

LUYỆN TẬP 2

**BÀI 1. SẮP XẾP ĐỔI CHỖ TRỰC TIẾP**

Hãy thực hiện thuật toán sắp xếp đổi chỗ trực tiếp trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán. **Dữ liệu vào:** Dòng 1 ghi số N (không quá 100). Dòng 2 ghi N số nguyên dương (không quá 100). **Kết quả:** Ghi ra màn hình từng bước thực hiện thuật toán. Mỗi bước trên một dòng, các số trong dãy cách nhau đúng một khoảng trống.

**Ví dụ:**

Input	Output
4 5 7 3 2	Buoc 1: 2 7 5 3 Buoc 2: 2 3 7 5 Buoc 3: 2 3 5 7

**BÀI 2. SẮP XẾP CHỌN**

Hãy thực hiện thuật toán sắp xếp chọn trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán.

**Dữ liệu vào:** Dòng 1 ghi số N (không quá 100). Dòng 2 ghi N số nguyên dương (không quá 100).

**Kết quả:** Ghi ra màn hình từng bước thực hiện thuật toán. Mỗi bước trên một dòng, các số trong dãy cách nhau đúng một khoảng trống.

**Ví dụ:**

Input	Output
4 5 7 3 2	Buoc 1: 2 7 3 5 Buoc 2: 2 3 7 5 Buoc 3: 2 3 5 7

**BÀI 3. SẮP XẾP CHÈN**

Hãy thực hiện thuật toán sắp xếp chèn trên dãy N số nguyên. Ghi ra các bước thực hiện thuật toán.

**Dữ liệu vào:** Dòng 1 ghi số N (không quá 100). Dòng 2 ghi N số nguyên dương (không quá 100).

**Kết quả:** Ghi ra màn hình từng bước thực hiện thuật toán. Mỗi bước trên một dòng, các số trong dãy cách nhau đúng một khoảng trống.

**Ví dụ:**

Input	Output
4 5 7 3 2	Buoc 0: 5 Buoc 1: 5 7 Buoc 2: 3 5 7 Buoc 3: 2 3 5 7

#### BÀI 4. SẮP XẾP NỔI BỌT

Hãy thực hiện thuật toán sắp xếp nổi bọt trên dãy  $N$  số nguyên. Ghi ra các bước thực hiện thuật toán.

**Dữ liệu vào:** Dòng 1 ghi số  $N$  (không quá 100). Dòng 2 ghi  $N$  số nguyên dương (không quá 100).

**Kết quả:** Ghi ra màn hình từng bước thực hiện thuật toán. Mỗi bước trên một dòng, các số trong dãy cách nhau đúng một khoảng trống.

**Ví dụ:**

Input	Output
4	Buoc 1: 3 2 5 7
5 3 2 7	Buoc 2: 2 3 5 7

#### BÀI 5. SẮP XẾP NHANH

Cho dãy số  $A[]$  gồm có  $N$  phần tử. Nhiệm vụ của bạn là sắp xếp dãy số theo thứ tự tăng dần.

**Dữ liệu vào:** Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 10$ ).

Mỗi test bắt đầu bằng số nguyên  $N$ , ( $N \leq 100\,000$ )

Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $0 \leq A[i] \leq 10^6$ ).

**Kết quả:**

In ra các phần tử của dãy số sau khi được sắp xếp.

**Ví dụ:**

Input	Output
1	1 2 3 4 5
5	
2 4 1 3 5	

#### BÀI 6. SẮP XẾP KHÔNG NHANH

Cho dãy số  $A[]$  gồm có  $N$  phần tử. Nhiệm vụ của bạn là sắp xếp dãy số theo thứ tự tăng dần.

Bộ test được xây dựng để bạn không thể đúng nếu sử dụng các phiên bản của sắp xếp nhanh (Quick Sort).

**Dữ liệu vào:**

Mỗi test bắt đầu bằng số nguyên  $N$ , ( $N \leq 100\,000$ )

Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $0 \leq A[i] \leq 10^{18}$ ).

**Kết quả:**

In ra các phần tử của dãy số sau khi được sắp xếp.

**Ví dụ:**

Input	Output
5 2 4 1 3 5	1 2 3 4 5

## BÀI 7. SẮP XẾP LẠI DẠY CON

Cho dãy số  $A[]$  gồm có  $N$  phần tử. Nhiệm vụ của bạn là hãy tìm một dãy con liên tiếp dài nhất, sao cho sau khi sắp xếp lại dãy con này, dãy số  $A[]$  sẽ là một dãy số được sắp xếp tăng dần.

### Dữ liệu vào:

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 10$ ).

Mỗi test bắt đầu bằng số nguyên  $N$  ( $N \leq 100\,000$ ).

Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $0 \leq A[i] \leq 10^6$ ).

### Kết quả:

Với mỗi test in ra độ dài dãy con tìm được trên một dòng. Nếu dãy đã cho đã được sắp xếp, in ra “YES”.

### Ví dụ:

Input	Output
3 9 0 1 15 25 6 7 30 40 50 5 1 2 4 3 5 4 1 2 3 4	4 2 YES

Giải thích test 1: Cần sắp xếp lại dãy con 15, 25, 6, 7.

## BÀI 8. BRT

Thành phố  $X$  có  $N$  thị trấn trên trục đường chính. Tọa độ của các thị trấn lần lượt là  $a[1]$ ,  $a[2]$ , ...,  $a[N]$ , các tọa độ này là phân biệt, không có 2 tọa độ nào trùng nhau.

Chính quyền thành phố muốn xây dựng một tuyến buýt nhanh BRT để kết nối 2 thị trấn gần nhau nhất với nhau.

Bạn hãy tính thử xem chiều dài của tuyến buýt này bằng bao nhiêu? Và có bao nhiêu cặp thị trấn có tiềm năng giống nhau để xây dựng tuyến BRT này.

### Dữ liệu vào:

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 10$ ).

Mỗi test bắt đầu bằng số nguyên  $N$  ( $N \leq 100\,000$ ).

Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $-10^9 \leq A[i] \leq 10^9$ ).

**Kết quả:**

Với mỗi test in ra 2 số nguyên C và D, lần lượt là khoảng cách ngắn nhất giữa 2 thị trấn, và số lượng cặp thị trấn có cùng khoảng cách ngắn nhất này.

**Ví dụ:**

Input	Output
2	2 1
4	2 2
6 -3 0 4	
3	
-2 0 2	

Giải thích test 2: Cặp thị trấn (1, 2) và (2, 3) có cùng khoảng cách.

**BÀI 9. TÌM KIẾM NHỊ PHÂN**

Cho dãy số  $A[]$  gồm có  $N$  phần tử đã được sắp xếp tăng dần và số  $K$ .

Nhiệm vụ của bạn là kiểm tra xem số  $K$  có xuất hiện trong dãy số hay không. Nếu có hãy in ra vị trí trong dãy  $A[]$ , nếu không in ra “NO”.

**Input:**

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 10$ ).

Mỗi test bắt đầu bằng số nguyên  $N$  và  $K$  ( $N \leq 100\,000$ ,  $0 \leq K \leq 10^6$ ).

Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$  ( $0 \leq A[i] \leq 10^6$ ), các phần tử là riêng biệt.

**Output:** Với mỗi test in ra trên một dòng đáp án tìm được.

**Ví dụ:**

Input	Output
2	3
5 3	NO
1 2 3 4 5	
6 5	
0 1 2 3 9 10	

**BÀI 10. MUA CÀ PHÊ**

Tí rất thích uống cà phê đá, một thứ mà rất nhiều lập trình viên khác cũng yêu thích. Có  $N$  cửa hàng bán cà phê ở gần chỗ làm việc của Tí, giá bán cho mỗi cốc cà phê tại đây lần lượt là  $X[i]$ .

Mỗi ngày, Tí có thể sử dụng số tiền bằng  $Y[i]$  để mua cà phê. Các bạn hãy xác định xem Tí có thể lựa chọn bao nhiêu cửa hàng để mua được cà phê cho mình?

**Input:**

Dòng đầu tiên số nguyên  $N$  ( $1 \leq N \leq 100\,000$ ).

Dòng tiếp theo gồm  $N$  số nguyên  $X[i]$  ( $0 \leq X[i] \leq 10^6$ ), là giá bán cà phê của cửa hàng thứ  $i$ .

Dòng tiếp gồm số nguyên  $Q$ , là số ngày mà Tí muốn uống cà phê.

$Q$  dòng cuối, mỗi dòng gồm một số nguyên  $Y[i]$  ( $0 \leq Y[i] \leq 10^6$ ), là số tiền mà Tí có thể dùng trong ngày thứ  $i$ .

### Output:

Với mỗi ngày, in ra số cửa hàng mà Tí có thể mua được cà phê.

### Ví dụ:

Input:	Output
5	0
3 10 8 6 11	4
4	1
1	5
10	
3	
11	

Giải thích test 1: Ngày 1, Tí không đủ tiền để mua cà phê.

## BÀI 11. KHUYẾN MẠI

Một cửa hàng thời trang đang thực hiện chương trình khuyến mại giảm giá. Ban đầu, giá của sản phẩm  $i$  là  $a[i]$ , khi đến tuần giảm giá, giá của chúng giảm xuống còn  $b[i]$ . Tuy nhiên, chủ cửa hàng rất khôn, nhằm đánh lừa khách hàng, mỗi số sản phẩm giá tăng lên chứ không hề giảm xuống.

Nhận biết được quy luật này, Tí mặc dù cần phải mua tổng cộng  $N$  sản phẩm, nhưng cậu quyết định mua  $K$  sản phẩm trước đợt khuyến mại, và  $N-K$  sản phẩm trong đợt khuyến mại.

Giả sử rằng Tí chọn tối ưu được  $K$  sản phẩm ban đầu, các bạn hãy tính xem số tiền ít nhất Tí cần bỏ ra để mua đủ  $N$  sản phẩm là bao nhiêu?

### Input:

Mỗi test bắt đầu bằng số nguyên  $N$  và  $K$  ( $1 \leq N, K \leq 100\,000$ ).

Dòng thứ hai gồm  $N$  số nguyên  $a[i]$ , giá sản phẩm thứ  $i$  mà trước đợt giảm giá.

Dòng cuối gồm  $N$  số nguyên  $b[i]$ , là giá của sản phẩm sau khi giảm giá.

( $1 \leq a[i], b[i] \leq 10^4$ ).

### Output:

In ra một số nguyên là đáp án của bài toán.

### Ví dụ:

Test 1	Test 2
Input : 3 1 5 4 6 3 1 5	Input : 5 3 3 4 7 10 3 4 5 5 12 5

Output: 10	Output: 25
---------------	---------------

Giải thích test 1: Tí mua sản phẩm 3 trước khi giảm giá, và sản phẩm 1, 2 trong thời gian khuyến mại.

Giải thích test 2: Tí mua sản phẩm 1, 2, 4, 5 trước, tới đợt khuyến mại thì mua sản phẩm 3.

## BÀI 12. LŨY THỪA

Cho số nguyên dương  $N$  và  $K$ . Hãy tính  $N^K$  modulo  $10^9+7$ .

### Input:

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 20$ ).

Mỗi test gồm 1 số nguyên  $N$  và  $K$  ( $1 \leq N \leq 1000$ ,  $1 \leq K \leq 10^9$ ).

### Output:

Với mỗi test, in ra đáp án trên một dòng.

### Ví dụ:

Input:	Output
2	8
2 3	16
4 2	

## BÀI 13. GẤP ĐÔI DÃY SỐ

Một dãy số tự nhiên bắt đầu bởi con số 1 và được thực hiện  $N-1$  phép biến đổi “gấp đôi” dãy số như sau:

Với dãy số  $A$  hiện tại, dãy số mới có dạng  $A, x, A$  trong đó  $x$  là số tự nhiên bé nhất chưa xuất hiện trong  $A$ .

Ví dụ với 2 bước biến đổi, ta có  $[1] \rightarrow [1\ 2\ 1] \rightarrow [1\ 2\ 1\ 3\ 1\ 2\ 1]$ .

Các bạn hãy xác định số thứ  $K$  trong dãy số cuối cùng là bao nhiêu?

### Input:

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 20$ ).

Mỗi test gồm số nguyên dương  $N$  và  $K$  ( $1 \leq N \leq 50$ ,  $1 \leq K \leq 2^N - 1$ ).

### Output:

Với mỗi test, in ra đáp án trên một dòng.

### Test ví dụ:

Input:	Output
2	2
3 2	4
4 8	

Giải thích test 1: Dãy số thu được là  $[1, 2, 1, 3, 1, 2, 1]$ .

Giải thích test 2: Dãy số thu được là  $[1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1]$ .

## BÀI 14. ĐẾM DÃY

Cho số nguyên dương  $n$ . Hãy cho biết có bao nhiêu dãy số nguyên dương có tổng các phần tử trong dãy bằng  $n$ .

**Dữ liệu vào:** dòng đầu tiên chứa số nguyên  $T$  là số bộ dữ liệu, mỗi bộ dữ liệu ghi một số nguyên dương  $n$  duy nhất không qua  $10^{18}$ .

**Kết quả:** Mỗi bộ dữ liệu ghi ra một số nguyên duy nhất là số dư của kết quả tìm được khi chia cho 123456789.

**Ví dụ:**

Input	Output
1 3	4

### BÀI 15. DÃY XÂU FIBONACI

Một dãy xâu ký tự  $G$  chỉ bao gồm các chữ cái A và B được gọi là dãy xâu Fibonacci nếu thỏa mãn tính chất:  $G(1) = A$ ;  $G(2) = B$ ;  $G(n) = G(n-2) + G(n-1)$ . Với phép cộng (+) là phép nối hai xâu với nhau. Bài toán đặt ra là tìm ký tự ở vị trí thứ  $i$  (tính từ 1) của xâu Fibonacci thứ  $n$ .

**Dữ liệu vào:** Dòng 1 ghi số bộ test. Mỗi bộ test ghi trên một dòng 2 số nguyên  $N$  và  $i$  ( $1 < N < 93$ ). Số  $i$  đảm bảo trong phạm vi của xâu  $G(N)$  và không quá 18 chữ số. **Kết quả:** Ghi ra màn hình kết quả tương ứng với từng bộ test.

Input	Output
2 6 4 8 19	A B

### BÀI 16. SỐ FIBONACCI THỨ $N$

Dãy số Fibonacci được xác định bằng công thức như sau:

$$F[0] = 0, F[1] = 1;$$

$$F[n] = F[n-1] + F[n-2] \text{ với mọi } n \geq 2.$$

Các phần tử đầu tiên của dãy số là 0, 1, 1, 2, 3, 5, 8, ...

Nhiệm vụ của bạn là hãy xác định số Fibonacci thứ  $n$ . Do đáp số có thể rất lớn, in ra kết quả theo modulo  $10^9+7$ .

**Input:**

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 1000$ ).

Mỗi test bắt gồm một số nguyên  $N$  ( $1 \leq N \leq 10^9$ ).

**Output:**

Với mỗi test, in ra đáp án trên một dòng.

**Ví dụ:**

Input:	Output
3 2 6 20	1 8 6765

### BÀI 17. LŨY THỪA MA TRẬN

Cho ma trận vuông  $A$  kích thước  $N \times N$ . Nhiệm vụ của bạn là hãy tính ma trận  $X = A^K$  với  $K$  là số nguyên cho trước. Đáp số có thể rất lớn, hãy in ra kết quả theo modulo  $10^9+7$ .

**Input:**

Dòng đầu tiên là số lượng bộ test T ( $T \leq 100$ ).

Mỗi test bắt gồm một số nguyên N và K ( $1 \leq N \leq 10$ ,  $1 \leq K \leq 10^9$ ) là kích thước của ma trận và số mũ.

**Output:**

Với mỗi test, in ra kết quả của ma trận X.

**Ví dụ:**

Input:	Output
2	8 5
2 5	5 3
1 1	597240088 35500972 473761863
1 0	781257150 154135232 527013321
3 1000000000	965274212 272769492 580264779
1 2 3	
4 5 6	
7 8 9	

**BÀI 18. NGĂN XẾP**

Cho một ngăn xếp các số nguyên. Các thao tác được mô tả trong file văn bản gồm 3 lệnh: push, pop và show. Trong đó thao tác push kèm theo một giá trị cần thêm (không quá 1000). Hãy viết chương trình ghi ra kết quả của các lệnh show.

**Dữ liệu vào:** Gồm nhiều dòng, mỗi dòng chứa một lệnh push, pop hoặc show. Input đảm bảo số lượng phần tử trong stack khi nhiều nhất cũng không vượt quá 200.

**Kết quả:** Ghi ra màn hình các phần tử đang có trong stack theo thứ tự lưu trữ mỗi khi gặp lệnh show. Các số viết cách nhau đúng một khoảng trống. Nếu trong stack không còn gì thì in ra dòng “empty”

**Ví dụ:**

Input	Output
push 3	3 5
push 5	3 5 7
show	3
push 7	
show	
pop	
pop	
show	

**BÀI 19. BIỂU THỨC HẬU TỔ**

Hãy sử dụng ngăn xếp để thực hiện việc chuyển các biểu thức trung tố sang hậu tố

**Dữ liệu vào:** Dòng 1 ghi số N (không quá 20) là số biểu thức trung tố (đúng khuôn dạng) chỉ bao gồm các phép cộng, trừ, nhân, chia, các chữ cái thường từ a đến z và các dấu ngoặc đơn. N dòng tiếp theo mỗi dòng ghi một biểu thức.

**Kết quả:** Ghi ra màn hình các biểu thức hậu tố kết quả.

**Ví dụ:**

Input	Output
1	ab+cd+*
((a+b) * (c+d) )	



## BÀI 20. KIỂM TRA DẪY NGOẶC ĐÚNG

Cho một chuỗi chỉ gồm các ký tự ‘(’, ‘)’, ‘[’, ‘]’, ‘{’, ‘}’. Một dãy ngoặc đúng được định nghĩa như sau:

- Chuỗi rỗng là 1 dãy ngoặc đúng.
- Nếu A là 1 dãy ngoặc đúng thì (A), [A], {A} là 1 dãy ngoặc đúng.
- Nếu A và B là 2 dãy ngoặc đúng thì AB là 1 dãy ngoặc đúng.

Cho một chuỗi S. Nhiệm vụ của bạn là xác định chuỗi S có là dãy ngoặc đúng hay không?

### Input:

Dòng đầu tiên là số lượng bộ test T ( $T \leq 20$ ).

Mỗi test gồm 1 chuỗi S có độ dài không vượt quá 100 000.

### Output:

Với mỗi test, in ra “YES” nếu như S là dãy ngoặc đúng, in ra “NO” trong trường hợp ngược lại.

### Ví dụ:

Input:	Output
2	YES
[ ( ) ] { } { [ ( ) ( ) ] ( ) }	NO
[ ( ] )	

## BÀI 21. PHẦN TỬ BÊN PHẢI ĐẦU TIÊN LỚN HƠN

Cho dãy số A[] gồm N phần tử. Với mỗi A[i], bạn cần tìm phần tử bên phải đầu tiên lớn hơn nó. Nếu không tồn tại, in ra -1.

### Input:

Dòng đầu tiên là số lượng bộ test T ( $T \leq 20$ ).

Mỗi test bắt đầu bởi số nguyên N ( $1 \leq N \leq 100000$ ).

Dòng tiếp theo gồm N số nguyên A[i] ( $0 \leq A[i] \leq 10^9$ ).

### Output:

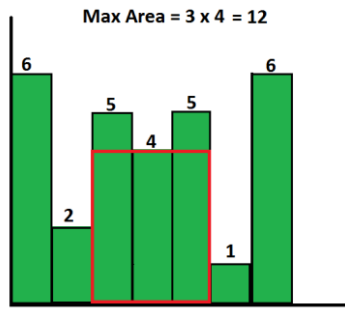
Với mỗi test, in ra trên một dòng N số R[i], với R[i] là giá trị phần tử đầu tiên lớn hơn A[i].

### Ví dụ

Input	Output
3	5 25 25 -1
4	-1 -1 -1
4 5 2 25	5 5 -1 -1
3	
2 2 2	
4	
4 4 5 5	

## BÀI 22. HÌNH CHỮ NHẬT LỚN NHẤT

Cho N cột, mỗi cột có chiều cao bằng H[i]. Bạn hãy tìm hình chữ nhật lớn nhất bị che phủ bởi các cột?



**Input:**

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 20$ ).

Mỗi test bắt đầu bởi số nguyên  $N$  ( $N \leq 100\,000$ ).

Dòng tiếp theo gồm  $N$  số nguyên  $H[i]$  ( $1 \leq H[i] \leq 10^9$ ).

**Output:**

Với mỗi test, in ra diện tích hình chữ nhật lớn nhất tìm được.

**Ví dụ:**

Input	Output
2	12
7	6
6 2 5 4 5 1 6	
3	
2 2 2	

## BÀI 23. CẤU TRÚC DỮ LIỆU HÀNG ĐỢI

Ban đầu cho một queue rỗng. Bạn cần thực hiện các truy vấn sau:

1. Trả về kích thước của queue
2. Kiểm tra xem queue có rỗng không, nếu có in ra “YES”, nếu không in ra “NO”.
3. Cho một số nguyên và đẩy số nguyên này vào cuối queue.
4. Loại bỏ phần tử ở đầu queue nếu queue không rỗng, nếu rỗng không cần thực hiện.
5. Trả về phần tử ở đầu queue, nếu queue rỗng in ra -1.
6. Trả về phần tử ở cuối queue, nếu queue rỗng in ra -1.

**Dữ liệu vào**

Dòng đầu tiên chứa số nguyên  $T$  là số bộ dữ liệu, mỗi bộ dữ liệu theo dạng sau.

Dòng đầu tiên chứa số nguyên  $n$  - lượng truy vấn ( $1 \leq n \leq 1000$ )

$N$  dòng tiếp theo, mỗi dòng sẽ ghi loại truy vấn như trên, với truy vấn loại 3 sẽ có thêm một số nguyên, không quá  $10^6$ .

**Kết quả:** In ra kết quả của các truy vấn..

**Ví dụ:**

Input	Output
1	1
1 4	3
3 1	5
3 2	2

3	3
5	
6	
4	
4	
4	
4	
4	
3	5
3	6
5	
1	

## BÀI 24. ĐƯỜNG NGUYÊN TỐ

Cho hai số nguyên tố khác nhau có bốn chữ số. Người ta cho rằng hoàn toàn có thể biến đổi từ số này thành số kia sau một số bước theo quy tắc: Tại mỗi bước ta chỉ thay đổi một chữ số trong số trước đó sao cho số tạo được trong mỗi bước đều là một số nguyên tố có bốn chữ số. Một cách biến đổi như vậy gọi là một “đường nguyên tố”.

Bài toán đặt ra là với một cặp số nguyên tố đầu vào, hãy tính ra số bước của đường nguyên tố ngắn nhất. Giả sử đầu vào là hai số 1033 và 8179 thì đường nguyên tố ngắn nhất sẽ có độ dài là 6 với các bước chuyển là:

1033

1733

3733

3739

3779

8779

8179

**Dữ liệu vào:** Dòng đầu tiên ghi số bộ test, không lớn hơn 100. Mỗi bộ test viết trên một dòng bao gồm hai số nguyên tố có 4 chữ số..

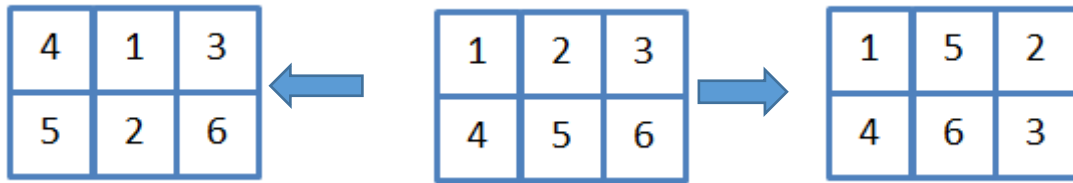
**Kết quả:** Với mỗi bộ test, in ra màn hình trên một dòng số bước của đường nguyên tố ngắn nhất.

**Ví dụ:**

Input	Output
3	6
1033 8179	7
1373 8017	0
1033 1033	

## BÀI 25. QUAY HÌNH VUÔNG

Có một chiếc bảng hình chữ nhật với 6 miếng ghép, trên mỗi miếng ghép được điền một số nguyên trong khoảng từ 1 đến 6. Tại mỗi bước, chọn một hình vuông (bên trái hoặc bên phải), rồi quay theo chiều kim đồng hồ.



**Yêu cầu:** Cho một trạng thái của bảng, hãy tính số phép biến đổi ít nhất để đưa bảng đến trạng thái đích.

**Dữ liệu vào:**

Dòng đầu tiên chứa 6 số là trạng thái bảng ban đầu (thứ tự từ trái qua phải, dòng 1 tới dòng 2).

Dòng thứ hai chứa 6 số là trạng thái bảng đích (thứ tự từ trái qua phải, dòng 1 tới dòng 2).

**Kết quả:**

In ra một số nguyên là đáp số của bài toán.

**Ví dụ:**

Input	Output
1 2 3 4 5 6 4 1 2 6 5 3	2

## BÀI 26. DI CHUYỂN

Cho một bảng kích thước  $N \times N$ , trong đó có các ô trống '.' và vật cản 'X'. Các hàng và các cột được đánh số từ 0.

Mỗi bước di chuyển, bạn có thể đi từ ô  $(x, y)$  tới ô  $(u, v)$  nếu như 2 ô này nằm trên cùng một hàng hoặc một cột, và không có vật cản nào ở giữa.

Cho điểm xuất phát và điểm đích. Bạn hãy tính số bước di chuyển ít nhất?

**Dữ liệu vào:**

Dòng đầu tiên là số nguyên dương  $N$  ( $1 \leq N \leq 100$ ).

$N$  dòng tiếp theo, mỗi dòng gồm  $N$  kí tự mô tả bảng.

Cuối cùng là 4 số nguyên  $a, b, c, d$  với  $(a, b)$  là tọa độ điểm xuất phát,  $(c, d)$  là tọa độ đích. Dữ liệu đảm bảo hai vị trí này không phải là ô cấm.

**Kết quả:**

In ra một số nguyên là đáp số của bài toán.

**Ví dụ:**

Input	Output
3 .X. .X. ... 0 0 0 2	3

## BÀI 27. SỐ BDN

Ta gọi số nguyên dương  $K$  là một số BDN nếu các chữ số trong  $K$  chỉ bao gồm các 0 hoặc 1 có nghĩa. Ví dụ số  $K = 1, 10, 101$ . Cho số tự nhiên  $N$  ( $N < 2^{63}$ ). Hãy cho biết có bao nhiêu số BDN không lớn hơn  $N$ . Ví dụ  $N=100$  ta có 4 số BDN bao gồm các số: 1, 10, 11, 100.

Input	Output
3	2
10	4
100	7
200	

**Dữ liệu vào:** Dòng đầu tiên ghi lại số tự nhiên  $T$  ( $T \leq 100$ ) là số lượng bộ test.

$T$  dòng kế tiếp mỗi dòng là một số tự nhiên  $N$  ( $1 \leq N \leq 2^{63}$ ).

**Kết quả:** Với mỗi số  $N$  in ra kết quả trên một dòng là số các số BDN không lớn hơn  $N$ .

## BÀI 28. SỐ BDN NHỎ NHẤT CHIA HẾT CHO $N$

Ta gọi số nguyên dương  $K$  là một số BDN nếu các chữ số trong  $K$  chỉ bao gồm các 0 hoặc 1 có nghĩa. Ví dụ số  $K = 101$  là số BDN,  $K = 102$  không phải là số BDN. Cho số tự nhiên  $N$  ( $N < 1000$ ), hãy tìm số BDN nhỏ nhất của  $N$ .

### Input

Dòng đầu tiên gồm một số nguyên  $T$  ( $1 \leq T \leq 100$ ) là số lượng bộ test.

$T$  dòng tiếp theo gồm  $T$  bộ test, mỗi bộ test gồm 1 dòng là một số nguyên  $N$  ( $1 \leq N \leq 1000$ ).

### Output

Với mỗi bộ test in ra trên một dòng là số BDN nhỏ nhất chia hết cho  $N$ . Kết quả không đảm bảo nhỏ hơn  $2^{63}$ .

### Ví dụ

Input	Output
3	10
2	11100
12	11101
17	

## BÀI 29. BIẾN ĐỔI S - T

Cho hai số nguyên dương  $S$  và  $T$  ( $S, T \leq 10^4$ ) và hai thao tác (a), (b) dưới đây:

- Thao tác (a): Trừ  $S$  đi 1.
- Thao tác (b): Nhân  $S$  với 2.

Hãy dịch chuyển  $S$  thành  $T$  sao cho số lần thực hiện các thao tác (a), (b) là ít nhất. Ví dụ với  $S = 2, T = 5$  thì số các bước ít nhất để dịch chuyển  $S$  thành  $T$  thông qua 4 thao tác sau:

- Thao tác (a):  $2 \times 2 = 4$
- Thao tác (b):  $4 - 1 = 3$
- Thao tác (a):  $3 \times 2 = 6$
- Thao tác (b):  $6 - 1 = 5$

### Input

Dòng đầu tiên gồm một số nguyên  $T$  ( $1 \leq T \leq 100$ ) là số lượng bộ test.

$T$  dòng tiếp theo, mỗi dòng gồm 2 số  $S, T$  ( $S, T \leq 10^4$ ).

### Output

Với mỗi bộ test, in ra số thao tác ít nhất để chuyển từ số  $S$  về số  $T$ .

**Ví dụ**

Input	Output
3	4
2 5	4
3 7	3
7 4	

**BÀI 30. BIẾN ĐỔI VỀ 1**

Cho số tự nhiên  $N$  ( $N \leq 10^7$ ) và hai phép biến đổi (a), (b) dưới đây.

- Thao tác (a): Trừ  $N$  đi 1. Ví dụ  $N=17$ , thao tác (a) biến đổi  $N = 16$ .
- Thao tác (b):  $N = \max(u, v)$  nếu  $u \times v = N$  ( $u, v > 1$ ). Ví dụ  $N=16$ , thao tác (b) có thể biến đổi  $N = \max(2, 8) = 8$  hoặc  $N = \max(4, 4) = 4$ .

Chỉ được phép sử dụng hai thao tác (a) hoặc (b), hãy biến đổi  $N$  thành 1 sao số các thao tác (a), (b) được thực hiện ít nhất. Ví dụ với  $N=17$ , số các phép (a), (b) nhỏ nhất biến đổi  $N$  thành 1 là 4 bước như sau:

- Thao tác (a):  $N = N - 1 = 16$
- Thao tác (b):  $N = \max(4, 4) = 4$
- Thao tác (b):  $N = \max(2, 2) = 2$
- Thao tác (a):  $N = N - 1 = 1$

**Input**

Dòng đầu tiên gồm một số nguyên  $T$  ( $1 \leq T \leq 100$ ) là số lượng bộ test.

$T$  dòng tiếp theo, mỗi dòng gồm một số nguyên  $N$  ( $1 \leq N \leq 10^7$ ).

**Output**

Với mỗi bộ test, in ra trên một dòng số phép biến đổi ít nhất tìm được.

**Ví dụ**

Input	Output
3	4
17	5
50	5
100	