÝ TỰ #1

Mã bài: NBSTR1

An được thầy giáo cho học về chuỗi ký tự. An thắc mắc rằng chuỗi ký tự khi được viết theo thứ tự ngược lại sẽ như thế nào, bạn hãy giúp An nhé.

INPUT

Nhập một chuỗi ký tự s viết liền không dấu, có độ dài không vượt quá 255.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
maerdedoc	codedream
abc	cba

Gợi ý: Sử dụng lệnh *reverse(s.begin(), s.end())* để đảo ngược tất cả các ký tự trong chuỗi s.

23. CHUỖI KÝ TỰ #2

Mã bài: NBSTR2

Cho chuỗi s, in ra chuỗi s và chuỗi đảo ngược của s (liền nhau)

INPUT

Nhập một chuỗi s gồm các chữ cái thường viết liền không dấu, có độ dài không vượt quá 255.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
abc	abccba
code	codeedoc

24. CHUỖI KÝ TỰ #3

Mã bài: NBSTR3

An có một chuỗi ký tự gồm các chữ cái trong bảng chữ cái tiếng Anh in thường. Trang hỏi An chữ cái thứ 2 và thứ 7 của chuỗi này dưới dạng in hoa là chữ cái nào? Hãy giúp An trả lời nhé!

INPUT

Nhập chuỗi ký tự chỉ chứa các chữ cái tiếng Anh in thường, có độ dài không vượt quá 255.

OUTPUT

In ra lần lượt chữ cái thứ 2 và thứ 7 của chuỗi đó dưới dạng in hoa, cách nhau bởi một dấu cách.

INPUT	OUTPUT
codedream	O E
whereyourdreamscometrue	H O

25. CHUỖI KÝ TỰ #4

Mã bài: NBSTR4

An rất thích chuỗi s. Vì vậy An muốn bạn in ra 2 lần chuỗi s đó để chia hai nửa tặng Trang.

INPUT

Nhập một chuỗi ký tự s, có độ dài không vượt quá 255.

OUTPUT

In ra hai lần chuỗi đó, cách nhau bởi dấu cách.

INPUT	OUTPUT
code	code code
dream	dream dream

26. CHUỖI KÝ TỰ #5

Mã bài: NBSTR5

Cho một chuỗi ký tự không bao gồm khoảng trống, dấu cách. In ra số lượng ký tự của chuỗi ký tự đó.

INPUT

Nhập một chuỗi ký tự có độ dài không vượt quá 255.

OUTPUT

In ra số lượng ký tự của chuỗi ký tự đó.

INPUT	OUTPUT
whereyourdreamscometrue	23

27. GHÉP SỐ

Mã bài: NBMATCH

Cho 2 số nguyên không âm a và b. Ghép 2 số a, b rồi đảo ngược lại ta được số nguyên không âm c.

INPUT

Nhập hai số nguyên không âm a, b ($0 \leq a, b \leq 10^4$).

OUTPUT

In ra số c tìm được.

INPUT	OUTPUT
12 34	4321
10 20	201

Gợi ý:

- Sử dụng hàm *string* $s = to_string(n)$ để đổi số thành chuỗi ký tự.
- Sử dụng *long* $n = stoll(s, 0, 10)$ để đổi chuỗi s thành số ở hệ cơ số 10 (giúp làm mất các số 0 ở đầu chuỗi).

28.ĐỔI GIỜ

Mã bài: NBHOUR

Cho số nguyên không âm n là số giây. Đổi n giây sang định dạng giờ:phút:giây.

INPUT

Nhập số nguyên n ($0 \leq n \leq 50000$).

OUTPUT

In ra theo định dạng hh:mm:ss trong đó hh là giờ, mm là phút, ss là giây.

INPUT	OUTPUT
3690	01:01:30

Gợi ý: Sử dụng hàm *printf* (" $\%0.2d$ ", x) để in thêm các số 0 ở đầu của số x nếu x không đủ độ dài 2 chữ số.

Ví dụ: $x = 1$ thì kết quả là "01", $x = 10$ thì kết quả là "10".

CHƯƠNG 2. CẤU TRÚC RẼ NHÁNH

2.1. Câu lệnh if, if..else

Câu lệnh điều kiện if thường được sử dụng để kiểm tra một điều kiện logic và cho phép chương trình thực hiện các hành động khác nhau dựa trên kết quả của điều kiện đó.

Cú pháp câu lệnh if:

```
if (biểu_thức_logic) {
    //khối lệnh cần thực hiện nếu biểu thức logic đúng;
}
```

- Code ví dụ:

```
if (a > b) {
    c = a;
    cout << a << " lon hơn " << b;
}
```

Cú pháp câu lệnh if...else:

```
if (mệnh_đề_logic) {
    //khối lệnh cần thực hiện nếu biểu thức logic đúng;
} else {
    //khối lệnh cần thực hiện nếu biểu thức logic sai;
}
```

- Code ví dụ:

```
if (a > b)
    c = a;
else
    c = b;
```

Lưu ý:

- Nếu chỉ có một khối lệnh cần thực hiện thì khối lệnh đó không cần đặt trong cặp ngoặc nhọn.

- Nếu có nhiều hơn một khối lệnh cần thực hiện thì các khối lệnh đó phải được đặt trong cùng cặp ngoặc nhọn với ý nghĩa: tất cả cùng xảy ra nếu mệnh đề đúng/sai.

Các điều kiện logic

- Sử dụng toán tử logic && để xét nhiều điều kiện thoả mãn đồng thời (xem bảng các phép toán ở phần Kiến thức chung)

if (a>b && a>c)

max = a;

- Sử dụng toán tử logic || để xét một trong những điều kiện thoả mãn (xem bảng các phép toán ở phần Kiến thức chung)

if (a>b || a>c) // a lớn hơn b hoặc a lớn hơn c

d = a;

Câu lệnh if có thể lồng nhau

- *Code ví dụ:*

```
if (n > 0) {  
    if (a > b)  
        c = a;  
    else  
        c = b;  
}
```

2.2. Phép toán quan hệ, phép toán logic

Để thực hiện kiểm tra điều kiện, ta sử dụng các phép toán quan hệ và phép toán logic. Dưới đây là các ví dụ về cách sử dụng các phép toán trong câu lệnh if:

Phép toán	Biểu diễn	Ví dụ
Phép so sánh nhỏ hơn	<	<i>if (a < b)</i> nếu a nhỏ hơn b
Phép so sánh nhỏ hơn hoặc bằng	<=	<i>if (a <= b)</i> nếu a nhỏ hơn hoặc bằng b

Chương 2. Cấu trúc rẽ nhánh

Phép so sánh lớn hơn	>	<i>if</i> ($a > b$) nếu a lớn hơn b
Phép so sánh lớn hơn hoặc bằng	>=	<i>if</i> ($a \geq b$) nếu a lớn hơn hoặc bằng b
Phép so sánh bằng	==	<i>if</i> ($a == b$) nếu a bằng b
Phép so sánh không bằng	!=	<i>if</i> ($a != b$) nếu a không bằng b
Phép và (and)	&&	<i>if</i> ($a != b \ \&\& \ a == c$) nếu (a không bằng b) đồng thời (a bằng c)
Phép hoặc (or)		<i>if</i> ($a != b \ \ a == c$) nếu (a không bằng b) hoặc (a bằng c)
Phép phủ định (not)	!	<i>if</i> ($!(a == b)$) nếu a không bằng b

2.3. Bài tập thực hành

1. SO SÁNH #1

Mã bài: NBCOMP1

Tìm số lớn hơn trong 2 số a , b cho trước.

INPUT

Nhập hai số nguyên a , b ($|a|, |b| \leq 10^9$), cách nhau một dấu cách.

OUTPUT

In ra “YES” nếu $a > b$, ngược lại in ra “NO”.

INPUT	OUTPUT
1 2	NO
5 1	YES
8 8	NO

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int a, b;
    cin >> a >> b;
    if(a > b)
        cout << "YES";
    else
        cout << "NO";
    return 0;
}
```

2. SO SÁNH #2

Mã bài: NBCOMP2

So sánh hai số a, b ($|a|, |b| \leq 10^9$) cho trước.

INPUT

Nhập hai số nguyên a, b cách nhau một dấu cách.

OUTPUT

In ra “YES” nếu $a = b$, ngược lại in “NO”.

INPUT	OUTPUT
1 2	NO

Gợi ý: Sử dụng toán tử “==” (hai dấu bằng liên nhau) để so sánh hai số có bằng nhau hay không.

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int a, b;
    cin >> a >> b;
    if(a == b)
        cout<< "YES";
    else
        cout<< "NO";
    return 0;
}
```

3. SO SÁNH #3

Mã bài: NBCOMP3

Tìm số bé nhất, lớn nhất trong ba số a, b, c cho trước.

INPUT

Nhập ba số nguyên a, b, c ($|a|, |b|, |c| \leq 10^9$), các số cách nhau một dấu cách.

OUTPUT

In ra lần lượt số bé nhất, lớn nhất cách nhau bởi một dấu cách.

INPUT	OUTPUT
9 6 2	2 9
1 9 5	1 9

4. SO SÁNH #4

Mã bài: NBCOMP4

Tìm số lớn nhất trong năm số nguyên cho trước.

INPUT

Nhập năm số a, b, c, d, e ($|a|, |b|, |c|, |d|, |e| \leq 10^9$), các số cách nhau bởi một dấu cách.

OUTPUT

In ra số lớn nhất trong năm số.

INPUT	OUTPUT
1 5 1 2 4	5
0 8 2 1 8	8

5. CHẴN LẺ

Mã bài: NBODEV

Cho một số nguyên không âm n . Hãy kiểm tra số n là số chẵn hay số lẻ?

INPUT

Nhập số n ($0 \leq n \leq 10^5$).

OUTPUT

In ra 1 nếu n chẵn, ngược lại in ra 0.

INPUT	OUTPUT
9	0
100	1

Gợi ý: Số chẵn là số có số dư khi chia cho 2 là 0. Sử dụng phép chia lấy dư (%).

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int n;
    cin >> n;
    if(n % 2 == 0)
        cout << 1;
    else
        cout << 0;
    return 0;
}
```

6. ĐỘ SÁNG

Mã bài: NBLIGHT

Cảm biến màn hình con mắt búc của An bị hỏng nên độ sáng nhấp lung tung. Khi ánh sáng thấp hơn 50 nó sẽ gấp đôi, tắt màn hình nếu sáng quá mức 70, còn lại nhấp nháy.

INPUT

Nhập một độ sáng màn hình x ($0 \leq x \leq 100$).

OUTPUT

In ra độ sáng màn hình, nếu nhấp nháy thì ghi "OH NO".

INPUT	OUTPUT
10	20
80	0
65	OH NO

Code mẫu:

```
#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    if (n < 50)
        cout << n * 2;
    else if (n > 70)
        cout << 0;
    else
        cout << "OH NO";
    return 0;
}
```

7. KIỂM TRA TAM GIÁC #1

Mã bài: NBTRI1

Cho ba số nguyên dương a, b, c. Kiểm tra xem ba số đó có phải là ba cạnh của một tam giác hay không?

INPUT

Nhập ba số nguyên dương a, b, c ($0 < a, b, c \leq 10^9$), các số cách nhau bởi một dấu cách.

OUTPUT

Nếu a, b, c là ba cạnh của một tam giác thì in ra “YES”, ngược lại in ra “NO”.

INPUT	OUTPUT
3 4 5	YES
3 4 10	NO

Gợi ý: Trong một tam giác, tổng hai cạnh luôn lớn hơn cạnh còn lại.

8. KIỂM TRA TAM GIÁC #2

Mã bài: NBTRI2

Hôm nay An được ôn tập về tam giác. Bạn hãy giúp An kiểm tra xem tam giác là vuông, nhọn hay tù nhé!

INPUT

Nhập ba số nguyên dương a, b, c được ngăn cách nhau bởi dấu cách, có giá trị không quá 10^9 , mô tả ba cạnh của một tam giác.

OUTPUT

In ra 0 nếu ba số đó mô tả ba cạnh của một tam giác vuông, in ra 1 nếu ba số đó mô tả ba cạnh của một tam giác nhọn, in ra 2 nếu ba số đó mô tả ba cạnh của một tam giác tù.

INPUT	OUTPUT
3 4 5	0
10 10 10	1
2 3 4	2

Gợi ý: Giả sử độ dài ba cạnh của tam giác là a, b, c trong đó c là cạnh lớn nhất trong 3 cạnh:

- Nếu $a^2 + b^2 = c^2$ thì a, b, c tạo thành tam giác vuông.
- Nếu $a^2 + b^2 < c^2$ thì a, b, c tạo thành tam giác tù.
- Nếu $a^2 + b^2 > c^2$ thì a, b, c tạo thành tam giác nhọn.

9. CHIA HẾT 15

Mã bài: NBMOD15

Kiểm tra tích của hai số a, b có chia hết cho 15 không.

INPUT

Nhập hai số nguyên a, b ($0 \leq a, b \leq 100$), cách nhau bởi một dấu cách.

OUTPUT

In ra 1 nếu tích a và b chia hết cho 15, in ra 0 trong trường hợp ngược lại.

INPUT	OUTPUT
5 3	1
19 2	0

10.XẾP LOẠI

Mã bài: NBCLASSIFY

An vừa kết thúc kỳ thi tốt nghiệp và biết được điểm thi 3 môn toán, lý, hóa. Bạn hãy kiểm tra xem An tốt nghiệp loại gì nhé! Biết:

- Nếu tổng 3 môn lớn hơn hoặc bằng 12 thì bạn An đỗ tốt nghiệp.
- Nếu đỗ tốt nghiệp mà điểm tất cả các môn đều lớn hơn 4 thì An xếp loại Giỏi, ngược lại thì xếp loại Trung bình.

INPUT

Nhập ba số nguyên a, b, c ($0 \leq a, b, c \leq 10$), các số cách nhau một dấu cách.

OUTPUT

In ra “Gioi” nếu An xếp loại Giỏi, “Trung binh” nếu xếp loại Trung bình, “Truot” nếu An không đỗ tốt nghiệp.

INPUT	OUTPUT
6 7 8	Gioi
1 1 1	Truot
4 5 3	Trung binh

11. TÍNH TOÁN

Mã bài: NBCAL5

Cho 2 số thực a, b và 1 ký tự thể hiện phép tính cộng, trừ, nhân, chia (“+”, “-”, “*”, “/”) để điền vào giữa 2 số a và b. Hãy tính kết quả của phép tính đó.

INPUT

Nhập hai số thực a, b ($|a|, |b| \leq 1000$) và phép tính c, cách nhau bởi một dấu cách.

OUTPUT

In ra kết quả phép tính lấy đến hai chữ số sau phần thập phân.

INPUT	OUTPUT
1.2 3.4 +	4.60

12. CẤP SỐ

Mã bài: NBARI1

Cho ba số nguyên dương a, b, c. Hãy kiểm tra xem ba số đó tạo thành một cấp số cộng hay là một cấp số nhân.

INPUT

Nhập ba số nguyên a, b, c ($1 \leq a, b, c \leq 10^9$), các số cách nhau một dấu cách.

OUTPUT

Nếu ba số tạo thành cấp số cộng thì in ra “cap so cong”, nếu tạo thành cấp số nhân thì in ra “cap so nhan”.

Biết dữ liệu đầu vào luôn đảm bảo có kết quả duy nhất.

INPUT	OUTPUT
2 4 6	cap so cong
2 4 8	cap so nhan

Gợi ý:

- Dãy số cấp số cộng là dãy số thoả mãn số đứng sau luôn bằng tổng số đứng trước với một số không đổi.
- Dãy số cấp số nhân là dãy số thoả mãn số đứng sau luôn bằng tích số đứng trước với một số không đổi (số khác 0).

13.CẤP SỐ TIẾP THEO

Mã bài: NBARI2

Cho 3 số nguyên dương a, b, c tạo thành một cấp số cộng hoặc cấp số nhân. Hãy tìm số tiếp theo của ba số đó trong dãy số.

INPUT

Nhập ba số nguyên dương a, b, c ($1 \leq a, b, c \leq 10^9$), các số cách nhau một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
2 4 6	8
2 4 8	16

14.SỐ THỨ K

Mã bài: NBFINDK

Cô Ngát có n số nguyên từ 1 tới n . Cô sắp xếp dãy số này theo một cách khá đặc biệt. Cô xếp tất cả các số lẻ theo thứ tự từ bé đến lớn trước, hết các số lẻ thì cô mới xếp đến các số chẵn, cũng theo thứ tự từ bé đến lớn. Bé Khánh thấy dãy số này và tự hỏi số thứ k của dãy sẽ là số nào. Hãy giúp bé nhé!

INPUT

Nhập hai số nguyên n, k ($1 \leq k \leq n \leq 10^{12}$), cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT	Giải thích
5 2	3	Dãy số đó là: 1 3 5 2 4 Số thứ 2 là 3.
5 4	2	Dãy số đó là: 1 3 5 2 4 Số thứ 4 là 2.

15. ĐƯỜNG VỀ NHÀ

Mã bài: NBHOME

Hãy tưởng tượng rằng bạn đang đi trên vỉa hè (vỉa hè ở Mỹ nhé, vỉa hè này thì xin khỏi nói rồi). Nhưng đang đi thì tự dưng bạn mỏi chân, không muốn đi nữa, trong khi phải có n bước nữa mới về tới nhà. Rất may bạn là một diễn viên giỏi, nên có thể nhờ người đi đường đưa về. Mỗi lần đi nhờ, bạn sẽ đi từ vị trí x , bạn có thể được đi nhờ tới vị trí $x + 1$, $x + 2$, $x + 3$, $x + 4$ hoặc $x + 5$.

Tuy bạn có thể xin người khác cho đi nhờ, nhưng do vẫn còn tí tự trọng, nên bạn mong muốn đi nhờ ít lần nhất để về tới nhà.

Hãy tính toán số lần nhỏ nhất phải đi nhờ để về tới nhà.

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10^6$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT	Giải thích
20	4	Sử dụng 4 lần đi nhờ, mỗi lần di chuyển 5 bước.
23	5	Sử dụng 5 lần đi nhờ, 4 lần di chuyển 5 bước, 1 lần di chuyển 3 bước.

16. BỘ NĂM SỐ #1

Mã bài: NBBNS1

Cho năm số nguyên dương a, b, c, d, e . Hãy kiểm tra xem năm số này có đôi một khác nhau hay không.

INPUT

Nhập lần lượt năm số nguyên dương a, b, c, d, e ($1 \leq a, b, c, d, e \leq 1000$) được ngăn cách nhau bởi một dấu cách.

OUTPUT

In ra “YES” nếu cả 5 số đều khác nhau, “NO” nếu ngược lại.

INPUT	OUTPUT
9 3 8 2 7	YES
9 7 3 9 2	NO

17.BỘ NĂM SỐ #2

Mã bài: NBBNS2

Cho năm số nguyên dương a, b, c, d, e . Hãy kiểm tra xem trong năm số này có tồn tại đúng ba số giống nhau hay không.

INPUT

Nhập lần lượt năm số nguyên dương a, b, c, d, e ($1 \leq a, b, c, d, e \leq 1000$) được ngăn cách nhau bởi một dấu cách.

OUTPUT

In ra “YES” nếu có đúng ba số giống nhau trong năm số, in ra “NO” nếu ngược lại

INPUT	OUTPUT
1 2 1 3 1	YES
1 1 1 1 2	NO

18. BỘ NĂM SỐ #3

Mã bài: NBBNS3

Cho năm số nguyên dương a, b, c, d, e . Hãy tìm ra số xuất hiện nhiều lần nhất trong năm số này.

INPUT

Nhập lần lượt 5 số nguyên dương a, b, c, d, e ($1 \leq a, b, c, d, e \leq 1000$) được ngăn cách nhau bởi một dấu cách.

OUTPUT

In ra số xuất hiện nhiều nhất trong 5 số. Nếu có nhiều hơn một số xuất hiện nhiều nhất, in ra số có giá trị lớn nhất.

INPUT	OUTPUT
1 2 2 1 1	1
1 2 5 2 1	2

19. BỘ NĂM SỐ #4

Mã bài: NBBNS4

Cho năm số nguyên dương a, b, c, d, e . Hãy kiểm tra xem trong năm số này có tồn tại đúng bốn số giống nhau hay không.

INPUT

Nhập lần lượt năm số nguyên dương a, b, c, d, e ($1 \leq a, b, c, d, e \leq 1000$) được ngăn cách nhau bởi một dấu cách.

OUTPUT

In ra “YES” nếu có đúng bốn số giống nhau, “NO” nếu ngược lại.

INPUT	OUTPUT
1 2 3 4 5	NO
2 5 2 2 2	YES

20. NGÀY SINH

Mã bài: NBD0B

Hôm nay gia đình của An lại có tranh luận lạ lẫm... An cãi nhau um tùm với người yêu vì muốn được gọi là anh trong khi đó bạn kia lại quả quyết nếu thế thì phải gọi là chị. Bạn hãy giúp An phân giải vụ này xem ai lớn hơn để cứu vớt hạnh phúc trên bờ vực tan vỡ này nhé.

INPUT

Dữ liệu đầu vào gồm hai dòng, mỗi dòng nhập ba số nguyên a, b, c ($1 \leq a \leq 31, 1 \leq b \leq 12, 1 \leq c \leq 10^9$) lần lượt mô tả ngày tháng năm sinh của An và người yêu.

OUTPUT

In ra 1 nếu An lớn tuổi hơn, ngược lại in ra 2.

INPUT	OUTPUT
18 1 2004 11 11 2004	1
1 1 2004 1 1 2004	2

21.GHÉP HÌNH

Mã bài: NBGH

Cho kích thước hai hình chữ nhật, hãy kiểm tra xem hai hình đó có thể ghép thành một hình chữ nhật khác không ? (Hai hình phải tiếp xúc nhau, không được đè lên nhau).

INPUT

Nhập bốn số nguyên dương a_1, b_1, a_2, b_2 có giá trị không vượt quá 10^9 , được ghi cách nhau bởi một dấu cách. Trong đó a_1, b_1 là kích thước của miếng bìa thứ nhất, a_2, b_2 là kích thước của miếng bìa thứ hai.

OUTPUT

In ra “1” nếu có thể ghép hai miếng bìa thành một hình chữ nhật, in ra “0” nếu ngược lại.

INPUT	OUTPUT
1 1 3 1	1
1 5 1 9	1
1 5 2 3	0

22.SẮP XẾP

Mã bài: NBSORT

An mới được học sắp xếp nên vẫn còn bỡ ngỡ. Bạn hãy giúp An sắp xếp lại 3 số theo thứ tự tăng dần nhé.

INPUT

Nhập ba số nguyên a, b, c ($1 \leq a, b, c \leq 10^9$) được cách nhau bởi một dấu cách.

OUTPUT

In ra ba số a, b, c sau khi đã sắp xếp tăng dần, cách nhau bởi một dấu cách.

INPUT	OUTPUT
5 3 4	3 4 5
2 1 6	1 2 6

23. NHỎ NHÌ

Mã bài: NBSECOND

Cho 4 số nguyên dương a, b, c, d. Tìm số nhỏ thứ nhì.

INPUT

Nhập bốn số nguyên dương a, b, c, d đôi một khác nhau ($1 \leq a, b, c, d \leq 10^9$), các số cách nhau bởi một dấu cách.

OUTPUT

In ra số nhỏ thứ nhì.

INPUT	OUTPUT
1 2 3 4	2
1 2 3 5	2

24. TỔNG HAI SỐ

Mã bài: NBSUM2N

Cho một số nguyên dương n. In ra số cách khác nhau phân tích số n thành tổng hai số nguyên không âm.

INPUT

Nhập một số n ($1 \leq n \leq 10^9$).

OUTPUT

In ra số cách phân tích số n thành tổng hai số nguyên không âm.

Lưu ý: $a + b$ và $b + a$ chỉ tính là một cách.

INPUT	OUTPUT
8	5

25. TỔNG BA SỐ

Mã bài: NBSUM3N

Cho một số nguyên dương n . Kiểm tra xem số n có thể phân tích thành tổng ba số nguyên dương liên tiếp hay không.

INPUT

Nhập một số n ($1 \leq n \leq 10^9$).

OUTPUT

In ra “YES” nếu có thể phân tích n thành ba số nguyên dương, ngược lại in ra “NO”.

INPUT	OUTPUT
60	YES
35	NO

Gợi ý: n có thể phân tích thành tổng ba số nguyên dương liên tiếp khi n có dạng: $n = a + (a + 1) + (a + 2)$ với $a > 0$.

Suy ra, $n = 3a + 3 = 3.(a + 1)$

Vì n là số nguyên nên n phải chia hết cho 3, vì $a > 0$ nên $n > 3$

26. TỔNG LIÊN TIẾP #1

Mã bài: NBSUMCONT1

Cho số nguyên dương x . Kiểm tra x có phải tổng các số tự nhiên liên tiếp bắt đầu từ 1 không.

INPUT

Nhập số nguyên dương x ($1 \leq x \leq 10^6$).

OUTPUT

In ra “YES” nếu x là tổng của các số tự nhiên bắt đầu từ 1, ngược lại in ra “NO”.

INPUT	OUTPUT
3	YES
4	NO

27. SỐ CHÍNH PHƯƠNG #1

Mã bài: NBSQRNUM1

Cho một số nguyên không âm n . Kiểm tra xem n có phải là số chính phương không?

INPUT

Nhập số nguyên không âm n ($0 \leq n \leq 10^9$).

OUTPUT

In ra “YES” nếu n là số chính phương, ngược lại in ra “NO”.

INPUT	OUTPUT
4	YES
5	NO

28.SỐ CHÍNH PHƯƠNG #2

Mã bài: NBSQRNUM2

Cho 1 số nguyên không âm n . In ra số chính phương nhỏ nhất lớn hơn hoặc bằng n .

INPUT

Nhập số nguyên không âm n ($0 \leq n \leq 10^9$).

OUTPUT

In ra số chính phương nhỏ nhất lớn hơn hoặc bằng n .

INPUT	OUTPUT
10	16

29.NĂM NHUẬN

Mã bài: NBLEAP

Cho một số nguyên dương mô tả một năm bất kì, hãy xác định xem năm đó có phải năm nhuận hay không.

INPUT

Nhập số nguyên dương có giá trị không vượt quá 10^5 .

OUTPUT

In ra 1 nếu đó là năm nhuận, ngược lại in ra 0.

INPUT	OUTPUT
1999	0
2000	1
1800	0

30. BA ĐIỂM THẲNG HÀNG

Mã bài: NBLINE

Cho tọa độ ba điểm, xác định xem ba điểm đó có phải là ba điểm thẳng hàng không.

INPUT

Nhập sáu số nguyên $x_1, y_1, x_2, y_2, x_3, y_3$ có giá trị tuyệt đối không vượt quá 10^4 lần lượt mô tả tọa độ của ba điểm, các số cách nhau một dấu cách.

OUTPUT

In ra 1 nếu ba điểm đó thẳng hàng, ngược lại in ra 0.

INPUT	OUTPUT
1 1 2 2 3 3	1
1 1 2 2 0 1	0

31. VỊ TRÍ TƯƠNG ĐỐI #1

Mã bài: NBRPOS1

Cho tọa độ bốn đỉnh của một tứ giác lồi cùng với tọa độ của một điểm bất kì. Kiểm tra xem điểm đó có nằm ngoài tứ giác đã cho hay không.

INPUT

- Bốn dòng đầu tiên, mỗi dòng nhập hai số thực mô tả tọa độ một đỉnh của tứ giác, cách nhau một dấu cách.
- Dòng cuối cùng nhập tọa độ của điểm cần kiểm tra, cách nhau một dấu cách.

Các tọa độ là số thực có giá trị tuyệt đối không vượt quá 10^5 . Các đỉnh được liệt kê theo chiều kim đồng hồ.

OUTPUT

In ra 1 nếu điểm đó nằm ngoài tứ giác đã cho, ngược lại in ra 0.

INPUT	OUTPUT
0 0 0 2 2 2 2 0 3 3	1
0 0 0 2 1 2 2 0 1.5 1	0

32. VỊ TRÍ TƯƠNG ĐỐI #2

Mã bài: NBRPOS2

Cho tọa độ 4 điểm A, B, C, D. Kiểm tra xem điểm D có nằm trong hoặc trên tam giác ABC hay không?

INPUT

- Ba dòng đầu tiên, mỗi dòng lần lượt nhập tọa độ của 3 đỉnh A, B, C, các số cách nhau một dấu cách.
- Dòng cuối cùng nhập tọa độ của điểm D, cách nhau một dấu cách.

Các tọa độ là số thực có giá trị tuyệt đối không vượt quá 10^6 . Các đỉnh tam giác được liệt kê theo chiều kim đồng hồ.

OUTPUT

In ra “YES” nếu điểm đó nằm trong hoặc trên cạnh tam giác, ngược lại in ra “NO”.

INPUT	OUTPUT
0 0 0 5 5 0 2.5 3	NO
0 0 0.5 2 2 0 0.5 1.5	YES

33.VỊ TRÍ TƯƠNG ĐỐI #3

Mã bài: NBRPOS3

Cho đường tròn tâm O bán kính R và điểm M. Kiểm tra xem điểm M có nằm trong hay trên đường tròn hay không.

INPUT

Nhập năm số thực x_O, y_O, x_M, y_M, R ($|x, y| \leq 10^6, 1 \leq R \leq 10^6$) lần lượt là toạ độ của tâm O, điểm M, bán kính R, cách số cách nhau một dấu cách.

OUTPUT

In ra “YES” nếu M nằm trong/trên đường tròn tâm O, ngược lại in ra “NO”.

INPUT	OUTPUT
0 0 2.8 3 4	NO

CHƯƠNG 3. CẤU TRÚC LẶP

3.1. Vòng lặp for

Vòng lặp for là một cấu trúc kiểm soát lặp lại (loop) cho phép thực hiện một chuỗi các hành động với số lần cụ thể. Câu lệnh for thường được sử dụng khi biết trước số lần lặp lại mà ta muốn thực hiện.

Cú pháp:

```
for (câu_lệnh_khởi_tạo; điều_kiện_lặp; câu_lệnh_cập_nhật) {  
    // khối_lệnh_lặp;  
}
```

Giải thích:

- Bước 1: *câu_lệnh_khởi_tạo* được thực hiện (chỉ thực hiện đúng một lần trước khi vòng lặp được diễn ra). Thường dùng để khởi tạo biến điều khiển lặp.

- Bước 2: *điều_kiện_lặp* sẽ được kiểm tra.

+ Nếu kết quả bằng *True* thì *khối_lệnh_lặp* sẽ được thực hiện, tiếp theo là *câu_lệnh_cập_nhật* (thường dùng để cập nhật giá trị của biến điều khiển lặp) và cuối cùng là chương trình sẽ lặp lại Bước 2.

+ Nếu kết quả bằng *False* thì kết thúc vòng lặp.

Ví dụ 1: Tính tổng các số nguyên từ 1 đến n.

```
int sum = 0;  
for (int i = 1; i <= n; i++){  
    sum += i;  
}
```

Ví dụ 2: Tính tổng các số nguyên lẻ từ 1 đến n.

```
int sum = 0;
for (int i = 1; i <= n; i++){
    if (i % 2 == 1)
        sum += i;
}
```

3.2. Vòng lặp while, do...while

3.2.1. Vòng lặp while

Vòng lặp while là một cấu trúc kiểm soát lặp lại (loop) cho phép thực hiện một chuỗi các hành động trong khi một điều kiện cụ thể vẫn còn đúng. Vòng lặp while thường được sử dụng khi không biết trước số lần lặp lại cụ thể.

Cú pháp:

```
while (điều_kiện_lặp) {
    // các_khối_lệnh_lặp;
}
```

Giải thích: *điều_kiện_lặp* được kiểm tra trước mỗi vòng lặp. Nếu *điều_kiện_lặp* trả về True thì *các_khối_lệnh_lặp* sẽ được thực hiện và vòng lặp tiếp tục thực hiện. Ngược lại câu lệnh while sẽ kết thúc.

Ví dụ: Tính tổng các số nguyên từ 1 đến n.

```
int i = 1, sum = 0;
while (i <= n){
    sum += i;
    i++;
}
```

3.2.2. Vòng lặp do...while

Vòng lặp do...while là một cấu trúc kiểm soát lặp lại (loop) tương tự như vòng lặp while, nhưng nó đảm bảo ít nhất một lần thực hiện các hành động bên trong vòng lặp, ngay cả khi điều kiện không đúng. Các khối lệnh sẽ được thực hiện trước khi điều kiện lặp được kiểm tra đúng hay sai.

Cú pháp:

```
do {  
    // các_khối_lệnh_lặp;  
} while (điều_kiện_lặp);
```

Giải thích: Đầu tiên, các khối_lệnh_lặp sẽ được thực hiện. Sau đó, chương trình sẽ kiểm tra điều_kiện_lặp, nếu kết quả trả về là *True* thì vòng lặp tiếp tục được thực hiện, ngược lại câu lệnh while sẽ kết thúc.

Ví dụ: Tính tổng các số nguyên chẵn từ 1 đến n.

```
int i = 1, sum = 0;  
do {  
    if (i % 2 == 0)  
        sum += i;  
    i++;  
} while (i <= n);
```

3.3. Bài tập thực hành

1. TỔNG ĐOẠN SỐ

Mã bài: NBSUMN

Tính tổng các số tự nhiên nhỏ hơn hoặc bằng n.

INPUT

Nhập một số tự nhiên n có giá trị không vượt quá 10^5 .

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
3	6
5	15

2. TỔNG CHẴN

Mã bài: NBSUMEVEN2

Tính tổng các số tự nhiên chẵn nhỏ hơn hoặc bằng n .

INPUT

Nhập một số tự nhiên n có giá trị không vượt quá 10^5 .

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
5	6
10	30

3. TÍCH ĐOẠN SỐ

Mã bài: NBMULMN

Cho hai số nguyên dương m, n . Hãy tính tích các số trong khoảng từ m đến n .

INPUT

Nhập hai số tự nhiên m, n ($0 < m \leq n \leq 15$), cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
3 5	60
1 3	6

4. TÍCH LẤY DƯ

Mã bài: NBMULMOD

Cho ba số nguyên dương a , b và c . Tìm số dư của phép chia tích các số nguyên trong khoảng từ a đến b cho c .

INPUT

Nhập ba số nguyên dương a , b và c ($a, b \leq 10^5$, $c \leq 10^9$), các số cách nhau một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
1 4 11	2

Gợi ý: Dùng vòng lặp for để tính tích các số trong khoảng từ a đến b . Vì tích này có thể rất lớn nên ta thực hiện vừa tính tích vừa chia lấy dư cho số c .

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int a, b, c;
    cin >> a >> b >> c;
    long long tích = 1;
    for (int i = a; i <= b; i++)
        tích = (tích * i) % c;
    cout << tích;
    return 0;
}
```

5. TÍCH LẺ

Mã bài: NBMULODD

Cho hai số nguyên dương m, n . Hãy tính tích các số lẻ trong khoảng từ m đến n .

INPUT

Nhập hai số nguyên dương m, n ($0 < m < n \leq 15$), cách nhau một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
3 5	15
3 8	105

6. GIAI THỪA #1

Mã bài: NBFACITOR1

Cho số nguyên không âm n . Hãy tính $n!$ (n giai thừa).

INPUT

Nhập số nguyên không âm n ($n \leq 10$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
3	6
4	24

7. GIAI THỪA #2

Mã bài: NBFACITOR2

Cho số nguyên dương n , mod ($n, \text{mod} \leq 1000$)

Tính $S = (1! + 2! + 3! + \dots + n!) \% \text{mod}$.

INPUT

Nhập hai số nguyên dương n , mod.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
3 11	9

Gợi ý: Dùng hai vòng lặp for lồng nhau để tính tổng các giai thừa. Vòng lặp đầu tiên để xác định giai thừa cần tính, vòng lặp thứ hai sẽ tính giai thừa đó. Vì các giai thừa và tổng các giai thừa này có thể rất lớn nên ta thực hiện vừa tính vừa chia lấy dư cho số mod.

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int n, mod;
    cin >> n >> mod;
    int sum = 0;
    for (int i = 1; i <= n; i++){
        int fac = 1;
        for (int j = 1; j <= i; j++)
            fac = (fac * j) % mod;
        sum = (sum + fac) % mod;
    }
    cout << sum;
    return 0;
}
```

8. TỔNG F

Mã bài: NBCALF

Cho số nguyên dương n .

Tính $F(n) = 1 * 3 + 2 * 4 + \dots + (n - 2) * n$.

INPUT

Nhập số nguyên dương n ($3 \leq n \leq 10^4$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
3	3
4	11

9. TỔNG C

Mã bài: NBCALC

Cho số nguyên dương n và một dãy a gồm n số nguyên $a_1, a_2, a_3, \dots, a_n$. Tính trọng số C của dãy a biết C được tính theo công thức như sau:

$$C = 2 * a_1 + 4 * a_2 + 6 * a_3 + \dots + 2 * n * a_n$$

INPUT

- Dòng 1: Nhập số nguyên dương n ($1 \leq n \leq 1000$).
- Dòng 2: Nhập n số nguyên có giá trị tuyệt đối không vượt quá 10^6 mô tả dãy a , các số cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
4 1 2 3 4	60

10.TỔ HỢP

Mã bài: NBCOMB

Cho hai số nguyên không âm k, n.

Tính $C(k, n) = \frac{n!}{k! * (n-k)!}$

INPUT

Nhập hai số nguyên không âm k, n ($0 \leq k \leq n \leq 15$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
1 2	2
3 6	20

11.CHIA HẾT #1

Mã bài: NBDIVISOR1

Cho hai số nguyên dương m, n. Hãy tìm các số chia hết cho m trong khoảng từ 1 đến n.

INPUT

Nhập hai số nguyên dương m, n ($0 < m \leq n \leq 10^6$) cách nhau một dấu cách.

OUTPUT

In ra các số tìm được theo thứ tự từ nhỏ đến lớn, cách nhau bởi một dấu cách.

INPUT	OUTPUT
2 10	2 4 6 8 10

12.CHỌN CHỖ

Mã bài: NBHALL

Ở Hội trường Trường THPT Nguyễn Huệ đang có hoạt động ngoại khóa dành cho học sinh lớp chuyên tin. An với Sa muốn ngồi cùng hàng ghế với nhau để nói chuyện cho dễ nhưng các hàng ghế lại rất đông, An muốn biết có bao nhiêu hàng ghế còn đủ chỗ trống để cho cả hai bạn ngồi (không nhất thiết phải ngồi cạnh nhau). Hãy giúp An nhé!

INPUT

- Dòng đầu tiên nhập số n là số hàng ghế ở hội trường ($1 \leq n \leq 100$)
- n dòng sau, mỗi dòng nhập hai số nguyên không âm a, b ($0 \leq a \leq b \leq 100$) cách nhau một dấu cách, lần lượt là số ghế đã có người ngồi và tổng số ghế ở hàng đó.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
3 1 5 0 4 6 6	2

13.HOÁN VỊ

Mã bài: NBPERM

Mẹ bé Khánh xếp ra được một được một dãy ký tự. Ngay lúc mẹ không để ý, bé Khánh đã nghịch làm hỏng mất dãy. Thật không may, mẹ bé lại không nhớ rõ dãy ký tự mà mình đã xếp, nên mẹ đã quyết định thử từng hoán vị khác nhau các ký tự có thể. Hãy giúp mẹ bé tính số lượng dãy ký tự phải thử nhé!

INPUT

Nhập một chuỗi gồm các ký tự đôi một phân biệt (độ dài của chuỗi không vượt quá 20).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
code	24

Gợi ý: Số lượng hoán vị của dãy n phần tử là $n!$

14.KÝ TỰ LẶP LẠI

Mã bài: NBREPCH

Cho một ký tự, hãy in ra ký tự đó 10 lần liên tiếp trên cùng một dòng.

INPUT

Nhập một ký tự duy nhất.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
b	bbbbbbbbbb

15. CHUỖI LẶP LẠI

Mã bài: NBREPSTR

Cho một số nguyên dương n và một chuỗi ký tự (không bao gồm dấu cách). In ra chuỗi ký tự đó n lần.

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 100$) và chuỗi ký tự trên cùng một dòng, ngăn cách nhau bởi một dấu cách.

OUTPUT

In ra chuỗi ký tự đó n lần, mỗi lần trên một dòng.

INPUT	OUTPUT
3 abc	abc abc abc

16. CHUỖI KÝ TỰ #6

Mã bài: NBCOUNTCH

Cho một dãy ký tự, hãy đếm xem trong dãy ký tự đó có bao nhiêu ký tự 'a', bao nhiêu ký tự 'b', bao nhiêu ký tự 'c' và bao nhiêu ký tự còn lại.

INPUT

Nhập một chuỗi ký tự có độ dài không vượt quá 1000.

OUTPUT

In ra số lượng ký tự 'a', ký tự 'b', ký tự 'c' và ký tự còn lại, mỗi số trên một dòng.

INPUT	OUTPUT
whereyourdreamscometrue	1 0 1 21

17.IN CHUỖI

Mã bài: NBPRSTR

Cho một chuỗi. In ra các tiền tố của chuỗi đó theo mẫu.

INPUT

Nhập một chuỗi ký tự có độ dài không vượt quá 255.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
abcd	a ab abc abcd abc ab a

Gợi ý: Sử dụng hai vòng lặp for lồng nhau để in ra các tiền tố của chuỗi ký tự.

18. ƯỚC SỐ

Mã bài: NBDIV

Cho một số nguyên dương n . In ra số cách khác nhau để phân tích số n thành tích hai số nguyên dương khác nhau.

INPUT

Nhập một số nguyên dương n ($1 \leq n \leq 10^9$).

OUTPUT

In ra số cách phân tích số n thành tích 2 số khác nhau.

Lưu ý: $a * b$ giống $b * a$.

INPUT	OUTPUT
6	2

Gợi ý: Số cách phân tích chính là số lượng ước số của n nằm trong khoảng từ 1 đến \sqrt{n} (trong trường hợp n là số chính phương thì không tính ước \sqrt{n}).

19. CHỮ SỐ TẬN CÙNG

Mã bài: NBLASTDIGIT

Cho số n và 1 dãy số gồm n số nguyên. Tìm chữ số tận cùng của tích các số trong dãy.

INPUT

- Dòng 1: chứa số nguyên dương n ($1 \leq n \leq 1000$).
- Dòng 2: chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^6 cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT	Giải thích
4 1 2 3 -4	4	Tích các số trong dãy là -24. Số tận cùng là 4

20.IN DÃY #1

Mã bài: NBPRARR1

Cho số nguyên dương n và dãy số gồm n số nguyên, in ra số đầu tiên, số cuối cùng của dãy số đó.

INPUT

- Dòng 1: chứa số nguyên dương n ($1 \leq n \leq 1000$).
- n dòng sau, mỗi dòng chứa một số nguyên có giá trị tuyệt đối không vượt quá 10^6 .

OUTPUT

In ra số đầu tiên, số cuối cùng của dãy số đó trên hai dòng khác nhau.

Lưu ý: Không dùng dữ liệu kiểu mảng để giải quyết bài toán.

INPUT	OUTPUT
5 1 4 5 6 9	1 9

21.IN DÃY #2

Mã bài: NBPRARR2

Cho số nguyên dương n , x và dãy số gồm n số nguyên.

Hãy in ra dãy số đó đến khi gặp số chẵn thứ x .

INPUT

- Dòng 1: Hai số n, x ($1 \leq x \leq n \leq 10^5$) cách nhau bởi một dấu cách.

- Dòng 2: Nhập n số nguyên có giá trị tuyệt đối không vượt quá 10^9 cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
5 2 1 2 3 4 5	1 2 3

Lưu ý: Nếu dãy không đủ x số chẵn thì in ra toàn bộ dãy.

22.IN DÃY #3

Mã bài: NBPRARR3

Cho số nguyên dương n . Tìm dãy số có n phần tử sao cho số thứ i là tổng các số từ 1 đến i .

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10000$).

OUTPUT

INPUT	OUTPUT	Giải thích
3	1 3 6	Dòng 1 chứa tổng: 1 Dòng 2 chứa tổng: 1 + 2 Dòng 3 chứa tổng: 1 + 2 + 3

In ra dãy số đó, mỗi số nằm trên một dòng.

23. TỔNG LIÊN TIẾP #2

Mã bài: NBSUMCONT2

Cho số nguyên dương n . Phân tích n thành tổng các số nguyên dương liên tiếp.

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10^6$).

OUTPUT

In ra dãy các số nguyên dương liên tiếp có tổng bằng n , theo thứ tự từ bé đến lớn cách nhau bởi 1 dấu cách. Nếu có nhiều cách phân tích, in ra cách phân tích có số đầu tiên của dãy là nhỏ nhất.

INPUT	OUTPUT
4	4
10	1 2 3 4

24. SỐ CHÍNH PHƯƠNG #3

Mã bài: NBSQRNUM3

Cho 1 số nguyên không âm n . Đếm xem có bao nhiêu số chính phương nhỏ hơn hoặc bằng n .

INPUT

Nhập số nguyên không âm n ($0 \leq n \leq 10^6$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT	Giải thích
10	4	Các số chính phương nhỏ hơn hoặc bằng 10 là: 0, 1, 4, 9.
16	5	Các số chính phương nhỏ hơn hoặc bằng 16 là: 0, 1, 4, 9, 16.

25.SỐ CHÍNH PHƯƠNG #4

Mã bài: NBSQRNUM4

Cho số n và dãy số có n số nguyên dương. Đếm xem trong dãy số đó có bao nhiêu số chính phương.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên dương có giá trị không vượt quá 10^6 , cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
5 1 9 3 5 4	3

26. PHÂN TÍCH SỐ #1

Mã bài: NBNUM1

Cho số nguyên dương n . Đếm số lượng chữ số của n .

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10^{18}$).

OUTPUT

Số lượng chữ số của số n .

INPUT	OUTPUT
123	3

Gợi ý: Loại bỏ dần các chữ số của số n và đếm số lần lặp.

- Bỏ đi chữ số cuối của số n bằng phép chia nguyên cho 10 ($n = n / 10$).
- Lặp lại cho đến khi $n = 0$.
- Số lần lặp chính là số lượng số chữ số của n .

27. PHÂN TÍCH SỐ #2

Mã bài: NBNUM2

Cho số nguyên dương n . Tìm tổng các chữ số của n .

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10^{18}$).

OUTPUT

Tổng các chữ số của n .

INPUT	OUTPUT	Giải thích
123	6	$1 + 2 + 3 = 6$

Gợi ý:

- Cách 1: Đổi kiểu số sang dạng chuỗi, sau đó xét từng ký tự của chuỗi: đổi ký tự đó sang số nguyên (dùng bảng mã ASCII) và cộng vào tổng.

- Cách 2: Tương tự bài trước, tách từng chữ số của số n bằng cách:

+ Dùng phép chia lấy dư cho 10 ($n \% 10$) để lấy ra chữ số cuối cùng của số n , cộng vào tổng ($\text{sum} += n \% 10$).

+ Bỏ đi chữ số cuối của số n bằng phép chia nguyên cho 10 ($n = n / 10$)

+ Lặp lại cho đến khi $n = 0$.

28. PHÂN TÍCH SỐ #3

Mã bài: NBNUM3

Cho số nguyên dương n . Hãy tính trung bình cộng các chữ số của n .

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10^{18}$).

OUTPUT

In ra trung bình cộng các chữ số của n , kết quả làm tròn đến chữ số thập phân thứ 2

INPUT	OUTPUT
346	4.33
123	2.00

29. TỔNG ƯỚC

Mã bài: NBSUMDIV

Cho số nguyên dương n . Tính tổng các ước của số n .

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10^6$).

OUTPUT

In ra tổng các ước của số n .

INPUT	OUTPUT
6	12
7	8

30.SỐ ĐẶC BIỆT #1**Mã bài: NBSPEC1**

Số đặc biệt là số chia hết cho tổng các chữ số của nó.

Cho số nguyên dương n . Kiểm tra xem n có phải là số đặc biệt không.

INPUT

Nhập số nguyên dương n ($0 < n \leq 10^{18}$).

OUTPUT

In ra “YES” nếu n là số đặc biệt, ngược lại in ra “NO”.

INPUT	OUTPUT
12	YES
13	NO

31. TỔNG NHỎ NHẤT

Mã bài: NBMINSUM

Cho số nguyên dương n . Tìm cách phân tích số n thành tích hai số nguyên dương a, b sao cho tổng của chúng là nhỏ nhất.

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10^9$).

OUTPUT

In ra 2 số nguyên dương a và b thỏa mãn đề bài ($a \leq b$), cách nhau bởi một dấu cách.

INPUT	OUTPUT
6	2 3
8	2 4

32. ƯỚC THỨ K

Mã bài: NBDIVK

Cho 2 số nguyên dương n và k . Tìm ước số thứ k của n .

INPUT

Nhập hai số nguyên dương n và k ($1 \leq n \leq 10^{12}$, $1 \leq k \leq 10^9$) cách nhau 1 dấu cách.

OUTPUT

In ra ước số thứ k của n . Nếu số lượng ước của n nhỏ hơn k thì in ra -1.

INPUT	OUTPUT
6 2	2

33.KIỂM TRA NGUYÊN TỐ

Mã bài: NBPRIME1

Số nguyên tố là số chỉ có hai ước là 1 và chính nó.

An muốn biết một số nguyên n có phải số nguyên tố không, bạn hãy giúp An nhé.

INPUT

Nhập một số nguyên n ($2 \leq n \leq 10^9$).

OUTPUT

Nếu số n là số nguyên tố in ra “YES”, ngược lại in ra ước nguyên tố dương nhỏ nhất của n .

INPUT	OUTPUT
5	YES
6	2

Gợi ý:

- Cách đơn giản: Tạo một vòng lặp để kiểm tra trong đoạn từ 2 đến $n - 1$ xem có số nào là ước của n hay không.

+ Nếu có ít nhất một số là ước của n thì n không phải số nguyên tố, đồng thời ước đầu tiên gặp được trong vòng lặp sẽ là ước nguyên tố nhỏ nhất của n .

+ Nếu không có số nào là ước của n thì n là số nguyên tố.

- Cách cải tiến (cách này sẽ đạt được điểm tối đa cho bài tập này):

+ Ta thấy, nếu một số có một ước là a trong đoạn từ 2 đến \sqrt{n} thì sẽ có một ước tương ứng là $b = n/a$ trong đoạn từ \sqrt{n} đến n .

+ Để làm rõ hơn, giả sử a là một ước của n , vậy ta có thể phân tích $n = a * (n/a)$. Coi $a \leq n/a$ thì ta có $a \leq \sqrt{n}$.

+ Ví dụ: 12 có 4 ước trong đoạn $[2, 11]$ là 2, 3, 4, 6.

Phân tích $12 = 2 * 6 = 3 * 4$ ($a = 3$ hoặc 4).

Vậy để kiểm tra xem n có phải là số nguyên tố hay không, ta chỉ cần kiểm tra từ 2 đến \sqrt{n} xem có số nào là ước của n hay không thay vì kiểm tra từ 2 đến $n - 1$.

34. ĐẾM SỐ NGUYÊN TỐ

Mã bài: NBPRIME3

Cho một số nguyên dương n . Đếm xem có bao nhiêu số nguyên tố nhỏ hơn hoặc bằng n .

INPUT

Nhập số nguyên dương n ($n \leq 10^4$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
9	4
20	8

35. SỐ NGUYÊN TỐ LỚN NHẤT

Mã bài: NBPRIME3

Cho 1 số nguyên dương n . In ra số nguyên tố lớn nhất nhỏ hơn hoặc bằng n .

INPUT

Nhập số n ($2 \leq n \leq 10^5$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
6	5

36.THỪA SỐ NGUYÊN TỐ

Mã bài: NBPRIME4

Cho số nguyên dương n . Phân tích số n thành tích các thừa số nguyên tố.

INPUT

Nhập số n ($2 \leq n \leq 10^9$).

OUTPUT

In ra các số là thừa số nguyên tố sau khi phân tích của n . Các số in ra cách nhau 1 dấu cách và sắp xếp theo thứ tự tăng dần.

INPUT	OUTPUT
24	2 2 2 3
30	2 3 5

37.ƯỚC CHUNG LỚN NHẤT

Mã bài: NBGCD

Cho 2 số nguyên dương a, b . Kiểm tra xem ước chung lớn nhất của chúng có phải số nguyên tố hay không.

INPUT

Nhập hai số nguyên dương a, b ($1 \leq a, b \leq 1000$).

OUTPUT

In ra “YES” nếu thỏa mãn, ngược lại in ra “NO”.

INPUT	OUTPUT
3 12	YES
4 12	NO

Gợi ý: sử dụng hàm $\text{gcd}(a, b)$ để tìm ước chung lớn nhất của hai số a, b .

38. BỘI CHUNG NHỎ NHẤT

Mã bài: NBLCM

Cho hai số nguyên dương a, b . Tìm bội chung nhỏ nhất của a và b .

INPUT

Nhập hai số nguyên dương a, b ($1 \leq a, b \leq 1000$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
3 4	12

Gợi ý: $\text{BCNN}(a, b) = (a * b) / \text{UCLN}(a, b)$.

39. TỐI GIẢN

Mã bài: NBSIMPLIFY

Cho 2 số nguyên a, b . In ra tử số và mẫu số của phân số a/b sau khi tối giản.

INPUT

Nhập hai số nguyên a, b ($-10^6 \leq a, b \leq 10^6, b \neq 0$).

OUTPUT

In ra tử số và mẫu số của phân số a/b sau khi tối giản (cách nhau 1 dấu cách).

Lưu ý: Nếu phân số kết quả nhỏ hơn 0 thì in ra có tử số là số âm, mẫu số là số dương.

INPUT	OUTPUT
60 40	3 2
60 -40	-3 2
0 10	0 1

40.ĐỔI HỆ CƠ SỐ #1

Mã bài: NBBINARY

Cho số nguyên dương n ở hệ thập phân. Hãy biểu diễn số n ở hệ nhị phân.

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10^9$).

OUTPUT

In ra số n ở hệ nhị phân.

INPUT	OUTPUT
10	1010

41.ĐỔI HỆ CƠ SỐ #2

Mã bài: NBDECIMAL

Cho số n ở hệ nhị phân. Hãy biểu diễn số n ở hệ thập phân

INPUT

Nhập số được biểu diễn ở hệ nhị phân có độ dài không quá 30.

OUTPUT

In ra số đó ở hệ thập phân.

INPUT	OUTPUT
101	5

42. TÌM SỐ**Mã bài: NBFIND1**

Cho số nguyên dương n nguyên tố cùng nhau với 10. Tìm số nguyên dương nhỏ nhất gồm toàn số 1 và chia hết cho n .

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 100$).

OUTPUT

INPUT	OUTPUT	Giải thích
37	3	Ta có 111 là số nhỏ nhất gồm toàn số 1 và chia hết cho 37. Số 111 có 3 chữ số nên kết quả là 3.

In ra số lượng số 1 của số thỏa mãn gồm toàn số 1 và chia hết cho n .

43. VẼ TAM GIÁC VUÔNG CÂN**Mã bài: NBDEQTRI**

Trang muốn vẽ tặng An một tam giác vuông cân gồm toàn ký tự '*' với độ dài cạnh n nhân dịp sinh nhật, hãy giúp Trang nhé.

INPUT

Nhập một số nguyên dương n . ($1 \leq n \leq 100$).

OUTPUT

Tam giác vuông cân gồm toàn các ký tự '*' theo mẫu.

INPUT	OUTPUT
6	***** ***** **** *** ** *

44. VẼ THÁP TAM GIÁC

Mã bài: NBDTRI

Đây là tháp tam giác bậc 1 ($n = 1$):

```
*  
***  
*
```

Đây là tháp tam giác bậc 2 ($n = 2$):

```
*  
***  
*****  
***  
*
```

Cho một số nguyên dương n , hãy vẽ tháp tam giác bậc n .

INPUT

Nhập số n ($1 \leq n \leq 10$).

OUTPUT

Tháp tam giác bậc n.

INPUT	OUTPUT
3	<pre> * *** ***** ***** ***** *** * </pre>

45. VẼ BẢNG #1

Mã bài: NBDTABLE1

Cho số nguyên dương n. In ra bảng có n dòng, mỗi dòng gồm n số nguyên có giá trị bằng n, cách nhau bởi một dấu cách.

INPUT

Nhập số nguyên dương n ($n \leq 1000$).

OUTPUT

In ra bảng theo yêu cầu đề bài.

INPUT	OUTPUT
4	<pre> 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 </pre>

46. VẼ BẢNG #2

Mã bài: NBDTABLE2

Cho hai số nguyên dương m, n . In ra bảng gồm m hàng, mỗi hàng chứa n ký tự '*' viết liền nhau.

INPUT

Nhập hai số nguyên dương m, n ($m, n \leq 100$).

OUTPUT

In ra bảng theo yêu cầu đề bài.

INPUT	OUTPUT
5 2	** ** ** ** **

47. VẼ BẢNG #3

Mã bài: NBDTABLE3

Cho hai số nguyên dương m, n và một ký tự ch . In ra bảng gồm m hàng, mỗi hàng chứa n ký tự ch cho trước viết liền nhau.

INPUT

Nhập hai số nguyên dương m, n ($m, n \leq 100$) và ký tự ch , cách nhau bởi một dấu cách.

OUTPUT

In ra bảng theo yêu cầu đề bài.

INPUT	OUTPUT
3 3 a	aaa aaa aaa

48.ĐƯỜNG GẤP KHÚC

Mã bài: NBLENGTH

Cho tọa độ n điểm $A_1, A_2, A_3, A_4 \dots A_n$. Tính độ dài đoạn gấp khúc $A_1A_2A_3A_4\dots A_n$.

INPUT

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- n dòng tiếp theo, mỗi dòng chứa 2 số nguyên có giá trị tuyệt đối không vượt quá 10^6 thể hiện tọa độ mỗi điểm.

OUTPUT

Kết quả bài toán lấy chính xác 6 chữ số sau phần thập phân.

INPUT	OUTPUT
2 0 0 0 2	2.000000

49.CHU VI ĐA GIÁC

Mã bài: NBPERI

Cho tọa độ n đỉnh của đa giác lồi theo chiều kim đồng hồ, tính chu vi đa giác đó.

INPUT

- Dòng đầu chứa số nguyên dương n ($1 \leq n \leq 10^5$).

- n dòng tiếp theo, mỗi dòng chứa 2 số thực có giá trị không vượt quá 10^6 thể hiện tọa độ mỗi đỉnh của đa giác đó.

OUTPUT

Kết quả bài toán lấy chính xác 6 chữ số sau phần thập phân.

INPUT	OUTPUT
4	5.000000
0 0	
0 1.5	
1 1.5	
1 0	

CHƯƠNG 4. XỬ LÝ CHUỖI KÝ TỰ

4.1. Một số hàm xử lý chuỗi thông dụng

Lưu ý: Các ký tự của chuỗi đánh số bắt đầu từ 0.

4.1.1. Hàm lấy độ dài chuỗi

Cú pháp:

length(), size()

Ví dụ: Có chuỗi *s* là "abcdefgh".

- Lệnh *s.length()* trả về kết quả độ dài chuỗi *s* là 8.
- Lệnh *s.size()* cũng trả về kết quả độ dài chuỗi *s* là 8.

Code minh họa:

```
string s = "abcdefgh";  
cout << s.size() << endl; // 8  
cout << s.length() << endl; // 8
```

4.1.2. Hàm lấy chuỗi con

Cú pháp:

substr(position, n)

Trong đó:

- *position*: Vị trí bắt đầu lấy chuỗi con.
- *n*: Số lượng ký tự muốn lấy.

Ví dụ: Có chuỗi *s* là "abcdefghi".

- Lệnh *s.substr(3,2)* trả về kết quả chuỗi con là "de".

Code minh họa:

```
string s = "abcdefghab";  
cout << s.substr(3,2) << endl; // de
```

4.1.3. Hàm chèn chuỗi

Chèn chuỗi *string* vào chuỗi *s*.

Cú pháp:

insert(position, string)

insert (position, n, char)

Trong đó:

- *position*: Vị trí cần chèn.
- *string*: Chuỗi cần chèn.
- *n*: Số lần ký tự cần chèn.
- *char*: Ký tự cần chèn.

Ví dụ: Có chuỗi *s* là “*abcdefghi*”

- Chèn chuỗi “*abcd*” vào chuỗi *s* tại vị trí 2 ta gọi lệnh *s.insert(2, “abcd”)*.

Ta được được chuỗi *s* là “*ababcdcdefghi*”

- Chèn 4 ký tự *a* vào chuỗi *s* tại vị trí 3 ta gọi lệnh *s.insert(3,4, 'a')*

Ta được chuỗi *s* là “*abcaaaadefghi*”.

Code minh họa:

```
string s = "abcdefghi";
s.insert(2, "abcd");
cout << s << endl; // ababcdcdefghi
s = "abcdefghi";
s.insert(3,4, 'a');
cout << s << endl; // abcaaaadefghi
```

4.1.4. Hàm tìm chuỗi

Tìm vị trí xuất hiện chuỗi *string*

Cú pháp:

find (string)

find (start_position, string)

Trong đó:

- *string*: Chuỗi cần tìm.
- *start_position*: Vị trí bắt đầu tìm kiếm.

Ví dụ: Có chuỗi *s* là “*abcdabpq*”

- Để tìm vị trí đầu tiên xuất hiện chuỗi “ab” ta gọi lệnh `s.find(“ab”)`, lệnh trả về kết quả là 0.
- Để tìm vị trí đầu tiên xuất hiện chuỗi “ab” kể từ sau vị trí 3 ta gọi lệnh `s.find(“ab”, 3)`, lệnh trả về kết quả là 4.

Lưu ý: Nếu không tìm thấy chuỗi cần tìm, kết quả trả về của hàm là `s.end()` hay -1.

Code minh họa:

```
string s = "abcdabpq";  
cout << s.find("ab") << endl; // 0  
cout << s.find("ab", 3) << endl; // 4
```

4.1.5. Hàm xoá chuỗi

Xoá một chuỗi con trong chuỗi s

Cú pháp:

erase (position, n)

Trong đó:

- *position*: Vị trí bắt đầu xoá.
- *n*: Số ký tự cần xoá.

Ví dụ: Có chuỗi s là “abcdefgh”

- Để xoá 3 ký tự kể từ vị trí 2 trong chuỗi s ta gọi lệnh `s.erase(2,3)`. Ta được chuỗi s là “abfgh” (ab**e**defgh)

Code minh họa:

```
string s = "abcdefgh";  
s.erase(2,3);  
cout << s << endl; // abfgh
```

4.2. Bài tập thực hành

1. ĐỘ DÀI CHUỖI

Mã bài: NBSTRLEN

Cho chuỗi s bao gồm các chữ cái và dấu cách. Đếm số ký tự trong chuỗi s.

INPUT

Nhập chuỗi *s* có độ dài không vượt quá 1000.

OUTPUT

In ra số lượng ký tự trong chuỗi *s*.

INPUT	OUTPUT
code dream	10

Gợi ý:

- Sử dụng lệnh `getline(cin, s)` thay vì `cin >> s` để đọc dữ liệu theo dòng (bao gồm cả dấu cách).
- Sử dụng hàm `s.size()` để lấy ra độ dài chuỗi *s*.

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    string s;
    getline(cin, s);
    cout << s.size();
    return 0;
}
```

2. ĐẾM SỐ TỪ

Mã bài: NBCNTWORD

Cho chuỗi *s*. Đếm số từ trong chuỗi *s*, biết mỗi từ được ngăn cách nhau bởi một hoặc nhiều dấu cách. Chuỗi *s* có thể có nhiều dấu cách thừa ở đầu và cuối.

INPUT

Nhập chuỗi *s* có độ dài không vượt quá 1000.

OUTPUT

In ra số lượng từ trong chuỗi *s*.

INPUT	OUTPUT
Where your dreams come true	5
Code dream	2

Gợi ý: Dấu hiệu nhận biết một từ mới xuất hiện: theo sau ký tự dấu cách là ký tự không phải dấu cách. Để chắc chắn chương trình sẽ đếm được từ đầu tiên ta thêm một dấu cách vào đầu chuỗi.

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    string s;
    getline(cin, s);
    s = " " + s;
    int count = 0;
    for (int i = 0; i < s.size() - 1; i++){
        if (s[i] == ' ' && s[i+1] != ' '){
            count ++;
        }
    }
    cout << count;
    return 0;
}
```

3. TỪ THỨ X

Mã bài: NBWORDX

Cho một chuỗi ký tự và một số x. Hãy in ra từ thứ x của chuỗi đó. Biết mỗi từ được ngăn cách nhau bởi một hoặc nhiều dấu cách. Chuỗi s có thể có nhiều dấu cách thừa ở đầu và cuối.

INPUT

- Dòng đầu chứa một chuỗi ký tự có độ dài không quá 255.
- Dòng thứ hai chứa số nguyên dương x (x không vượt quá số từ trong chuỗi đã cho).

OUTPUT

In ra từ tìm được.

INPUT	OUTPUT
codedream is the best 4	best

4. HỌ & TÊN

Mã bài: NBNNAME

Trong tiếng Anh, tên được đặt theo công thức Tên + Đệm + Họ, nhưng trong tiếng Việt thì ngược lại Họ + Đệm + Tên. Hãy viết chương trình để chuyển từ họ tên tiếng Anh thành họ tên tiếng Việt.

INPUT

Nhập một chuỗi mô tả tên trong tiếng Anh gồm 3 từ Tên + Đệm + Họ có độ dài không vượt quá 1000, mỗi từ cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
An Thanh Nguyen	Nguyen Thanh An
Son Hoang Lai	Lai Hoang Son

Gợi ý:

- Dùng tính chất của lệnh *cin*: đọc đến khi gặp khoảng trống thì dừng.
- Đọc vào 3 chuỗi tương ứng với Tên, Đệm, Họ và in lại theo thứ tự đề bài yêu cầu.

5. TẦN SUẤT CỦA KÝ TỰ

Mã bài: NBCNTCHR

Cho chuỗi ký tự *s* và một số ký tự. Đếm số lần xuất hiện của các ký tự đó trong chuỗi *s*.

INPUT

- Dòng đầu chứa chuỗi *s* có độ dài không vượt quá 1000.
- Dòng thứ hai chứa số nguyên *m* ($0 < m \leq 100$) là số lượng ký tự cần kiểm tra.
- *m* dòng sau, mỗi dòng nhập 1 ký tự cần kiểm tra.

OUTPUT

In ra *m* dòng, dòng thứ *i* chứa số lần xuất hiện của ký tự thứ *i* cần kiểm tra.

INPUT	OUTPUT
Code Dream Where your dreams come	1
true	1
4	7
C	0
c	
e	
f	

6. Bảng chữ cái tiếng Anh

Mã bài: NBALPHABET

Cho một chuỗi ký tự. Kiểm tra xem chuỗi đó có đủ 26 chữ cái trong bảng chữ cái tiếng Anh không (không phân biệt chữ hoa, chữ thường).

INPUT

Nhập chuỗi s có độ dài không vượt quá 1000, có thể có cả chữ hoa và chữ thường.

OUTPUT

In ra “YES” nếu chuỗi s có đủ 26 chữ cái, ngược lại in ra “NO”.

INPUT	OUTPUT
The quick brown fox jumps over a lazy dog	YES
Code Dream Where your dreams come true	NO

Gợi ý:

- Biến đổi chuỗi s về toàn bộ chữ cái in thường.
- Xét các chữ cái từ ‘a’ đến ‘z’, nếu tất cả 26 chữ cái đều xuất hiện trong chuỗi s thì kết quả trả về là “YES”, ngược lại kết quả trả về là “NO”.

7. PALINDROME

Mã bài: NBSTRPALIN

Biết chuỗi đối xứng là chuỗi đọc ngược hay đọc xuôi đều giống nhau. Cho chuỗi s, kiểm tra xem một chuỗi s có phải là chuỗi đối xứng hay không.

INPUT

Nhập chuỗi s có độ dài không vượt quá 1000.

OUTPUT

In ra “YES” nếu chuỗi s là chuỗi đối xứng, ngược lại in ra “NO”.

INPUT	OUTPUT
hah	YES
abc	NO

Gợi ý:

- Cách 1: Tạo một chuỗi s_1 là đảo ngược của chuỗi s (dùng lệnh reverse), so sánh s_1 xem có bằng s hay không.
- Cách 2: Kiểm tra ký tự thứ 0 có bằng ký tự $s.size()-1$ hay không, ký tự thứ 1 có bằng ký tự $s.size()-2$ hay không . . .

Vậy cần một vòng lặp for từ vị trí 0 tới vị trí $s.size()/2$, kiểm tra xem ký tự tại tất cả các vị trí i đó có bằng ký tự tại vị trí $s.size()-i-1$ hay không.

8. XOÁ CHỮ

Mã bài: NBSTRERASE

Cho một chuỗi s và hai vị trí i, j. In ra chuỗi s sau khi xoá các ký tự từ i đến j.

INPUT

- Dòng 1: Chứa chuỗi s có độ dài không vượt quá 1000.
- Dòng 2: Chứa lần lượt hai vị trí i và j nằm trong chuỗi s, cách nhau một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
codethedream 5 7	codedream

9. CAPITALIZATION

Mã bài: NBCAP

Cho một chuỗi có cả chữ in hoa và chữ in thường. Hãy biến đổi chuỗi đó sao cho tất cả các chữ cái đều viết hoa.

INPUT

Nhập chuỗi s có độ dài không vượt quá 1000.

OUTPUT

- Dòng đầu tiên là số ký tự đã được biến đổi
- Dòng tiếp theo là chuỗi ký tự sau khi biến đổi.

INPUT	OUTPUT
Hi Hi	2 HI HI

Gợi ý: Để kiểm tra một chữ cái có phải chữ thường hay không, ta so sánh chữ cái đó với ký tự 'a' và ký tự 'z'.

Code ví dụ: *if (s[i] >= 'a' && s[i] <= 'z')*

10. ĐẢO NGƯỢC

Mã bài: NBUPLOW

Cho chuỗi s, in ra chuỗi s sau khi thay đổi các ký tự từ vị trí i đến vị trí j theo quy tắc: thay đổi chữ in hoa thành chữ in thường và chữ in thường thành chữ in hoa.

INPUT

- Dòng 1: Chứa chuỗi s có độ dài không vượt quá 1000.
- Dòng 2: Chứa lần lượt hai vị trí i và j thuộc chuỗi s.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
CodeDream 3 6	CoDEdReam

11. MẬT KHẨU

Mã bài: NBPASS

Một mật khẩu mạnh là một chuỗi ký tự có cả chữ in thường, chữ in hoa, chữ số và có ít nhất 8 ký tự.

Một mật khẩu không hợp lệ là một chuỗi ký tự có dấu cách, hoặc có ít hơn 5 ký tự.

Hãy kiểm tra chuỗi cho sẵn là loại mật khẩu nào.

INPUT

Nhập chuỗi s có độ dài không vượt quá 1000 chỉ bao gồm chữ in hoa, chữ in thường, chữ số và dấu cách.

OUTPUT

In ra “Invalid” nếu chuỗi đó là mật khẩu không hợp lệ, “Weak” nếu là mật khẩu yếu và “Strong” nếu là mật khẩu mạnh.

INPUT	OUTPUT
dung	Invalid
Dung12345	Strong
dung123	Weak

12.SẮP XẾP CHỮ SỐ

Mã bài: NBMINMAX

Cho một số nguyên dương n . Hãy tìm cách sắp xếp lại các chữ số của số n để tạo ra số nhỏ nhất và lớn nhất có thể (không được phép đặt chữ số 0 ở đầu).

INPUT

Nhập số n có số lượng chữ số không vượt quá 50.

OUTPUT

Số nhỏ nhất và lớn nhất tìm được, mỗi số trên 1 dòng.

INPUT	OUTPUT
1801	1018 8110
1000	1000 1000

Gợi ý: Chuyển số n sang dạng string hoặc nhập số n với kiểu dữ liệu string. Sử dụng hàm `sort(s.begin(), s.end())` để sắp xếp lại các ký tự trong chuỗi s .

13. TÌM CHUỖI

Mã bài: NBFINDSTR

Bé Khánh đang học các từ khoá trong ngôn ngữ lập trình, cậu đã viết các từ khoá và mã số tương ứng với từ khoá đó lên những mảnh giấy. Anh An thấy vậy liền kiểm tra em mình bằng cách: chọn ra một mảnh giấy bất kỳ và đọc mã số xuất hiện trên mảnh giấy, sau đó yêu cầu bé Khánh trả lời từ khoá tương ứng với mã số đã cho. Do mới học nên Khánh không thể nhớ hết được các từ khoá, bạn hãy giúp bé Khánh trả lời những câu hỏi của anh An nhé!

INPUT

- Dòng đầu tiên chứa hai số nguyên dương n, m lần lượt là số mảnh giấy mà Khánh có và số lần anh An muốn hỏi bé Khánh ($0 < n, m \leq 100$)
- n dòng tiếp, mỗi dòng chứa một số nguyên và một chuỗi ký tự, lần lượt là mã số và từ khoá, mô tả mảnh giấy. Mã số là một số nguyên không âm có giá trị không vượt quá 1000, từ khoá có độ dài không vượt quá 255 và không có dấu cách.
- m dòng tiếp theo, mỗi dòng nhập một số nguyên không âm có giá trị không vượt quá 1000 mô tả mã số anh An muốn kiểm tra bé Khánh.

OUTPUT

In ra m dòng, dòng thứ i là từ khoá tương ứng với câu hỏi thứ i , nếu không tồn tại từ khoá nào tương ứng với mã số đã cho thì in ra "Not found".

INPUT	OUTPUT
4 5	Not found
20 continue	continue
3 return	break

4 while	while
54 break	Not found
1	
20	
54	
4	
7	

14. CHUẨN HOÁ CHUỖI KÝ TỰ

Mã bài: NBSTRFIX

An đang phải viết lại một câu văn. Nhưng câu văn này lạ lắm, mỗi từ cách nhau nhiều hơn một dấu cách, mà các chữ cái viết hoa, viết thường không theo quy tắc nào cả. An mới nhìn thôi đã không muốn làm, bạn hãy giúp An xoá hết những dấu cách thừa giữa các từ, đầu câu, cuối câu, chuyển các ký tự chữ hoa thành chữ thường và chỉ viết hoa chữ cái đầu của câu văn nhé.

INPUT

Nhập câu văn gốc, bao gồm các chữ cái in thường, in hoa và dấu cách có độ dài không vượt quá 1000.

OUTPUT

Đưa ra kết quả mà đề bài yêu cầu.

INPUT	OUTPUT
coDE DREAMS	Code dreams

Gợi ý:

- Xoá dấu cách ở đầu: Tạo một vòng while, kiểm tra xem nếu s[0] là dấu cách thì xoá ký tự 0 của chuỗi s cho đến khi hết dấu cách ở đầu.

- Xoá dấu cách ở cuối làm tương tự xoá dấu cách ở đầu.
- Xoá dấu cách thừa ở giữa: Tạo một vòng while, kiểm tra xem trong chuỗi s có tồn tại vị trí nào mà xuất hiện hai dấu cách liền nhau hay không (dùng hàm find), nếu tồn tại thì xoá một dấu cách đi. Xoá cho đến khi không xuất hiện hai dấu cách liền nhau nữa thì dừng.

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    string s;
    getline(cin, s);
    // xoá dấu cách ở đầu
    while (s[0] == ' ') s.erase(0,1);
    // xoá dấu cách ở cuối
    while (s[s.size() - 1] == ' ')
        s.erase(s.size() - 1,1);
    // xoá dấu cách thừa ở giữa
    while(true){
        int pos = s.find(" ");
        if (pos == -1) break;
        s.erase(pos, 1);
    }
    // chuyển cả chuỗi thành chữ thường
    for (int i = 0; i < s.size(); i++)
        s[i] = tolower(s[i]);
    // chuyển chữ đầu viết hoa
    s[0] = toupper(s[0]);
    cout << s;
    return 0;
}
```

15.IN CHUỖI

Mã bài: NBPRWORD

Cho một chuỗi gồm các chữ cái và dấu cách. In ra các từ của chuỗi đó, mỗi từ trên một dòng khác nhau. Biết mỗi từ được ngăn cách nhau bởi một hoặc nhiều dấu cách. Chuỗi đó có thể có nhiều dấu cách thừa ở đầu và cuối.

INPUT

Nhập chuỗi có độ dài không vượt quá 1000.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
Where your dreams come true	Where your dreams come true

Gợi ý:

- Cách 1: Sử dụng phép chuẩn hoá chuỗi như bài 14 để các từ được cách nhau bởi đúng một dấu cách. Sau đó, dùng các hàm find và substr để tìm vị trí dấu cách tiếp theo rồi lấy từng từ trong chuỗi.

- Cách 2:

+ Dùng tính chất của lệnh cin: đọc đến khi gặp khoảng trống thì dừng. Mỗi lệnh cin chỉ đọc được một từ. Lệnh cin sẽ trả về giá trị True/False tương ứng với có đọc được dữ liệu hay không,

+ Kết hợp với while, ta có lệnh *while (cin >> s)* với ý nghĩa là đọc cho đến khi không đọc được nữa thì dừng. Vậy

với mỗi từ đọc được từ input, ta in luôn từ đó trên một dòng trước khi đọc từ tiếp theo.

+ Code mẫu: `while (cin >> s) cout << s << endl;`

+ Lưu ý: Cách này không test được trên terminal vì chương trình sẽ luôn chờ người dùng nhập dữ liệu, không biết bao giờ sẽ hết. Các bạn muốn test cần tìm hiểu về đọc dữ liệu từ file hoặc sử dụng một số IDE cho phép nhập input sẵn.

16. MÃ HOÁ #1

Mã bài: NBCODING1

Khi tìm hiểu về mã hoá chuỗi, An có ý tưởng thay vì viết câu văn, bạn ấy sẽ viết câu văn đã được mã hoá ra giấy để gửi thư cho T.

Để mã hoá một chuỗi, ta thực hiện thay thế các ký tự trong chuỗi gốc theo một quy tắc rất đơn giản: $a \rightarrow 1$, $b \rightarrow 2$, ... $z \rightarrow 26$, dấu cách $\rightarrow 27$.

Nhưng T sau khi nhận được thư đọc xong không hiểu gì, bạn hãy giúp T giải mã câu văn nhé!

INPUT

- Dòng thứ nhất chứa số nguyên dương n là độ dài câu văn ($0 < n \leq 225$).
- Dòng thứ hai chứa n số nguyên dương có giá trị không vượt quá 27 mô tả câu văn, mỗi số cách nhau bởi một dấu cách.

OUTPUT

Câu văn sau khi giải mã.

INPUT	OUTPUT
10 3 15 4 5 27 4 18 5 1 13	code dream

17.MÃ HOÁ #2

Mã bài: NBCODING2

Sau khi được các lập trình viên giúp đỡ, việc giải mã thư của An là quá dễ đối với T. Lần này để tăng độ khó, An sử dụng mật mã Atbash để mã hoá câu văn. Mật mã này có quy tắc khá đặc biệt, đó là ĐẢO NGƯỢC lại bảng chữ cái:

$a \rightarrow z, b \rightarrow y, c \rightarrow x, d \rightarrow w, e \rightarrow v, \dots$

$z \rightarrow a, y \rightarrow b, x \rightarrow c, w \rightarrow d, v \rightarrow e \dots$

Bạn hãy giúp T giải mã câu văn nhé.

INPUT

Nhập câu văn gốc có độ dài không vượt quá 1000, chỉ bao gồm các chữ cái in thường và dấu cách.

OUTPUT

Câu văn sau khi giải mã.

INPUT	OUTPUT
xlwv wivzn	code dream

18.MÃ HOÁ #3

Mã bài: NBCODING3

Trong mật mã học, mật mã Caesar là một trong những mật mã đơn giản và được biết đến nhiều nhất. Hệ mã Caesar là

một hệ mã hóa thay thế đơn âm, làm việc trên bảng chữ cái tiếng Anh 26 ký tự.

Để mã hoá một chuỗi, ta thực hiện thay thế các ký tự trong chuỗi gốc bằng các ký tự đứng sau nó 3 vị trí trong bảng chữ cái: $a \rightarrow x$, $b \rightarrow y$, $c \rightarrow z$, $d \rightarrow a$, ..., $z \rightarrow w$. Dấu cách thì giữ nguyên.

Lần này, An sử dụng mật mã Caesar để mã hoá câu văn. Bạn hãy giúp T giải mã câu văn nhé!

INPUT

Nhập câu văn đã mã hoá có độ dài không vượt quá 1000 chỉ bao gồm chữ in thường và dấu cách.

OUTPUT

Câu văn sau khi giải mã.

INPUT	OUTPUT
frgh guhdp	code dream

19. MÃ HOÁ #4

Mã bài: NBCODING4

Vì các lần trước T được gợi ý loại mật mã được sử dụng và nhờ các bạn chuyên Tin giúp đỡ nên đã giải mã được hết thư của An. Lần này, An không nói cho T biết đó là loại mật mã gì, mà chỉ cho T một câu văn mẫu (chỉ gồm các chữ cái tiếng anh in thường và dấu cách) đã mã hoá kèm theo giải mã của câu đó để T tự rút ra quy luật.

Câu văn mẫu đã mã hoá được mô tả bởi một dãy số:

21 9 20 19 27 20 6 19 23 11

Câu văn sau khi giải mã: code dream

Với câu văn mẫu trên, bạn hãy rút ra quy luật mã hoá và giúp T giải mã nốt lần này nhé!

INPUT

- Dòng thứ nhất chứa số nguyên dương n là độ dài câu văn ($0 < n \leq 225$).
- Dòng thứ hai chứa n số nguyên dương có giá trị không vượt quá 27 mô tả câu văn, cách nhau bởi một dấu cách.

OUTPUT

Câu văn sau khi giải mã.

INPUT	OUTPUT
10 21 9 20 19 27 20 6 19 23 11	code dream

Gợi ý: Quy tắc mã hoá:

- 27 \rightarrow space
- 23 \rightarrow a, 22 \rightarrow b, 21 \rightarrow c, 20 \rightarrow d, 19 \rightarrow e...

20.XOÁ SỐ #1

Mã bài: NBDELNUM1

An đang viết dở thư gửi T trên word nhưng lúc không để ý lại bị em Khánh vào phá. Bé Khánh rất thích những con số nên đã gõ bừa các chữ số vào các vị trí bất kỳ trong thư. Hãy tìm lại chuỗi gốc giúp An nhé!

INPUT

Nhập chuỗi S chỉ gồm các ký tự chữ cái in thường và chữ số, có độ dài không vượt quá 10000.

OUTPUT

In ra chuỗi gốc sau khi đã xoá tất cả các chữ số.

INPUT	OUTPUT
co3333d3ed64re5a24m	codedream

21.XOÁ CHỮ HOA

Mã bài: NBDELCAP

Có thể các bạn chưa biết, Sa là người có cách phát âm rất khó nghe. Dù đã cố gắng sửa nhưng cậu vẫn chưa phát âm rõ ràng được, vì thế sếp đã yêu cầu bạn phát minh ra một cái máy có thể phiên dịch được những gì Sa nói. Biết rằng cách phát âm của Sa nhấn mạnh vào chữ cái theo một cách rất riêng, nên hãy xóa bỏ trọng âm của từ đó đi.

INPUT

Nhập chuỗi s chỉ gồm các ký tự chữ cái in thường và in hoa, độ dài của chuỗi không quá 10000. Trong đó chuỗi s là tiếng Sa nói, các ký tự hoa là trọng âm cần phải bỏ.

OUTPUT

Đưa ra kết quả mà đề bài yêu cầu

INPUT	OUTPUT
PcodQeTdream	codedream

22.ĐỔI CHỖ KÝ TỰ

Mã bài: NBSWAPSTR

Cho một chuỗi S và n mệnh lệnh yêu cầu đổi chỗ ký tự ở vị trí i cho ký tự ở vị trí j.

Hỏi: sau n mệnh lệnh, ta có được chuỗi mới là chuỗi gì?

INPUT

- Dòng đầu chứa chuỗi S có độ dài không vượt quá 1000.

- Dòng thứ hai chứa số nguyên dương n ($0 < n \leq 1000$) là số mệnh lệnh.

- n dòng tiếp theo, mỗi dòng chứa 2 số nguyên i, j thể hiện mệnh lệnh cần đổi chỗ ký tự thứ i cho ký tự thứ j (i, j đảm bảo là các vị trí nằm trong chuỗi S).

OUTPUT

In ra chuỗi sau khi thực hiện n mệnh lệnh.

INPUT	OUTPUT
ocreddeam	codedream
2	
1 2	
3 6	

23.KÝ TỰ GIỐNG NHAU

Mã bài: NBCHKCOM

Cho hai chuỗi s_1, s_2 chỉ bao gồm các chữ cái trong bảng chữ cái tiếng Anh.

Bạn cần trả lời n câu hỏi dạng: Cho số nguyên i , hỏi tại vị trí i của hai chuỗi s_1 và s_2 có giống nhau hay không?

INPUT

- Dòng đầu tiên chứa hai chuỗi s_1, s_2 cách nhau bởi một dấu cách gồm các ký tự thường, độ dài hai chuỗi s_1, s_2 không quá 10000.

- Dòng thứ hai chứa số nguyên n ($0 < n \leq 1000$) là số câu hỏi.

- n dòng tiếp theo, mỗi dòng chứa một số nguyên không âm i là vị trí cần kiểm tra (vị trí i đảm bảo thuộc hai chuỗi s_1, s_2).

OUTPUT

Với mỗi câu hỏi, in ra “YES” nếu tại vị trí cần kiểm tra của hai chuỗi có ký tự giống nhau và “NO” nếu ngược lại. Mỗi câu trả lời in trên một dòng.

INPUT	OUTPUT
hihi hello	YES
2	NO
1	
4	

24.KÝ TỰ CHUNG

Mã bài: NBCNTCOM

Cho hai chuỗi ký tự s_1 , s_2 . Đếm số lượng ký tự cùng xuất hiện trong cả hai chuỗi.

INPUT

Nhập lần lượt chuỗi s_1 , s_2 có độ dài không vượt quá 1000, mỗi chuỗi nằm trên một dòng.

OUTPUT

Đưa ra kết quả mà đề bài yêu cầu.

INPUT	OUTPUT
abcdabc223 dbcgh34	4

Gợi ý:

- Tạo một chuỗi s_3 gồm các ký tự khác nhau nằm trong chuỗi s_1 .
- Xét từng ký tự thuộc chuỗi s_1 , nếu ký tự đó cũng thuộc chuỗi s_2 thì kết quả tăng lên 1.

CHƯƠNG 5. DỮ LIỆU KIỂU MẢNG

5.1. Mảng một chiều

Mảng một chiều là một cấu trúc dữ liệu cho phép lưu trữ nhiều giá trị có cùng kiểu dữ liệu trong một dãy liên tiếp của bộ nhớ. Mỗi phần tử trong mảng được xác định bằng một chỉ số số nguyên, thường bắt đầu từ 0 cho phần tử đầu tiên.

Cú pháp khai báo mảng 1 chiều:

kiểu_dữ_liệu_tên_mảng [kích_thước_mảng]

Trong đó:

- *kiểu_dữ_liệu*: Là kiểu dữ liệu của các phần tử trong mảng (ví dụ: int, double, char, ...).

- *tên_mảng*: Là tên mảng, tương tự tên biến.

- *kích_thước_mảng*: Là số phần tử trong mảng. Kích thước phải là một số nguyên không âm.

Dưới đây là một ví dụ về cách khai báo và sử dụng mảng một chiều trong C++:

Ví dụ 1:

```
int main() {  
    // Khai báo và khởi tạo mảng  
    int numbers[5] = {10, 20, 30, 40, 50};  
    // Truy cập và in các phần tử của mảng  
    cout << "Phần tử thứ 2 của mảng: " <<  
    numbers[1] << endl; //20  
    cout << "Phần tử thứ 4 của mảng: " <<  
    numbers[3] << endl; //40  
    // Sử dụng vòng lặp để duyệt và in tất cả các  
    phần tử trong mảng  
    for (int i = 0; i < 5; i++) {  
        cout << "Phần tử thứ " << i << ": " <<  
        numbers[i] << endl;  
    }  
}
```

```

    }
    return 0;
}

```

Trong ví dụ 1, một mảng *numbers* gồm 5 phần tử được khai báo và khởi tạo giá trị ban đầu lần lượt là 10, 20, 30, 40, 50. Vì các phần tử được đánh số từ 0 đến 4. Nên phần tử thứ 2 của mảng được truy cập bằng cách gọi *numbers[1]*, trả về kết quả là 20, phần tử thứ 4 của mảng được truy cập bằng cách gọi *numbers[3]*, trả về kết quả là 40.

Ví dụ 2:

```

int main() {
    const int SIZE = 5; // Kích thước của mảng
    int numbers[SIZE]; // Khai báo mảng
    // Nhập dữ liệu cho mảng
    for (int i = 0; i < SIZE; i++) {
        cout << "Nhập giá trị cho phần tử thứ " <<
i << ": ";
        cin >> numbers[i];
    }
    // In dữ liệu của mảng
    cout << "Các phần tử của mảng là: ";
    for (int i = 0; i < SIZE; i++) {
        cout << numbers[i] << " ";
    }
    return 0;
}

```

Trong ví dụ 2, ta khai báo một mảng *numbers* có kích thước *SIZE* (thay đổi tùy ý) và sau đó sử dụng vòng lặp để nhập dữ liệu cho từng phần tử của mảng.

Vòng lặp yêu cầu người dùng nhập giá trị cho phần tử thứ *i* (với *i* nhận giá trị từ 0 đến *SIZE-1*), và sau đó lưu giá trị này vào *numbers[i]*.

Cuối cùng, chúng ta sử dụng vòng lặp khác để in ra các phần tử của mảng.

5.2. Mảng hai chiều

Mảng hai chiều là một cấu trúc dữ liệu cho phép lưu trữ các giá trị trong một bảng có hàng và cột. Mảng hai chiều được tổ chức theo một cấu trúc lưới, với mỗi ô (phần tử) được xác định bởi một cặp chỉ số (hàng, cột). Mỗi phần tử trong mảng được xác định bằng một cặp chỉ số số nguyên hàng và cột, thường bắt đầu từ cặp (0, 0) cho phần tử đầu tiên.

Cú pháp khai báo mảng 2 chiều:

kiểu_dữ_liệu_tên_mảng [số_hàng] [số_cột]

Trong đó:

- *kiểu_dữ_liệu*: Là kiểu dữ liệu của các phần tử trong mảng (ví dụ: int, double, char, ...).
- *tên_mảng*: Là tên của mảng.
- *số_hàng* và *số_cột*: Là số lượng hàng và cột trong mảng.

Ví dụ 1:

```
int matrix[3][4] = {
    {1, 2, 3, 4},
    {5, 6, 7, 8},
    {9, 10, 11, 12}
};

cout << "Phần tử ở hàng 1 cột 1 là:" <<
matrix[0][0] << endl; // 1
cout << "Phần tử ở hàng 3 cột 2 là:" <<
matrix[2][1] << endl; // 10
// In dữ liệu của mảng
cout << "Các phần tử của mảng là:" << endl;
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 4; j++) {
        cout << matrix[i][j] << " ";
    }
    cout << endl;
}
```

Trong ví dụ 1, ta khai báo và khởi tạo giá trị cho mảng hai chiều *matrix* có 3 hàng và 4 cột. Mỗi hàng chứa các giá trị tương ứng trong dấu ngoặc nhọn {...}. Các hàng và cột của mảng sẽ được đánh số bắt đầu từ 0, theo chiều từ trên xuống dưới, từ trái sang phải.

Ví dụ 2:

```
int main() {
    const int ROWS = 3;    // Số hàng
    const int COLS = 4;    // Số cột
    int matrix[ROWS][COLS]; // Khai báo mảng hai chiều

    // Nhập dữ liệu cho mảng
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            cout << "Nhập giá trị phần tử hàng "
            << i << ", cột " << j << ": ";
            cin >> matrix[i][j];
        }
    }

    // In dữ liệu của mảng
    cout << "Các phần tử của mảng là:" << endl;
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

Trong ví dụ 2 trên, một mảng hai chiều *matrix* với ROWS hàng và COLS cột được khai báo (kích thước thay đổi tùy ý). Ta sử dụng hai vòng lặp lồng nhau để nhập dữ liệu cho từng phần tử của mảng và sau đó sử dụng vòng lặp khác để in ra các phần tử của mảng.

5.3. Kiểu Vector

Vector là một lớp (class) trong thư viện chuẩn (STL - Standard Template Library), nó có thể sử dụng để tạo và quản lý mảng động, tức là mảng có thể thay đổi kích thước linh hoạt trong quá trình chạy. Để sử dụng vector, ta cần khai báo thư viện `<vector>`

Cú pháp khai báo vector:

`vector<kiểu_dữ_liệu> tên_vector;`

Trong đó:

- *kiểu_dữ_liệu*: Là kiểu dữ liệu của các phần tử trong vector (ví dụ: int, double, char, ...).

- *tên_vector*: Là tên của vector.

Ví dụ 1:

```
#include <iostream>
#include <vector>
int main() {
    // Khai báo và khởi tạo vector
    vector<int> numbers = {1, 2, 3, 4, 5};
    // Thêm phần tử vào vector
    numbers.push_back(6);
    // Truy cập và in các phần tử của vector
    for (int i = 0; i < numbers.size(); i++) {
        cout << numbers[i] << " ";
    }
    return 0;
}
```

Trong ví dụ 1, ta tạo một vector chứa các số nguyên 1, 2, 3, 4, 5. Sau đó, ta đã sử dụng phương thức *push_back* để thêm một phần tử mới vào vector. Các phần tử trong vector cũng được đánh số bắt đầu từ 0 giống như các phần tử trong mảng.

Ví dụ 2:

```
#include <iostream>
#include <vector>
int main() {
    vector<int> numbers;
    int n;
    cout << "Nhập số lượng phần tử của vector: ";
    cin >> n;
    // Nhập dữ liệu cho vector
    for (int i = 0; i < n; i++) {
        int value;
        cout << "Nhập giá trị cho phần tử thứ " <<
i << ": ";
        cin >> value;
        numbers.push_back(value);
    }
    // In dữ liệu của vector
    cout << "Các phần tử của vector là:";
    for (int i = 0; i < numbers.size(); i++) {
        cout << " " << numbers[i];
    }
    return 0;
}
```

Trong ví dụ 2, mô tả cách nhập dữ liệu cho các phần tử trong vector. Ban đầu, vector *numbers* được khai báo và không có phần tử nào. Sau đó, khi người dùng nhập dữ liệu cho vector, bắt buộc phải sử dụng phương thức `push_back` để thêm một phần tử vào cuối của vector do vector không có sẵn chỗ trống để có thể truy cập thẳng `numbers[i]` như sử dụng mảng, phương thức `push_back` sẽ vừa tăng kích thước của vector lên 1, vừa gán giá trị của vị trí cuối cùng trong vector bằng giá trị tham số trong phương thức.

Một số hàm của Vector trong C++

5.3.1. Hàm lấy độ dài của vector

Cú pháp: `size()`

Ví dụ: Có vector `numbers = {1, 2, 3, 4, 5};`

- Lệnh `numbers.size()` trả về kết quả độ dài vector là 5.

Code minh họa:

```
vector<int> numbers = {1, 2, 3, 4, 5};  
cout << s.size() << endl; //5
```

5.3.2. Hàm thay đổi kích thước

Cú pháp: `resize(SIZE)`

Trong đó:

- `SIZE` là kích thước muốn thay đổi

Sau khi thay đổi kích thước, nếu kích thước mới lớn hơn kích thước cũ thì các phần tử mới ở trong vector sẽ được gán giá trị mặc định là 0.

Ví dụ: điều chỉnh kích thước của vector thành 100 ta gọi lệnh `a.resize(100)`

Code minh họa:

```
vector<int> numbers = {1, 2, 3, 4, 5};  
numbers.resize(7); // numbers = {1, 2, 3, 4, 5,  
0, 0};  
cout << s.size() << endl; //7
```

5.3.3. Hàm khởi tạo giá trị cho vector

Cú pháp: `assign(size, value)`

Trong đó:

- `size`: kích thước cần khởi tạo

- `value`: giá trị của từng phần tử trong vector

Ví dụ: Khởi tạo vector `numbers` có 5 phần tử, các phần tử có giá trị bằng 3 ta gọi lệnh `numbers.assign(5, 3)`

Code minh họa:

```
vector<int> numbers;  
numbers.assign(5, 3); // numbers = {3, 3, 3, 3, 3};
```

5.3.4. Hàm thêm một phần tử vào cuối vector

Cú pháp: push_back (x)

Trong đó: *x* là giá trị của phần tử cần thêm

Ví dụ: Để thêm phần tử 3 vào cuối vector *numbers* ta gọi lệnh *numbers.push_back(3)*

Code minh họa:

```
vector<int> numbers = {1, 2, 3, 4, 5};  
numbers.push_back(6); // numbers = {1, 2, 3, 4, 5, 6};
```

5.3.5. Hàm bỏ một phần tử vào cuối vector

Cú pháp: pop_back ()

Ví dụ: Để bỏ một phần tử ở cuối vector *numbers* ta gọi lệnh *numbers.pop_back()*

Code minh họa:

```
vector<int> numbers = {1, 2, 3, 4, 5};  
numbers.pop_back(); // numbers = {1, 2, 3, 4};
```

5.3.6. Hàm chèn một phần tử vào vector

Cú pháp: insert (position, value)

Trong đó:

- *position*: vị trí cần chèn phần tử mới

- *value*: là giá trị phần tử cần chèn

Ví dụ: Để thêm một phần tử có giá trị là 2 vào vào vị trí 1 trong vector *numbers* ta gọi lệnh *numbers.insert(1, 2)*

Code minh họa:

```
vector<int> numbers = {1, 2, 3, 4, 5};  
numbers.insert(1, 2); // numbers = {1, 2, 2, 3, 4, 5};
```

5.3.7. Hàm xóa phần tử trong vector

Cú pháp: erase (position) / erase (start, end)

Trong đó:

- *position*: Vị trí cần xoá.

- *start, end*: Vị trí bắt đầu và kết thúc của đoạn cần xoá.

Ví dụ:

- Để xoá phần tử ở vị trí 1 trong vector *numbers* ta gọi lệnh *numbers.erase(1)*;

- Để xoá phần tử từ vị trí 1 đến vị trí 4 trong vector *numbers* ta gọi lệnh *numbers.erase(1,4)*.

Code minh hoạ:

```
vector<int> numbers = {1, 2, 3, 4, 5, 6, 7, 8};  
numbers.erase(1); // numbers = {1, 3, 4, 5, 6, 7, 8};  
numbers.erase(1,4); // numbers = {1, 7, 8};
```

5.3.8. Hàm xoá toàn bộ phần tử trong vector

Cú pháp: clear()

Ví dụ:

- Để xoá toàn bộ phần tử trong vector *numbers* ta gọi lệnh *numbers.clear()*;

Code minh hoạ:

```
vector<int> numbers = {1, 2, 3, 4, 5, 6, 7, 8};  
numbers.clear(); // numbers = {};
```

5.4. Bài tập thực hành

1. TÍNH TỔNG #1

Mã bài: NBSUMARR1

Cho một dãy gồm n số nguyên, tính tổng dãy số đó.

INPUT

- Dòng 1: Chứa số nguyên n ($1 \leq n \leq 10^5$).

- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 , cách nhau bởi một dấu cách.

OUTPUT

In ra tổng của dãy số.

INPUT	OUTPUT
4 2 3 4 5	14

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
int a[100005];
int main() {
    int n;
    cin >> n;
    //nhập dãy a
    for (int i = 1; i <= n; i++)
        cin >> a[i];
    //tính tổng
    long long sum = 0;
    for (int i = 1; i <= n; i++)
        sum = sum + a[i];
    cout << sum;
    return 0;
}
```

2. TÍNH TỔNG #2

Mã bài: NBSUMARR2

Cho một dãy gồm n số nguyên, tính tổng các số trong dãy cho đến khi gặp số 0.

INPUT

- Dòng 1: Chứa số nguyên n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 , cách nhau bởi một dấu cách.

OUTPUT

In ra tổng các số đã nhập cho đến khi gặp số 0. Nếu không xuất hiện số 0, đưa ra tổng của cả dãy đó.

INPUT	OUTPUT
7 2 3 4 5 0 1 2	14

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
int a[100005];
int main() {
    int n;
    cin >> n;
    //nhập dãy a
    for (int i = 1; i <= n; i++)
        cin >> a[i];
    //tính tổng
    long long sum = 0;
    for (int i = 1; i <= n; i++)
        if (a[i] != 0)
            sum = sum + a[i];
        else
            break;
    cout << sum;
    return 0;
}
```

3. TÍNH TỔNG #3

Mã bài: NBSUMARR3

Cho một dãy gồm n số nguyên, tính trung bình cộng của dãy số đó.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 , cách nhau bởi một dấu cách.

OUTPUT

In ra trung bình cộng của dãy số, lấy đến 2 chữ số sau phần thập phân.

INPUT	OUTPUT
3 19 20 21	20.00

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
int a[100005];
int main() {
    int n;
    cin >> n;
    //nhập dãy a
    for (int i = 1; i <= n; i++)
        cin >> a[i];
    //tính tổng
    long long sum = 0;
    for (int i = 1; i <= n; i++)
        sum = sum + a[i];
    cout << fixed << setprecision(2) <<
(double)sum/n;
    return 0;
}
```

4. TÍNH TỔNG #4

Mã bài: NBSUMARR4

Cho một dãy gồm n số nguyên. Hãy tính tổng các số trong dãy thoả mãn: chia 5 dư 1, đồng thời chia hết cho 2.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 , cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
7 6 5 3 16 2 3 5	22

5. TÍNH TỔNG #5

Mã bài: NBSUMARR5

Bé Khánh đang dùng máy tính casio để tính toán, nhưng chiếc máy tính của Khánh hình như đang hỏng. Hãy giúp bé tính toán n số được liệt kê sẵn trên giấy. Với số ở vị trí lẻ tính từ đầu dãy, hãy thực hiện phép cộng. Còn nếu số đó ở vị trí chẵn, hãy thực hiện phép trừ.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 , cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT	Giải thích
3 1 2 3	2	$1 - 2 + 3 = 2$
5 7 4 5 1 3	10	$7 - 4 + 5 - 1 + 3 = 10$

6. TÍNH TỔNG #6

Mã bài: NBSUMARR6

Hôm nay bạn An được học về trung bình cộng của dãy số. Tuy nhiên An lại không thích số âm. Bạn hãy giúp An tính trung bình cộng của các phần tử không âm trong dãy số nhé.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 , cách nhau bởi một dấu cách.

OUTPUT

In ra một số thực là trung bình cộng của các phần tử không âm (lấy 2 chữ số thập phân). Nếu không có số không âm nào trong dãy thì in ra số -1.

INPUT	OUTPUT
6 -3 4 -2 0 5 2	2.75

7. SỐ ĐẶC BIỆT #2

Mã bài: NBSPEC2

Số đặc biệt là số nguyên dương chia hết cho tổng các chữ số của nó. Cho một dãy gồm n số nguyên. Hãy đếm xem trong dãy số đó có bao nhiêu số đặc biệt?

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^{18} , cách nhau bởi một dấu cách.

OUTPUT

In ra số lượng số đặc biệt trong dãy.

INPUT	OUTPUT	Giải thích
5 1 3 21 -12 5	4	Các số đặc biệt là: 1, 3, 5, 21

8. SỐ NGUYÊN TỐ TRONG DÃY

Mã bài: NBPRIME5

Cho một dãy gồm n số nguyên. Hãy đếm xem trong dãy số đó có bao nhiêu số nguyên tố.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^3$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 , cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
5 1 2 3 -4 5	3

9. ƯỚC NGUYÊN TỐ

Mã bài: NBPRIME6

Cho hai số nguyên n , m và 1 dãy số gồm m số nguyên. Với mỗi số nguyên thuộc dãy số đó, hãy kiểm tra xem nó có phải là ước nguyên tố của n hay không (vừa là số nguyên tố, vừa là ước của số n).

INPUT

- Dòng 1: Chứa hai số nguyên dương n , m ($1 \leq n, m \leq 10^3$) cách nhau bởi một dấu cách.
- m dòng tiếp theo, mỗi dòng chứa một số nguyên dương có giá trị tuyệt đối không vượt quá 10^9 ngăn cách nhau bởi một dấu cách.

OUTPUT

In ra m dòng tương ứng với đáp án của từng số trong dãy, nếu số đó là ước nguyên tố của n thì in ra "YES", ngược lại in ra "NO".

INPUT	OUTPUT
12 5	NO
1	YES
2	YES
3	NO
4	NO
5	

10. VỊ TRÍ

Mã bài: NBRANK

Cho hai số n , x và n số nguyên phân biệt, xác định xem nếu sắp xếp lại dãy theo thứ tự tăng dần thì x đứng thứ mấy trong dãy.

INPUT

- Dòng 1: Chứa số nguyên n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 ngăn cách nhau bởi một dấu cách.
- Dòng 3: Chứa số nguyên x ($|x| \leq 10^9$).

Dữ liệu đảm bảo x là số thuộc dãy.

OUTPUT

In ra thứ tự của số x sau khi đã sắp xếp.

INPUT	OUTPUT
5 2 3 1 5 4 4	4

11. TẦN SUẤT #1

Mã bài: NBFREQ1

Cho số n , x và dãy số có n số nguyên. Đếm số lần xuất hiện của số x trong dãy số đó.

INPUT

- Dòng 1: Chứa số nguyên n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 ngăn cách nhau bởi một dấu cách.
- Dòng 3: Chứa số nguyên x ($|x| \leq 10^9$).

OUTPUT

In ra số lần xuất hiện của x.

INPUT	OUTPUT
5 1 4 1 4 1 4	2

12. TẦN SUẤT #2

Mã bài: NBFREQ2

Cho số nguyên dương n và một dãy chứa n số nguyên. Hãy đếm số lượng số chia hết cho 4 xuất hiện trong dãy.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 ngăn cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
2 1 0	1
5 1 -8 -1 2 9	1
8 10 -16 2 -8 24 8 4 1	5

13. TẦN SUẤT #3

Mã bài: NBFREQ3

Cho số n , k và một dãy chứa n số nguyên. Hãy đếm số lượng số chia hết cho k xuất hiện trong dãy.

INPUT

- Dòng 1: Chứa số nguyên n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 ngăn cách nhau bởi một dấu cách.
- Dòng 3: chứa số nguyên k ($1 \leq k \leq 10^9$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
5 2 1 -8 -1 2 9	2
8 3 10 -16 2 -8 24 8 4 1	1

14. TẦN SUẤT #4

Mã bài: NBFREQ4

Cho số n , x và một dãy chứa n số nguyên. Hãy đếm số lượng số lớn hơn hoặc bằng x xuất hiện trong dãy.

INPUT

- Dòng 1: Chứa số nguyên n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 ngăn cách nhau bởi một dấu cách.
- Dòng 3: chứa số nguyên x ($|x| \leq 10^9$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
5 1 4 5 2 3 3	3

15. TẦN SUẤT #5

Mã bài: NBFREQ5

Cho số n và một dãy chứa n số nguyên. Hãy đếm số lượng số xuất hiện từ 2 lần trở lên.

INPUT

- Dòng 1: Chứa số nguyên n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^6 ngăn cách nhau bởi một dấu cách.

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
7 9 4 2 4 9 5 4	2

16. TẦN SUẤT #6

Mã bài: NBFREQ6

Cho số n , x và một dãy chứa n số nguyên. Hãy tính xem phải cần thêm ít nhất bao nhiêu số có giá trị tuyệt đối nhỏ

hơn hoặc bằng x vào dãy để tổng các phần tử trong dãy là 0.

INPUT

- Dòng 1: Chứa số nguyên n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 ngăn cách nhau bởi một dấu cách.
- Dòng 3: Chứa số nguyên x ($|x| \leq 10^9$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT	Giải thích
2 -1 1 2 2	1	Tổng dãy số là 2, cần cộng thêm 1 lần số -2 vào để tổng dãy số bằng 0.
4 -1 1 2 -5 -4 2	4	Tổng dãy số là -7, cần cộng thêm 3 lần số 2 và 1 lần số 1 vào để tổng dãy số bằng 0.

17. TÂN CÔNG THÀNH TRÌ

Mã bài: NBATTACK

Tùng là một vị vua bất bại, một chiến binh mà khiến cho bất cứ ai nghe đến tên cũng phải khiếp sợ, nhưng giờ đây, sau khi trải qua hơn 10 năm chìm đắm trong vòng xoay rượu chè, Sa đã bị tha hóa. Sau những chuỗi ngày 18 tiếng liên tục, Sa nhận ra sự sai lầm của mình, anh ấy muốn tìm lại vị thế của mình ngày xưa nên đã quay trở lại chiến trường.

Có n thành trì với các giá trị cụ thể (có thể âm) được Sa đưa vào tầm ngắm. Cú đánh bom sẽ làm phá huỷ một đoạn liên tiếp các thành trì.

Sa đã chọn ra được m phương án để tấn công. Mỗi phương án được miêu tả bởi hai số x, y là vị trí bắt đầu và kết thúc của đoạn các thành trì có thể phá huỷ được. Lợi ích của phương án tấn công bằng tổng giá trị các thành trì trong đoạn từ x đến y .

Vì chỉ có duy nhất một quả bom, nhưng do quá dốt toán nên anh ấy loay hoay cả nửa ngày mà không tính được kết quả cho từng phương án của mình. Bạn hãy giúp anh ấy tính toán lợi ích trong từng phương án nhé!

INPUT

- Dòng đầu chứa 2 số nguyên n và m cách nhau bởi một dấu cách ($1 \leq n, m \leq 30000$).
- Dòng tiếp theo chứa n số nguyên có giá trị tuyệt đối không vượt quá 1000, mô tả giá trị của các thành trì, các số được viết cách nhau bởi một dấu cách.
- m dòng tiếp theo, mỗi dòng chứa 2 số x, y ($1 \leq x \leq y \leq n$) mô tả phương án tấn công của Sa.

OUTPUT

Gồm m dòng, dòng thứ i ghi ra lợi ích của phương án tấn công thứ i .

INPUT	OUTPUT
5 3	3
1 -8 -1 2 9	-1
1 5	-7
3 3	
2 4	

18. DÂY PALINDROME

Mã bài: NBARRPALIN

Là một người hướng đến sự hoàn hảo, bé Khánh đặc biệt thích những thứ có sự đối xứng, vì thế nên bút chì của bé Khánh có hân 2 đầu viết được (hoặc hai đầu đều không viết được). Đạo gân đây, Khánh biết được một thứ vô cùng mới mẻ là chuỗi đối xứng (chuỗi Palindrome).

“Chuỗi đối xứng là chuỗi khi đọc từ trái qua phải hay từ phải qua trái đều ra một kết quả giống nhau”.

Bé Khánh đã dành cả một buổi tối để tìm hiểu về nó và phát hiện ra không chỉ dãy ký tự mà dãy số cũng có thể là một dãy đối xứng.

Hãy giúp bé Khánh kiểm tra xem, với dãy số mà bé nghĩ ra, đó có phải là một dãy palindrome không nhé!

INPUT

- Dòng đầu tiên chứa số n ($1 \leq n \leq 30000$) là số lượng số của dãy số.
- Dòng tiếp theo chứa n số nguyên có giá trị tuyệt đối không vượt quá 30000 mô tả dãy số, các số nguyên cách nhau bởi dấu cách.

OUTPUT

Nếu dãy số của bé đối xứng thì in ra “YES”, ngược lại thì in ra “NO”.

INPUT	OUTPUT
5 1 2 3 2 1	YES
5 1 2 3 4 5	NO

19.SỐ FIBONACCI #1

Mã bài: NBFIBO1

Dãy số Fibonacci là dãy số có dạng:

$$F_1 = 1$$

$$F_2 = 1$$

$$F_i = F_{i-1} + F_{i-2} \ (i > 2)$$

Cho số nguyên dương n . In ra số Fibonacci thứ n .

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 40$).

OUTPUT

Số Fibonacci thứ n .

INPUT	OUTPUT
6	8

20.SỐ FIBONACCI #2

Mã bài: NBFIBO2

Cho số nguyên dương n . In ra số Fibonacci lớn nhất mà nhỏ hơn hoặc bằng n .

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10^5$).

OUTPUT

Số Fibonacci lớn nhất mà nhỏ hơn hoặc bằng n .

INPUT	OUTPUT
6	5

21.SỐ FIBONACCI #3

Mã bài: NBFIBO3

Cho 1 số nguyên dương n . Hãy kiểm tra xem số n có phải số Fibonacci ko.

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10^9$).

OUTPUT

In ra “YES” nếu n là số Fibonacci, “NO” nếu không phải.

INPUT	OUTPUT
6	NO
3	YES

22.SỐ FIBONACCI #4

Mã bài: NBFIBO4

Cho số nguyên dương n . In ra chuỗi Fibonacci thứ n , biết:

$F_1 = \text{“A”}$

$F_2 = \text{“B”}$

$F_i = F_{i-1} + F_{i-2}$ ($i > 2$)

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 20$).

OUTPUT

In ra chuỗi Fibonacci thứ n .

INPUT	OUTPUT
4	BAB

23.SỐ FIBONACCI #5

Mã bài: NBFIBO5

Cho số nguyên dương n . Tính tổng các số Fibonacci từ 1 đến n .

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 15$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
3	4

24.LỚN NHẤT

Mã bài: NBARRMIMA1

An thích những thứ bé bé nhưng T lại không như vậy. Để chiều lòng T, An quyết định tặng cho cô phần tử lớn nhất của dãy số chứa n số nguyên. Hãy giúp An tìm nó nhé!

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^5 , mỗi số cách nhau bởi một dấu cách.

OUTPUT

In ra số lớn nhất của dãy.

INPUT	OUTPUT
5 1 3 5 7 6	7

25.NHỎ NHẤT

Mã bài: NBARRMIMA2

Cảm kích vì món quà của An, nhân ngày Valentine Trắng, T quyết định cảm ơn bằng phần tử nhỏ nhất trong dãy số chứa n số nguyên.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^5 , mỗi số cách nhau bởi một dấu cách.

OUTPUT

In ra phần tử nhỏ nhất của dãy.

INPUT	OUTPUT
5 5 9 2 7 9	2

26.NHỎ NHÌ, LỚN NHÌ

Mã bài: NBARRMIMA3

Thấy hai đứa An và T tặng nhau qua lại, Sa cũng thấy ganh tỵ. Nhân ngày độc thân 14/4 Valentine Đen, Sa quyết định tự tặng cho mình phần tử nhỏ nhì và lớn nhì trong dãy số chứa n số nguyên để cho bằng bạn bằng bè. Nhưng tìm hai số đó khó quá, bạn hãy giúp Sa nhé!

INPUT

- Dòng 1: Chứa số nguyên dương n ($4 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên phân biệt có giá trị tuyệt đối không vượt quá 10^5 , mỗi số cách nhau bởi một dấu cách.

OUTPUT

In ra lần lượt số bé thứ nhì và lớn thứ nhì trên cùng một dòng, cách nhau bởi một dấu cách.

INPUT	OUTPUT
5 1 2 3 4 5	2 4

27.BA LỚN NHẤT

Mã bài: NBARRMIMA4

Đề chơi lớn hơn nữa cho hai đứa kia lác mắt. Lần này Sa quyết định nhờ các bạn “Code thủ” giúp mình tìm ra 3 số lớn nhất trong cả dãy số chứa n số nguyên rồi khoe với An và T. Bạn hãy giúp Sa nhé!

INPUT

- Dòng 1: Chứa số nguyên dương n ($3 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^5 , mỗi số cách nhau bởi một dấu cách.

OUTPUT

In ra ba số lớn nhất trong dãy theo thứ tự giảm dần trên cùng một dòng, cách nhau bởi một dấu cách.

INPUT	OUTPUT
5 2 1 3 5 4	5 4 3

28.SẮP XẾP

Mã bài: NBARRSORT

Cho số n và dãy số chứa n số nguyên. In ra dãy số đó theo thứ tự từ bé đến lớn.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^5 , mỗi số cách nhau bởi một dấu cách.

OUTPUT

In ra dãy số theo thứ tự từ bé đến lớn trên cùng một dòng, cách nhau bởi một dấu cách.

INPUT	OUTPUT
5 2 1 3 5 4	1 2 3 4 5

Gợi ý: Sử dụng hàm *sort* để sắp xếp lại dãy số.

- Với mảng:

sort(a, a+n), sắp xếp từ vị trí 0 đến $n-1$.

sort(a+l, a+n+l), sắp xếp từ vị trí l đến n .

sort(a+l, a+r+l), sắp xếp từ vị trí l đến r .

- Với vector:

sort(a.begin(), a.end()), sắp xếp toàn bộ phần tử thuộc vector.

sort(a.begin()+l, a.begin()+r+l), sắp xếp từ vị trí l đến r trong vector.

29. DÂY GIỐNG NHAU

Mã bài: NBARRCMP

Cho hai dãy số chứa n số nguyên. Hãy kiểm tra xem 2 dãy này có giống nhau không? Hai dãy được coi là giống nhau nếu tồn tại một hoán vị của dãy này mà giống hệt dãy kia.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^5 mô tả dãy a , mỗi số cách nhau bởi một dấu cách.
- Dòng 3: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^5 mô tả dãy b , mỗi số cách nhau bởi một dấu cách.

OUTPUT

In ra “YES” nếu 2 dãy giống nhau, “NO” nếu hai dãy không giống nhau

INPUT	OUTPUT
5 2 1 3 5 4 1 2 3 4 5	YES
3 1 1 2 1 2 2	NO

30. KHOẢNG CÁCH

Mã bài: NBDIFF

An là một người hướng nội. Anh giữ khoảng cách với mọi người, tỏ ra vẻ lạnh lùng và thích những khoảng lặng tâm

hòn. Vì vậy bạn hãy giúp An tìm ra khoảng cách lớn nhất giữa hai phần tử liên tiếp trong dãy số chứa n số nguyên.

INPUT

- Dòng 1: Chứa số nguyên dương n ($2 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^5 , mỗi số cách nhau bởi một dấu cách.

OUTPUT

In ra khoảng cách lớn nhất giữa hai phần tử liên tiếp trong dãy.

INPUT	OUTPUT
5 2 8 -2 10 4	12

31.XẾP HÀNG

Mã bài: NBLINEUP

Có n người đang xếp hàng theo chiều dọc, biết chiều cao cụ thể của từng người, hãy kiểm tra xem n người đó có đứng theo thứ tự từ thấp đến cao không (tính từ người thứ nhất).

INPUT

- Dòng đầu chứa số nguyên dương n thể hiện số lượng người ($1 \leq n \leq 10^5$).
- Dòng tiếp theo chứa n số nguyên dương có giá trị tuyệt đối không vượt quá 10^9 thể hiện chiều cao lần lượt của từ người thứ nhất đến người thứ n , mỗi số cách nhau bởi một dấu cách.

OUTPUT

Nếu thứ tự xếp hàng của n người thỏa mãn yêu cầu đề bài thì in ra "YES", ngược lại in ra "NO".

INPUT	OUTPUT
3 1 2 3	YES
3 1 3 2	NO

32.SẮP XẾP SÁCH

Mã bài: NBBOOK

Cũng như bé Bon, bé Thảo là một người theo chủ nghĩa hoàn hảo. Cô bé thích mọi thứ luôn được sắp xếp theo trình tự trước sau, từ nhỏ đến lớn. Vì vậy nên mỗi lần ra hiệu sách, thấy những cuốn sách sắp xếp lung tung không theo một trình tự nào, cô bé cảm thấy rất khó chịu. Chính vì thế, bé Thảo quyết định xin làm nhân viên của hiệu sách để có thể sắp xếp lại những quyển sách này.

Trùng hợp thay, công việc đầu tiên của bé lại chính là sắp xếp sách theo thứ tự để mọi người dễ tìm hơn. Với bản tính hoàn hảo của mình, bé Thảo muốn sắp xếp mã số của quyển sách theo thứ tự từ bé tới lớn, ngoài ra cô bé còn muốn một quyển sách chỉ xuất hiện một lần trên kệ sách (nói cách khác thì mã số của các quyển sách là khác nhau) nên sẽ bỏ đi một số quyển sách giống nhau.

INPUT

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 3000$) là số lượng cuốn sách cần sắp xếp.
- Dòng tiếp theo chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 thể hiện mã số của mỗi cuốn sách, mỗi số cách nhau bởi một dấu cách.

OUTPUT

In ra thứ tự các cuốn sách sau khi sắp xếp và bỏ đi một số cuốn giống nhau.

INPUT	OUTPUT
5 -1 8 -1 2 9	-1 2 8 9

33.GAME

Mã bài: NBGAME

Sa và An cùng chơi một trò chơi có n màn chơi, các màn chơi được đánh số từ 1 tới n . Sa vì chơi game quá nhiều nên đã có thể thông thạo được p màn của trò chơi. An cũng không kém cạnh khi thông thạo được q màn trong đó. Sa và An sẽ hợp tác với nhau để phá đảo trò chơi này, một màn chơi chỉ có thể vượt qua nếu một trong hai người thông thạo nó. Bạn hãy kiểm tra xem liệu 2 đứa nghiện game này có phá đảo được trò chơi hay không.

INPUT

- Dòng đầu chứa số nguyên dương n ($1 \leq n \leq 100$) là số lượng màn chơi.
- Dòng thứ hai chứa số nguyên dương p ($1 \leq p \leq n$) là số lượng màn chơi mà Sa đã thông thạo, ngay sau đó là p số nguyên mô tả số hiệu của các màn chơi mà Sa đã thông thạo, mỗi số cách nhau bởi một dấu cách.
- Dòng thứ ba chứa số nguyên dương q ($1 \leq q \leq n$) là số lượng màn chơi mà An đã thông thạo, ngay sau đó là q số nguyên mô tả số hiệu của các màn chơi mà An đã thông thạo, mỗi số cách nhau bởi một dấu cách.

OUTPUT

In ra “WIN” nếu An và Sa thắng, ngược lại in “LOSE”

INPUT	OUTPUT	Giải thích
4 3 1 2 3 2 2 4	WIN	Có tổng cộng 4 màn chơi, Sa thông thạo màn chơi 1, 2, 3. An thông thạo màn chơi 2, 4. Vậy cả 4 màn chơi đều sẽ có người thông thạo.
4 3 1 2 3 2 1 3	LOSE	

34. TRUY VẤN #1

Mã bài: NBQUERY1

Cho dãy A có n số nguyên và một số nguyên dương q là số truy vấn (yêu cầu) cần thực hiện với dãy số đã cho.

Mỗi truy vấn cho 2 số nguyên dương x, y (không vượt quá n) yêu cầu tăng mỗi phần tử từ vị trí x tới vị trí y trong dãy A lên 1 đơn vị.

Hãy in ra dãy A sau khi thực hiện toàn bộ q truy vấn.

INPUT

- Dòng 1: Chứa hai số nguyên dương n, q ($1 \leq n, q \leq 1000$) cách nhau bởi một dấu cách, lần lượt là số lượng phần tử trong dãy A và số lượng truy vấn.
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 1000, các số ngăn cách nhau bởi một dấu cách.
- q dòng sau, mỗi dòng chứa 2 số nguyên dương x, y ($1 \leq x \leq y \leq n$) cách nhau bởi một dấu cách, là vị trí bắt đầu và vị

trí kết thúc của các phần tử dãy A cần thay đổi trong mỗi truy vấn.

OUTPUT

In ra dãy A sau khi thực hiện toàn bộ các truy vấn.

INPUT	OUTPUT
5 2	2 2 4 1 4
1 0 2 0 3	
1 3	
2 5	

35. TRUY VẤN #2

Mã bài: NBQUERY2

Cho dãy A có n số nguyên và một số nguyên dương q là số truy vấn (yêu cầu) cần thực hiện với dãy số đã cho.

Mỗi truy vấn cho số nguyên dương x (không vượt quá n) yêu cầu tăng mỗi phần tử ở vị trí x, bên trái x và bên phải x (nếu có) lên x đơn vị.

Hãy in ra dãy A sau khi thực hiện toàn bộ q truy vấn.

INPUT

- Dòng 1: Chứa hai số nguyên dương n, q ($1 \leq n, q \leq 10^5$) cách nhau bởi một dấu cách, lần lượt là số lượng phần tử trong dãy A và số lượng truy vấn.
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^5 , các số ngăn cách nhau bởi một dấu cách.
- q dòng sau, mỗi dòng chứa số nguyên dương x ($1 \leq x \leq n$).

OUTPUT

In ra dãy A sau khi thực hiện toàn bộ các truy vấn.

INPUT	OUTPUT
5 2 1 -1 2 0 1 2 3	3 4 7 3 1

36. TAM GIÁC PASCAL

Mã bài: NBPASCAL

Cho số nguyên dương n . In ra tam giác pascal bậc n theo mẫu (xem output mẫu).

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10$).

OUTPUT

Kết quả của bài toán.

INPUT	OUTPUT
5	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1

37. VẼ BẢNG #4

Mã bài: NBDTABLE4

Cho số nguyên dương n . In ra bảng hình vuông có kích thước $n \times n$ (n hàng, n cột) gồm các số nguyên tăng dần từ trái sang phải, từ trên xuống dưới.

INPUT

Nhập số nguyên dương n ($n \leq 1000$).

OUTPUT

In ra bảng theo yêu cầu đề bài.

INPUT	OUTPUT
4	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

38. VẼ BẢNG #5**Mã bài: NBDTABLE5**

Cho số nguyên dương n . In ra bảng hình vuông có kích thước $n \times n$ (n hàng, n cột) gồm các số nguyên tăng dần từ phải sang trái, từ dưới lên trên, mỗi số cách nhau bởi một dấu cách.

INPUT

Nhập số nguyên dương n ($n \leq 1000$).

OUTPUT

In ra bảng theo yêu cầu đề bài.

INPUT	OUTPUT
4	16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

39. VẼ BẢNG #6**Mã bài: NBDTABLE6**

Cho 2 số nguyên dương m, n . In ra bảng hình chữ nhật có kích thước $m \times n$ (m hàng, n cột) gồm các số nguyên tăng dần từ trên xuống dưới, từ trái sang phải, mỗi số cách nhau bởi một dấu cách.

INPUT

Nhập hai số nguyên dương m, n ($m, n \leq 1000$), cách nhau bởi dấu cách.

OUTPUT

In ra bảng theo yêu cầu đề bài.

INPUT	OUTPUT
4 5	1 5 9 13 17 2 6 10 14 18 3 7 11 15 19 4 8 12 16 20

40. VẼ BẢNG #7**Mã bài: NBDTABLE7**

Cho 2 số nguyên dương m, n . In ra bảng hình chữ nhật có kích thước $m \times n$ (m hàng, n cột) gồm các số nguyên tăng dần theo hình zigzag bắt đầu từ trái sang phải, và đi từ dưới lên trên.

INPUT

Nhập hai số nguyên dương m, n ($m, n \leq 1000$), cách nhau bởi dấu cách.

OUTPUT

In ra bảng theo yêu cầu đề bài.

INPUT	OUTPUT	Giải thích
4 5	20 19 18 17 16 11 12 13 14 15 10 9 8 7 6 1 2 3 4 5	

41. VẼ BẢNG #8

Mã bài: NBDTABLE8

Cho 2 số nguyên dương m, n . In ra bảng hình chữ nhật có kích thước $m \times n$ (m hàng, n cột) gồm các số nguyên tăng dần theo hình xoắn ốc (đi cùng chiều kim đồng hồ và bắt đầu từ ô trái trên), mỗi số cách nhau bởi một dấu cách.

INPUT

Nhập hai số nguyên dương m, n ($m, n \leq 1000$), cách nhau bởi dấu cách.

OUTPUT

In ra bảng theo yêu cầu đề bài.

INPUT	OUTPUT
4 5	1 2 3 4 5 14 15 16 17 6 13 20 19 18 7 12 11 10 9 8

42. TỔNG TRÊN MA TRẬN #1

Mã bài: NBMATRIX1

Cho một bảng hình chữ nhật có kích thước m hàng n cột. Các hàng được đánh số từ 1 đến m , các cột được đánh số từ

1 đến n , từ trái sang phải, từ trên xuống dưới. Trên mỗi ô của hình chữ nhật có chứa một số nguyên. Bé Khánh muốn tìm tổng các phần tử của một hình chữ nhật con của hình chữ nhật đó. Hãy giúp bé Khánh nhé!

INPUT

- Dòng đầu tiên chứa hai số nguyên m và n . ($1 \leq m, n \leq 1000$), mỗi số cách nhau bởi dấu cách.
- m dòng tiếp theo, mỗi dòng chứa n số nguyên có giá trị tuyệt đối không quá 1000 cách nhau bởi dấu cách mô tả hình chữ nhật.
- Dòng cuối cùng chứa bốn số nguyên x_1, y_1, x_2, y_2 là tọa độ góc trái trên (x_1, y_1) và phải dưới (x_2, y_2) của một hình chữ nhật nhỏ hơn bên trong hình chữ nhật ban đầu ($1 \leq x_1 \leq x_2 \leq m, 1 \leq y_1 \leq y_2 \leq n$), mỗi số cách nhau bởi dấu cách.

OUTPUT

In ra tổng hình chữ nhật tìm được.

INPUT	OUTPUT
2 3 1 1 9 8 2 9 1 1 2 2	12

43. TỔNG TRÊN MA TRẬN #2

Mã bài: NBMATRIX2

Cho một bảng hình vuông có kích thước n hàng n cột. Các hàng và cột được đánh số từ 1 đến n , từ trái sang phải, từ trên xuống dưới. Trên mỗi ô của hình vuông có chứa một số nguyên. Bé Khánh muốn tìm tổng các phần tử có chỉ số

hàng và cột đều là lẻ của hình vuông đó. Hãy giúp bé Khánh nhé!

INPUT

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 1000$).
- n dòng tiếp theo, mỗi dòng chứa n số nguyên có giá trị tuyệt đối không quá 1000 cách nhau bởi dấu cách mô tả hình vuông.

OUTPUT

In ra kết quả

INPUT	OUTPUT
3 1 2 3 2 3 1 2 3 3	9

44. TỔNG TRÊN MA TRẬN #3

Mã bài: NBMATRIX3

Cho một bảng hình chữ nhật có kích thước m hàng n cột. Các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n , từ trái sang phải, từ trên xuống dưới. Trên mỗi ô của hình chữ nhật có chứa một số nguyên. Bé Khánh muốn tìm tổng các phần tử mà có tổng chỉ số hàng và chỉ số cột là một số chẵn. Hãy giúp bé Khánh nhé!

INPUT

- Dòng đầu tiên chứa hai số nguyên m và n ($1 \leq m, n \leq 1000$) cách nhau bởi dấu cách.
- m dòng tiếp theo, mỗi dòng chứa n số nguyên có giá trị tuyệt đối không quá 1000 cách nhau bởi dấu cách mô tả hình chữ nhật.

OUTPUT

Đưa ra kết quả mà đề bài yêu cầu.

INPUT	OUTPUT
2 3 1 1 9 8 2 9	12

45. TỔNG TRÊN MA TRẬN #4

Mã bài: NBMATRIX4

Cho một bảng hình vuông có kích thước n hàng n cột. Các hàng và các cột được đánh số từ 1 đến n , từ trái sang phải, từ trên xuống dưới. Trên mỗi ô của hình vuông có chứa một số nguyên. Bé Khánh muốn tìm tổng các phần tử nằm trên hai đường chéo chính của hình vuông đó. Hãy giúp bé Khánh nhé!

INPUT

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 1000$).
- n dòng tiếp theo, mỗi dòng chứa n số nguyên có giá trị tuyệt đối không quá 1000 cách nhau bởi dấu cách mô tả hình vuông.

OUTPUT

Đưa ra kết quả mà đề bài yêu cầu.

INPUT	OUTPUT	Giải thích
3 5 -2 3 -1 1 5 -1 4 -3	5	Các số trên hai đường chéo chính và chéo phụ là các số được in đậm.

46. TỔNG TRÊN MA TRẬN #5

Mã bài: NBMATRIX5

Cho một bảng hình vuông có kích thước $n * n$. Các hàng, các cột được đánh số từ 1 đến n , từ trái sang phải, từ trên xuống dưới. Trên mỗi ô của hình vuông có chứa một số nguyên. Bé Khánh muốn tìm tổng các phần tử của các hình vuông có tâm trùng với tâm của bảng. Hãy giúp bé Khánh nhé!

INPUT

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 100$).
- n dòng tiếp theo, mỗi dòng chứa n số nguyên có giá trị tuyệt đối không quá 1000 cách nhau bởi dấu cách mô tả hình vuông.

OUTPUT

In ra nhiều dòng, mỗi dòng chứa tổng các phần tử thuộc hình vuông có tâm là tâm của bảng. In ra theo thứ tự tăng dần của độ dài cạnh hình vuông.

INPUT	OUTPUT
3 1 1 9 8 2 9 1 4 6	2 41
4 1 1 9 4 8 2 9 2 1 4 6 3 1 1 1 1	21 54

47. THĂM HIỂM

Mã bài: NBEXPLOR

Cho một mê cung không tường hình chữ nhật có diện tích $m \times n$ (m hàng, n cột). Các hàng được đánh số từ 1 tới m, các cột được đánh số từ 1 đến n, từ trái sang phải, từ trên xuống dưới. Trên mỗi ô vuông 1×1 của mê cung đều có đánh một con số thể hiện giá trị của rương kho báu trong ô đó. Nhà thám hiểm Dũng là một người rất biết cách chọn, anh chỉ chọn những rương kho báu xịn nhất (kho báu có giá trị lớn nhất) để mang về. Bạn hãy giúp nhà thám hiểm Dũng tìm xem có bao nhiêu rương kho báu xịn nhất xuất hiện trong mê cung nhé!

INPUT

- Dòng đầu tiên chứa hai số nguyên m, n ($1 \leq m, n \leq 1000$) cách nhau bởi dấu cách.
- m dòng tiếp theo, mỗi dòng chứa n số nguyên có giá trị tuyệt đối không quá 1000 cách nhau bởi dấu cách mô tả giá trị của các kho báu trong mê cung.

OUTPUT

In ra số lượng kho báu xịn nhất xuất hiện.

INPUT	OUTPUT
2 3 1 1 9 8 2 9	2

48. CỘT MAX

Mã bài: NBMAXCOL

Cho một hình chữ nhật có kích thước m hàng n cột. Các hàng được đánh số từ 1 đến m, các cột được đánh số từ 1

đến n, từ trái sang phải, từ trên xuống dưới. Trên mỗi ô của hình chữ nhật có chứa một số nguyên. Bé Khánh muốn tìm ra cột có tổng lớn nhất. Bạn hãy giúp bé Khánh nhé!

INPUT

- Dòng đầu tiên chứa hai số nguyên m, n ($1 \leq m, n \leq 1000$) cách nhau bởi dấu cách.

- m dòng tiếp theo, mỗi dòng chứa n số nguyên có giá trị tuyệt đối không quá 1000 cách nhau bởi dấu cách mô tả hình chữ nhật.

OUTPUT

Đưa ra tổng của cột lớn nhất tìm được và số hiệu của cột có tổng lớn nhất trên cùng một dòng, cách nhau bởi một dấu cách. Nếu có nhiều cột cùng có tổng lớn nhất thì đưa ra cột có số hiệu lớn nhất.

INPUT	OUTPUT
2 3 1 1 9 8 2 9	18 3

49. PHẦN TỬ CHỐT

Mã bài: NBMAXTABLE

Cho một hình chữ nhật có kích thước m hàng n cột. Các hàng được đánh số từ 1 đến m, các cột được đánh số từ 1 đến n, từ trái sang phải, từ trên xuống dưới. Trên mỗi ô của hình chữ nhật có chứa một số nguyên.

Ô (i, j) được gọi là phần tử chốt nếu ô đó có giá trị bằng giá trị nhỏ nhất của hàng i và giá trị lớn nhất của cột j.

Bé Khánh muốn đếm xem có bao nhiêu phần tử chốt trong hình chữ nhật. Bạn hãy giúp bé Khánh nhé!

INPUT

- Dòng đầu tiên chứa hai số nguyên m, n ($1 \leq m, n \leq 1000$) cách nhau bởi dấu cách.
- m dòng tiếp theo, mỗi dòng chứa n số nguyên có giá trị tuyệt đối không quá 1000 cách nhau bởi dấu cách mô tả hình chữ nhật.

OUTPUT

Số lượng phần tử chẵn.

INPUT	OUTPUT
3 4 2 2 2 5 7 9 7 8 5 8 6 7	2

50.SẮP XẾP MA TRẬN

Mã bài: NBMTSORT

Cho ma trận A kích thước m hàng n cột. Sắp xếp các phần tử của ma trận A theo chiều tăng dần, in ra theo chiều xoắn ốc (xem output mẫu).

INPUT

- Dòng đầu tiên chứa hai số nguyên m, n ($1 \leq m, n \leq 1000$) cách nhau bởi dấu cách.
- m dòng tiếp theo, mỗi dòng chứa n số nguyên có giá trị tuyệt đối không quá 1000 cách nhau bởi dấu cách mô tả ma trận A .

OUTPUT

Ma trận sau khi sắp xếp.

INPUT	OUTPUT
4 4	1 2 3 4
4 5 7 1	20 21 32 5
21 32 10 6	13 67 34 6
3 2 9 11	11 10 9 7
13 20 34 67	

51. TRUY VẤN #3

Mã bài: NBQUERY3

Cho một bảng hình chữ nhật có kích thước m hàng n cột. Các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n , từ trái sang phải, từ trên xuống dưới. Trên mỗi ô của hình chữ nhật có chứa một số 0 hoặc 1.

Cho một số nguyên dương q là số truy vấn (yêu cầu) cần thực hiện với bảng số đã cho.

Mỗi truy vấn cho số nguyên dương x (không vượt quá m) yêu cầu đảo ngược toàn bộ giá trị tại các ô thuộc hàng x : Nếu một ô đang chứa số 1 thì sẽ bị thay đổi về số 0, và ngược lại, ô đang chứa số 0 sẽ bị thay đổi về số 1.

Hãy in ra bảng số sau khi thực hiện toàn bộ q truy vấn.

INPUT

- Dòng 1: Chứa ba số nguyên dương m, n, q ($1 \leq m, n, q \leq 1000$) cách nhau bởi một dấu cách, lần lượt là số lượng phần tử trong dãy A và số lượng truy vấn.

- m dòng sau, mỗi dòng chứa n số chỉ có giá trị là 0 hoặc 1, các số ngăn cách nhau bởi một dấu cách.

- q dòng cuối cùng, mỗi dòng chứa số nguyên dương x ($1 \leq x \leq m$) là hàng cần đảo ngược lại toàn bộ giá trị của các ô.

OUTPUT

In ra bảng số sau khi thực hiện q truy vấn.

Lưu ý: Một hàng có thể bị truy vấn nhiều lần.

INPUT	OUTPUT
3 2 2	0 1
1 0	0 0
1 1	0 0
0 0	
1	
2	

CHƯƠNG 6: ĐỆ QUY VÀ BÀI TẬP NÂNG CAO

6.1. Đệ quy

Đệ quy xảy ra khi một hàm gọi chính nó trong quá trình thực thi. Quá trình này có thể tiếp tục lặp lại cho đến khi một điều kiện cơ sở được thỏa mãn, khi đó các cuộc gọi đệ quy sẽ ngừng và hàm sẽ trả về kết quả.

Đệ quy thường được sử dụng để giải quyết các vấn đề có cấu trúc tương tự như các cây, danh sách liên kết, hoặc bất kỳ cấu trúc lặp lại nào.

Một hàm đệ quy thường gồm hai phần:

- Trường hợp cơ sở (base case): Đây là trường hợp đơn giản và cơ bản mà hàm đệ quy dùng để dừng việc gọi đệ quy và trả về kết quả. Nếu không có trường hợp cơ sở, hàm đệ quy có thể lặp vô hạn.

- Bước đệ quy (recursive step): Đây là phần trong hàm đệ quy mà hàm sẽ gọi chính nó với các tham số khác nhau để giải quyết một vấn đề lớn hơn bằng cách giảm dần kích thước của vấn đề.

Một số ví dụ phổ biến về đệ quy bao gồm tính giai thừa, tìm kiếm trong cây, sắp xếp đệ quy (như quicksort và mergesort), và nhiều thuật toán khác.

Ví dụ 1: Sử dụng đệ quy để tính tổng các số nguyên từ 1 đến n.

```
#include <iostream>
int sum(int n) {
    // Trường hợp cơ sở: nếu n là 0, tổng cần tính
    // là 0
    if (n == 0) return 0;
    // Bước đệ quy: tính tổng n và tổng của các số
    // nguyên từ 1 đến (n - 1)
    return n + sum(n - 1);
}
int main() {
    int num = 5;
    cout << "Tổng các số từ 1 đến " << num << " là
    " << sum(num) << endl;
    return 0;
}
```

Với đệ quy, hàm sum(5) sẽ hoạt động như sau:

$$\text{sum}(5) = 5 + \text{sum}(4)$$

$$\text{sum}(4) = 4 + \text{sum}(3)$$

$$\text{sum}(3) = 3 + \text{sum}(2)$$

$$\text{sum}(2) = 2 + \text{sum}(1)$$

$$\text{sum}(1) = 1 + \text{sum}(0)$$

$$\text{sum}(0) = 0 \text{ (base case)}$$

Ví dụ 2: Sử dụng đệ quy để tính giai thừa

```
#include <iostream>

int factorial(int n) {
    // Trường hợp cơ sở: Giai thừa của 0 và 1 là 1
    if (n == 0 || n == 1) return 1;
    // Bước đệ quy: tính giai thừa của n dựa trên
    // giai thừa của (n - 1)
    return n * factorial(n - 1);
}

int main() {
    int num = 5;
    cout << "Giai thừa của " << num << " là " <<
    factorial(num) << endl;
    return 0;
}
```

6.2. Bài tập thực hành

1. SỐ CATALAN

Mã bài: NBCATALAN

Số Catalan là số được định nghĩa như sau:

$$C_0 = 1$$

$$C_1 = 1$$

$$C_i = \frac{(2i)!}{(i+1)!i!} = \sum_{j=0}^{i-1} C_j C_{i-1-j}$$

Các số đầu tiên của dãy Catalan là 1, 1, 2, 5, 14, 42, 132 ...

Cho số nguyên dương n. Hãy tìm số Catalan thứ n.

INPUT

Nhập số nguyên không âm n ($0 \leq n \leq 13$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
4	14

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
long long catalan(int i){
    if (i <= 1) return 1;
    long long res = 0;
    for (int j = 0; j < i; j++)
        res += catalan (j) * catalan (i-1-j);
    return res;
}
int main() {
    int n;
    cin >> n;
    cout << catalan (n);
    return 0;
}
```

2. ƯỚC CHUNG LỚN NHẤT #1

Mã bài: NBGCD1

Nhà hiền triết vĩ đại A đã từng dạy: “Phàm là hai cá thể độc lập đang cùng tồn tại, giữa họ sẽ là luôn có những điểm chung, từ đó mà có thể dẫn đến sự phát triển trong mối quan hệ của hai cá thể”.

Tuy vậy, theo ý kiến không mấy khách quan của hội knpy, dẫn đầu là Sa, đã đưa ra quan điểm rằng lời dạy đó không đúng khi áp dụng vào mối quan hệ của bạn với crush.

Cho hai số thể hiện đặc điểm của hai cá thể, bạn hãy đưa ra ước chung lớn nhất của hai số để chứng minh lời dạy của nhà hiền triết là đúng nhé!

INPUT

Nhập hai số nguyên a, b cách nhau bởi một dấu cách, thể hiện đặc điểm của hai cá thể ($0 < a, b \leq 10^{18}$).

OUTPUT

Đưa ra kết quả của bài toán.

INPUT	OUTPUT
2 4	2
21 6	3

Gợi ý: Sử dụng thuật toán Euclid để tìm UCLN và cài đặt bằng đệ quy.

Code mẫu:

```
#include <bits/stdc++.h>
using namespace std;
long long gcd (long long a, long long b){
    if (a == 0 || b == 0) return a + b;
    if (a == b) return a;
    return gcd (a % b, b % a);
}
int main() {
    int a, b;
    cin >> a >> b;
    cout << gcd (a, b);
    return 0;
}
```

3. ƯỚC CHUNG LỚN NHẤT #2

Mã bài: NBGCD2

Tiếp nối câu chuyện nhà hiền triết vĩ đại, sau khi các bạn chứng minh rằng nhà hiền triết luôn đúng, thì 1000 năm sau, các nhà khoa học đã phát hiện thêm rằng, lời dạy đó không chỉ đúng với 2 cá thể, mà còn đúng với một nhóm cá thể.

Cho một dãy số thể hiện đặc điểm của một nhóm cá thể, bạn hãy tìm ước chung lớn nhất của cả dãy số để chứng minh các nhà khoa học vẫn đúng nhé!

INPUT

- Dòng đầu tiên chứa số n ($1 \leq n \leq 10^5$) là độ dài của dãy.
- Dòng tiếp theo, chứa n số nguyên có giá trị tuyệt đối không quá 10^{18} cách nhau bởi dấu cách mô tả dãy số.

OUTPUT

In ra kết quả của bài toán.

INPUT	OUTPUT
3 12 16 18	2

4. QUÂN MÃ

Mã bài: NBKNIGHT

Cho một bàn cờ vua có kích thước $n * n$ (n hàng, n cột). Các hàng, các cột được đánh số từ 1 tới n , từ trên xuống dưới, từ trái sang phải. Trên bàn cờ chứa một con tướng và một con mã (vị trí ban đầu của con tướng và con mã không trùng nhau). Hỏi con mã có thể di chuyển để ăn được con tướng không, biết rằng con tướng không di chuyển, số lượng nước đi của con mã là không giới hạn.

INPUT

- Dòng đầu chứa số nguyên dương n ($1 < n \leq 15$).
- Dòng tiếp theo chứa bốn số lần lượt là tọa độ hàng và cột của con tướng và con mã trên bàn cờ, mỗi số cách nhau một dấu cách.

OUTPUT

In ra “YES” nếu con mã có thể ăn được con tướng, ngược lại thì in ra “NO”.

INPUT	OUTPUT
3 2 2 1 2	NO

5. DÂY NHỊ PHÂN

Mã bài: NBBIN

Cho số nguyên dương n . Hãy in ra số cách tạo ra một dãy nhị phân độ dài n sao cho dãy đó không tồn tại hai số 0 đứng cạnh nhau.

INPUT

Nhập số nguyên dương n ($0 < n \leq 20$).

OUTPUT

Đưa ra kết quả mà đề bài yêu cầu.

INPUT	OUTPUT	Giải thích
1	2	Dãy nhị phân thoả mãn là: “0”, “1”.
2	3	Dãy nhị phân thoả mãn là: “01”, “10”, “11”.

6. DIỆN TÍCH ĐA GIÁC

Mã bài: NBSQPOLY

Cho đa giác lồi n đỉnh $A_1A_2A_3\dots A_n$. Tính diện tích đa giác đó.

INPUT

- Dòng đầu chứa số nguyên dương n ($3 \leq n \leq 100$).
- n dòng sau, mỗi dòng chứa một cặp số thực x, y ($|x|, |y| \leq 1000$) lần lượt là tung độ và hoành độ của các đỉnh đa giác, cách nhau bởi một dấu cách.

Biết rằng các đỉnh của đa giác được liệt kê theo chiều kim đồng hồ.

OUTPUT

Kết quả bài toán lấy đến chữ số thập phân thứ 6.

INPUT	OUTPUT
4	1.500000
0 0	
0 1	
1 1	
2 0	

7. ĐÁNH SỐ TRANG

Mã bài: NBPAGE

Sau khi về tới nhà, bạn đã quyết định đọc sách để thanh lọc tâm hồn. Cuốn sách có tổng cộng n trang nhưng các trang sách lại chưa được đánh số. Bạn bắt đầu đánh số trang từ số 1 và tự hỏi, cần bao nhiêu chữ số để có thể đánh số trang cả cuốn sách?

INPUT

Nhập số nguyên n ($1 \leq n \leq 10^9$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT
13	17
10	11
5	5

8. DÃY SỐ #1**Mã bài: NBSEQ1**

Cho dãy A có n phần tử. Tính độ dài đoạn con dài nhất chứa các phần tử bằng nhau của dãy A .

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^6 , cách nhau bởi một dấu cách.

OUTPUT

In ra độ dài của đoạn con tìm được.

INPUT	OUTPUT
8 1 2 2 2 3 4 3 4	3

9. DÃY SỐ #2

Mã bài: NBSEQ2

Cho dãy A có n phần tử. Tìm độ dài đoạn con dài nhất chỉ chứa các phần tử dương của dãy A.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^6 , cách nhau bởi một dấu cách.

OUTPUT

In ra độ dài của đoạn con tìm được.

INPUT	OUTPUT
8 1 2 2 -2 3 -4 3 4	3

10. DÃY SỐ #3

Mã bài: NBSEQ3

Cho dãy A có n phần tử và một số nguyên dương k. Tìm độ dài đoạn con dài nhất chứa các phần tử đều chia hết cho k trong dãy A

INPUT

- Dòng 1: Chứa hai số nguyên dương n, k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^6$), cách nhau bởi một dấu cách.
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^6 , cách nhau bởi một dấu cách.

OUTPUT

In ra độ dài của đoạn con tìm được.

INPUT	OUTPUT
8 2 1 2 2 -2 6 -4 3 3	5

11.DÃY SỐ #4

Mã bài: NBSEQ4

Cho dãy A có n phần tử. Tìm độ dài đoạn con dài nhất chứa các phần tử tăng hoặc giảm dần trong dãy A.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^6 , cách nhau bởi một dấu cách.

OUTPUT

In ra độ dài của đoạn con tìm được.

INPUT	OUTPUT
8 1 2 2 -2 -2 2 3 5	4

12.DÃY SỐ #5

Mã bài: NBSEQ5

Cho dãy A có n phần tử. Tìm độ dài đoạn con dài nhất chứa các phần tử âm dương đan xen nhau trong dãy A.

INPUT

- Dòng 1: Chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng 2: Chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^6 , cách nhau bởi một dấu cách.

OUTPUT

In ra độ dài của đoạn con tìm được.

INPUT	OUTPUT
8 1 2 2 -2 3 -4 3 3	5

13.XOÁ SỐ #2

Mã bài: NBDELNUM2

Cho số nguyên dương S. Hãy tìm cách xoá đi k chữ số của số S để tạo thành số lớn nhất (không chứa số 0 ở đầu).

INPUT

- Dòng 1: Chứa số nguyên không âm k (k nhỏ hơn độ dài của số S).
- Dòng 2: Chứa số S có số lượng chữ số nhỏ hơn 100.

OUTPUT

Số lớn nhất tạo được.

INPUT	OUTPUT
3 82676113	87613

14.XOÁ SỐ #3

Mã bài: NBDELNUM3

Cho số nguyên dương S. Không thay đổi vị trí của các chữ số, hãy tìm cách xoá đi k chữ số của số S để tạo thành số bé nhất (không chứa số 0 ở đầu).

INPUT

- Dòng 1: Chứa số nguyên không âm k (k nhỏ hơn độ dài của số S).
- Dòng 2: Chứa số S có số lượng chữ số nhỏ hơn 100.

OUTPUT

Số bé nhất tạo được.

INPUT	OUTPUT
3 82676113	26113

15.COUNTING ZERO

Mã bài: NBLASTZERO

Cho số nguyên dương n. Hãy đếm số lượng chữ số 0 ở tận cùng của n!

INPUT

Nhập số nguyên dương n ($1 \leq n \leq 10^9$).

OUTPUT

Kết quả bài toán.

INPUT	OUTPUT	Giải thích
10	2	$10! = 3628800$ Số lượng số 0 tận cùng là 2.
16	3	

PHỤ LỤC : BỘ MÃ ASCII CƠ SỞ

Mã thập phân	Mã Hexa	Mã nhị phân	Ký tự
000	000	00000000	NUL
001	001	00000001	SOH
002	002	00000010	STX
003	003	00000011	RTX
004	004	00000100	EOT
005	005	00000101	ENQ
006	006	00000110	ACK
007	007	00000111	BEL
008	008	00001000	BS
009	009	00001001	HT
010	00A	00001010	LF
011	00B	00001011	VT
012	00C	00001100	FF
013	00D	00001101	CR
014	00E	00001110	SO
015	00F	00001111	SI
016	010	00010000	DLE
017	011	00010001	DC1
018	012	00010010	DC2
019	013	00010011	DC3
020	014	00010100	DC4
021	015	00010101	NAK
022	016	00010110	SYN
023	017	00010111	ETB

024	018	00011000	CAN
025	019	00011001	EM
026	01A	00011010	SUB
027	01B	00011011	ESC
028	01C	00011100	FS
029	01D	00011101	GS
030	01E	00011110	RS
031	01F	00011111	US
032	020	00100000	SP
033	021	00100001	!
034	022	00100010	“
035	023	00100011	#
036	024	00100100	\$
037	025	00100101	%
038	026	00100110	&
039	027	00100111	‘
040	028	00101000	(
041	029	00101001)
042	02A	00101010	*
043	02B	00101011	+
044	02C	00101100	,
045	02D	00101101	—
046	02E	00101110	.
047	02F	00101111	/
048	030	00110000	0
049	031	00110001	1
050	032	00110010	2

051	033	00110011	3
052	034	00110100	4
053	035	00110101	5
054	036	00110110	6
055	037	00110111	7
056	038	00111000	8
057	039	00111001	9
058	03A	00111010	:
059	03B	00111011	;
060	03C	00111100	<
061	03D	00111101	=
062	03E	00111110	>
063	03F	00111111	?
064	040	01000000	@
065	041	01000001	A
066	042	01000010	B
067	043	01000011	C
068	044	01000100	D
069	045	01000101	E
070	046	01000110	F
071	047	01000111	G
072	048	01001000	H
073	049	01001001	I
074	04A	01001010	J
075	04B	01001011	K
076	04C	01001100	L
077	04D	01001101	M

078	04E	01001110	N
079	04F	01001111	O
080	050	01010000	P
081	051	01010001	Q
082	052	01010010	R
083	053	01010011	S
084	054	01010100	T
085	055	01010101	U
086	056	01010110	V
087	057	01010111	W
088	058	01011000	X
089	059	01011001	Y
090	05A	01011010	Z
091	05B	01011011	[
092	05C	01011100	\
093	05D	01011101]
094	05E	01011110	^
095	05F	01011111	_
096	060	01100000	`
097	061	01100001	a
098	062	01100010	b
099	063	01100011	c
100	064	01100100	d
101	065	01100101	e
102	066	01100110	f
103	067	01100111	g
104	068	01101000	h

105	069	01101001	i
106	06A	01101010	j
107	06B	01101011	k
108	06C	01101100	l
109	06D	01101101	m
110	06E	01101110	n
111	06F	01101111	o
112	070	01110000	p
113	071	01110001	q
114	072	01110010	r
115	073	01110011	s
116	074	01110100	t
117	075	01110101	u
118	076	01110110	v
119	077	01110111	w
120	078	01111000	x
121	079	01111001	y
122	07A	01111010	z
123	07B	01111011	{
124	07C	01111100	
125	07D	01111101	}
126	07E	01111110	~
127	07F	01111111	DEL

MỤC LỤC

LỜI NÓI ĐẦU	5
KIẾN THỨC CHUNG.....	1
1. Kiểu dữ liệu (Data type)	1
2. Biến số (Variables)	2
3. Hằng số (Constant)	3
4. Phép toán (Operation).....	3
5. Biểu thức (Expression)	4
6. Phép gán (Assignment).....	5
7. Một số hàm số học chuẩn thông dụng	6
CHƯƠNG 1. NHẬP - XUẤT.....	7
1.1. Câu lệnh nhập - xuất.....	7
1.2. Ép kiểu dữ liệu trong lập trình C/C++	9
1.3. Kiểu ký tự và chuỗi ký tự	11
1.4. Bài tập thực hành	13
CHƯƠNG 2. CẤU TRÚC Rẽ NHÁNH.....	31
2.1. Câu lệnh if, if..else	31
2.2. Phép toán quan hệ, phép toán logic	32
2.3. Bài tập thực hành	33
CHƯƠNG 3. CẤU TRÚC LẶP.....	56
3.1. Vòng lặp for	56
3.2. Vòng lặp while, do...while.....	57
3.3. Bài tập thực hành	58
CHƯƠNG 4. XỬ LÝ CHUỖI KÝ TỰ.....	89

4.1. Một số hàm xử lý chuỗi thông dụng	89
4.2. Bài tập thực hành	91
CHƯƠNG 5. DỮ LIỆU KIỂU MẢNG	112
5.1. Mảng một chiều	112
5.2. Mảng hai chiều.....	114
5.3. Kiểu Vector.....	116
5.4. Bài tập thực hành	120
CHƯƠNG 6: ĐỆ QUY VÀ BÀI TẬP NÂNG CAO ..	160
6.1. Đệ quy.....	160
6.2. Bài tập thực hành	162
PHỤ LỤC : BỘ MÃ ASCII CƠ SỞ.....	173
MỤC LỤC.....	178



Trung tâm đào tạo lập trình CodeDream



Số 12 Vũ Trọng Khánh, P.Mộ Lao, Q.Hà Đông, TP.Hà Nội



0971 730 550



info@codedream.edu.vn



www.facebook.com/codedreamedu