

Bài tập C++ Code PTIT

1: LIỆT KÊ SỐ FIBONACCI

LIỆT KÊ SỐ FIBONACCI

Bài làm tốt nhất

Đây số Fibonacci được định nghĩa theo công thức như sau:

- $F_1 = 1$
- $F_2 = 1$
- $F_n = F_{n-1} + F_{n-2}$ với $n > 2$

Cho hai số nguyên dương a và b ($1 \leq a < b < 93$). Viết chương trình liệt kê các số Fibonacci từ số thứ a đến số thứ b.

Input

Dòng đầu ghi số bộ test, không quá 10.

Mỗi bộ test viết trên một dòng hai số a và b.

Output

Ghi ra kết quả của mỗi test trên một dòng, mỗi số cách nhau một khoảng trắng.

Ví dụ

Input	Output
1	1 1 2 3 5 8 13 21 34 55
1 10	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 200000 Kb

2 Hello World

Hello World

Bài làm tốt nhất

Viết chương trình in ra màn hình dòng chữ:

Hello PTIT.

Input

Không có dữ liệu vào

Output

Hello PTIT.

3 TÍNH TỔNG 1 ĐẾN N

TÍNH TỔNG 1 ĐẾN N

Bài làm tốt nhất

Cho số nguyên dương N.

Hãy tính $S = 1 + 2 + \dots + N$

Dữ liệu vào:

- Dòng đầu ghi số bộ test, không quá 10
- Mỗi dòng ghi một số nguyên dương N, không quá 10^9

Kết quả:

Với mỗi test, ghi kết quả trên một dòng.

Ví dụ:

Input	Output
2	55
10	210
20	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

4 CHỮ HOA – CHỮ THƯỜNG

Cho một chữ cái bất kỳ. Hãy kiểm tra xem đó là chữ hoa hay chữ thường. Nếu là chữ thường thì in ra chữ hoa, nếu là chữ hoa thì in ra chữ thường tương ứng.

Input

Dòng đầu ghi số bộ test. Mỗi bộ test có duy nhất một chữ cái.

Output

Với mỗi bộ test, ghi ra kết quả trên một dòng.

Ví dụ

Input	Output
2	b
B	Q
q	

5 TÍNH TỔNG PHÂN THỨC

Viết chương trình tính tổng $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{N}$

Input

Dữ liệu vào chỉ có 1 dòng ghi số N không quá 6 chữ số.

Output

Kết quả ghi chính xác với **4 số phần thập phân**.

Ví dụ

Input	Output
2	1.5000

Chú ý: Giới hạn thời gian chạy là 5 giây.

Giới hạn thời gian: 5s

Giới hạn bộ nhớ: 65536 Kb

6 TÍNH TỔNG GIAI THỦA

Viết chương trình tính tổng $S = 1 + 1.2 + 1.2.3 + \dots + 1.2.3\dots.N$

Input

Dữ liệu vào chỉ có 1 dòng ghi số N không quá 20.

Output

Kết quả ghi trên một dòng.

Ví dụ

Input	Output
3	9

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

7 SỐ LỘC PHÁT

Một số được gọi là “lộc phát” nếu chỉ có các chữ số 0,6,8. Nhập vào một số nguyên dương không quá 9 chữ số, hãy kiểm tra xem đó có phải số lộc phát hay không.

Input

Dòng đầu là số bộ test, mỗi dòng tiếp theo ghi một số cần kiểm tra.

Output

Nếu đúng in ra YES, nếu sai in ra NO.

Ví dụ

Input	Output
3	NO
181	YES
88888	YES
608868	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

8 SỐ THUẬN NGHỊCH

Một số nguyên dương được gọi là số thuận nghịch nếu viết theo chiều ngược lại vẫn có cùng giá trị với số ban đầu. Ví dụ số 121 là số thuận nghịch, số 123 không phải số thuận nghịch.

Viết chương trình kiểm tra số thuận nghịch

Input

Dòng đầu ghi số bộ test, mỗi bộ test ghi một số nguyên dương không quá 18 chữ số

Output

Với mỗi bộ test, nếu đúng ghi ra YES, nếu sai ghi ra NO.

Ví dụ

Input	Output
2	NO
123456789012345	YES
12345678987654321	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

9 CHẤM ĐIỂM TRẮC NGHIỆM

Trong những năm gần đây, hình thức thi trắc nghiệm được sử dụng phổ biến trong kỳ thi Trung học phổ thông quốc gia. Ưu điểm của hình thức thi trắc nghiệm là có thể chấm tự động hoàn toàn mà không cần sự can thiệp của con người, điều này đảm bảo tính khách quan và chính xác. Giả sử có 2 mã đề thi 101 và 102 có các đáp án được cho trong bảng dưới đây:

Câu	Đề 101	Đề 102
1	A	A
2	B	C
3	B	C
4	A	A
5	D	B
6	C	C
7	C	D
8	A	D
9	B	B
10	D	B
11	C	C
12	C	D
13	A	D
14	B	B
15	D	B

Nhiệm vụ của bạn là chấm điểm cho bài thi của các thí sinh trong một phòng thi. Điểm thi được chấm trên thang điểm 10, làm tròn đến 2 chữ số phần thập phân.

Input:

Đòng đầu tiên đưa vào số bài thi T.

Các dòng tiếp theo mỗi đưa vào mã đề thi và các đáp án làm bài của thí sinh cách nhau bởi một khoảng trắng. Giả sử tất cả các đáp án đều được điền đầy đủ.

T thỏa mãn điều kiện $0 < T \leq 1000$

Output:

Điểm thi được làm tròn đến hai chữ số phần thập phân.

Input	Output
1	10.00
101 A B B A D C C A B D C C A B D	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

10 CÂN BẰNG CHẴN LẺ

CÂN BẰNG CHẴN LẺ

Bài làm tốt nhất

Một số được gọi là “cân bằng chẵn lẻ” nếu số chữ số chẵn và số chữ số lẻ là bằng nhau. Tất nhiên khi đó số chữ số của nó phải là chẵn.

Cho số N là một số chẵn ($1 < N < 7$). Hãy liệt kê các số cân bằng chẵn lẻ có N chữ số. Mỗi dòng ghi ra 10 số thỏa mãn.

Input

Chi có duy nhất số N (chẵn)

Output

Ghi ra các số cân bằng chẵn lẻ có N chữ số theo thứ tự từ nhỏ đến lớn. Mỗi dòng ghi 10 số.

Ví dụ

Input	Output
2	10 12 14 16 18 21 23 25 27 29 30 32 34 36 38 41 43 45 47 49 50 52 54 56 58 61 63 65 67 69 70 72 74 76 78 81 83 85 87 89 90 92 94 96 98

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

11 MÃ SỐ QUỐC GIA

Trong mã hàng hóa người ta thường ghi kèm theo mã số quốc gia sản xuất. Nếu sản xuất tại Việt Nam thì mã tương ứng là 084. Bài toán đặt ra là cho một dãy mã dạng số nguyên không quá 18 chữ số. Hãy loại bỏ đoạn mã 084 ra khỏi mã ban đầu.

Dữ liệu đảm bảo dãy mã luôn có duy nhất một lần cụm 084.

Input

Dòng đầu ghi số bộ test. Mỗi test là một số nguyên có ít nhất 4 chữ số nhưng không quá 18 chữ số.

Output

Ghi ra kết quả sau khi loại bỏ 084

Ví dụ

Input	Output
3	123567
123084567	3300478845
3300478808445	1
1084	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

12 Số liền kề

Viết chương trình kiểm tra số nguyên dương N có thỏa mãn tính chất: tất cả các chữ số cạnh nhau chỉ sai khác nhau đúng một đơn vị hay không. Ví dụ: số 123212 là số thỏa mãn, số 34578 không thỏa mãn.

Input

Dòng đầu ghi số bộ test (không quá 20). Mỗi test là 1 số nguyên dương N có ít nhất 2 chữ số, nhưng không quá 18 chữ số.

Output

Ghi ra YES hoặc NO

Ví dụ

Input	Output
3	YES
123212	YES
5654345676	NO
10233211123	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

13 Khoảng cách

Cho 2 điểm A và B trong không gian Oxy. Hãy tính khoảng cách giữa hai điểm đó.

Input

- Dòng đầu ghi số bộ test, không quá 20.
- Mỗi bộ test có 4 số thực lần lượt là tọa độ của 2 điểm A và B, giá trị tuyệt đối không quá 1000.

Output

Với mỗi bộ test, viết ra khoảng cách giữa 2 điểm với 4 chữ số sau dấu phẩy.

Ví dụ

Input	Output
2	5.0000
0 0 0 5	193.0648
0 199 5 6	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

14 Số may mắn

SỐ MAY MẮN

Bài làm tốt nhất

John rất thích con số 86 vì theo John đó là con số may mắn. Khi bắt gặp một số nguyên X John muốn kiểm tra xem trong dạng biểu diễn của nó kết thúc là số 86 hay không? Ví dụ số 111539786 kết thúc là số 86, còn số 123456789 thì không.

Nhiệm vụ của bạn là viết một chương trình đọc số nguyên X và kiểm tra xem trong dạng biểu diễn của nó kết thúc là số 86 hay không?

Input

Dữ liệu vào gồm nhiều bộ dữ liệu tương ứng với nhiều test. Dòng đầu tiên chứa một số nguyên dương không lớn hơn 20 là số lượng các bộ dữ liệu. Các dòng tiếp theo chứa các bộ dữ liệu, các số không quá 9 chữ số.

Output

Với mỗi bộ dữ liệu, ghi ra trên một dòng câu trả lời, ghi số 1 nếu trong dạng biểu diễn của nó kết thúc là số 86, ghi số 0 trong trường hợp ngược lại.

Ví dụ

Input:

```
3  
111539786  
123456789  
8686
```

Output:

```
1  
0  
1
```

Giới hạn thời gian: 1s
Giới hạn bộ nhớ: 2048 Kb

15 Phân tích thừa số nguyên tố - 1

PHÂN TÍCH THỪA SỐ NGUYÊN TỐ - 1

Bài làm tốt nhất

Cho số tự nhiên N. Nhiệm vụ của bạn là hãy đưa ra tất cả các ước số nguyên tố của N cùng lũy thừa của nó. Ví dụ $N=100 = 2^2 \times 5^2$.

$$N = 35 = 5^1 \times 7^1$$

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên N.
- T, N thỏa mãn ràng buộc $1 \leq T \leq 100$; $1 \leq N \leq 10000$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2 2 5 2
100	5 1 7 1
35	

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

16 Tính tổng chữ số

TÍNH TỔNG CHỮ SỐ

Bài làm tốt nhất

Thầy Tí rất thích thú với những con số, cô giáo đã giao cho Tí một bài tập về rút gọn các con số. Phép rút gọn được thực hiện như sau: từ một số ban đầu, số mới được tạo thành bằng cách cộng các chữ số của số ban đầu với nhau. Sau đó Tí phải thực hiện tiếp tục với con số vừa mới thu được.

Quá trình rút gọn kết thúc khi số thu được chỉ có duy nhất 1 chữ số.

Các bạn hãy cùng Tí đi tìm chữ số cuối cùng của phép rút gọn!

Input

Dòng đầu tiên gồm số lượng test T ($T \leq 100$).
T dòng tiếp theo, mỗi dòng gồm một số nguyên dương ($\leq 10^9$).

Output

Hãy in ra chữ số cuối cùng sau khi thực hiện phép rút gọn.

Example

Input:

```
3  
1009  
167  
102
```

Output:

```
1  
5  
3
```

Giới hạn thời gian: 1s
Giới hạn bộ nhớ: 65536 Kb

17 Ước số chung - Bội số chung

Cho hai số nguyên a, b. Nhiệm vụ của bạn là tìm bộ số chung nhỏ nhất và ước số chung lớn nhất của a và b. Bộ số chung nhỏ nhất của a và b ký hiệu là LCM(a, b) và ước số chung lớn nhất của a và b ký hiệu là GCD(a,b).

Input:

- Dòng đầu tiên đưa vào T là số lượng bộ test.
- T dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một cặp số a, b được viết cách nhau một vài khoảng trắng.
- T, a, b thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq a, b \leq 10^8$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	10 5
5 10	56 2
14 8	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

18 Ước số chung lớn nhất của N số nguyên dương đầu tiên (Sửa lại: BCNN của N số đầu tiên)

Cho số tự nhiên n. Nhiệm vụ của bạn là tìm số nguyên nhỏ nhất chia hết cho 1, 2, .., n.

Input:

- Dòng đầu tiên đưa vào T là số lượng bộ test.
- T dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một số tự nhiên n.
- T thỏa mãn ràng buộc: $1 \leq T \leq 10^4$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	6
3	60
5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

19 Kiểm tra nguyên tố

Một số được gọi là số nguyên tố nếu nó chỉ có 2 ước là 1 và chính nó. Số 0 và 1 không được coi là số nguyên tố.

Yêu cầu: Cho số n, hãy kiểm tra xem n có là số nguyên tố hay không.

Dữ liệu:

Một dòng duy nhất chứa số n ($0 \leq n \leq 10^9$)

Kết quả:

In ra "YES" nếu n là số nguyên tố, và "NO" trong trường hợp còn lại.

Input	Output
2	YES

Input	Output
4	NO

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

20 Phân tích thừa số nguyên tố - 2

Cho số nguyên dương n ($2 \leq n \leq 10^9$), hãy phân tích n ra thừa số nguyên tố.

Input:

Một dòng duy nhất chứa số n.

Kết quả:

Mỗi dòng ghi một thừa số nguyên tố và số mũ tương ứng cách nhau bởi dấu cách.

Các thừa số nguyên tố in ra theo thứ tự tăng dần.

Ví dụ:

Input	Output
4	2 2

Input	Output
168	2 3
	3 1
	7 1

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 524288 Kb

21 Liệt kê số nguyên tố - 1

Viết chương trình cho phép nhập vào hai số nguyên dương và tìm tất cả các số nguyên tố nằm trong khoảng đó.

Input:

Chỉ có 2 số nguyên dương a và b (không quá 10^6)

Output:

Ghi ra lần lượt các số nguyên tố trong khoảng. Cách nhau một khoảng trống.

Ví dụ:

Input	Output
10 50	11 13 17 19 23 29 31 37 41 43 47

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

22 Liệt kê số nguyên tố - 2

Hãy sinh ra tất cả các số nguyên tố trong khoảng [M, N]. Ví dụ M=1, N=10 ta có kết quả 2 3 5 7.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp mỗi dòng đưa vào một bộ test. Mỗi bộ test là bộ đôi M, N được viết cách nhau một vài khoảng trống.
- T, M, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq M \leq N \leq 10000$; $N - M \leq 10000$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2 3 5 7
1 10	3 5
3 5	

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

23 Cặp số nguyên tố đầu tiên có tổng bằng N

Cho số tự nhiên N. Hãy tìm cặp số nguyên tố đầu tiên có tổng là N. Nếu không tồn tại cặp số nguyên tố có tổng bằng N, hãy đưa ra -1.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm là một số N được ghi trên một dòng.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2 2
4	3 5
8	

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

24 Ước số nguyên tố

ƯỚC SỐ NGUYÊN TỐ

Bài làm tốt nhất

Cho số nguyên dương N. Hãy đưa ra tất cả các ước số nguyên tố của N.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một số nguyên dương N được ghi trên một dòng.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq N \leq 10^{10}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3 3 5 7
315	31
31	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

25 Ước số nguyên tố lớn nhất

ƯỚC SỐ NGUYÊN TỐ LỚN NHẤT

Bài làm tốt nhất

Cho số nguyên dương N. Hãy đưa ra ước số nguyên tố lớn nhất của N.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một số nguyên dương N được ghi trên một dòng.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq N \leq 10^{10}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7
315	31
31	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

26 Ước số nguyên tố nhỏ hơn N

ƯỚC SỐ NGUYÊN TỐ NHỎ HƠN N

Bài làm tốt nhất

Cho số nguyên dương N. Hãy đưa ra tất cả các ước số nguyên tố nhỏ hơn hoặc bằng N.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một số nguyên dương N được ghi trên một dòng.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq N \leq 10^4$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2 3 5 7
10	2 3 5 7 11 13 17 19 23 29 31
35	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

27 Ước số nguyên tố thứ K

ƯỚC SỐ NGUYÊN TỐ THỨ K

Bài làm tốt nhất

Cho số tự nhiên N. Nhiệm vụ của bạn là hãy đưa ra ước số nguyên tố thứ k của N. Đưa ra -1 nếu không tồn tại ước số thứ k của N.

Ví dụ N = 225, k=2 ta có kết quả là 3 vì $225 = 3 \times 3 \times 5 \times 5$. Với N = 81, k = 5 ta có kết quả -1 vì $81 = 3 \times 3 \times 3 \times 3$.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một bộ đôi N và k.
- T, N thỏa mãn ràng buộc $1 \leq T \leq 100$; $1 \leq N \leq 10^4$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3
225 2	-1
81 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

28 Liệt kê số có ba ước số

LIỆT KÊ SỐ CÓ BA ƯỚC SỐ

Bài làm tốt nhất

Cho số tự nhiên N. Nhiệm vụ của bạn là hãy liệt kê tất cả các số có đúng ba ước số. Ví dụ n=100, ta có các số 4, 9, 25, 49.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số N.
- T, N thỏa mãn ràng buộc $1 \leq T \leq 100$; $1 \leq N \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4 9 25 49
50	4 9 25 49 121 169
200	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

29 Cặp số nguyên tố

CẶP SỐ NGUYÊN TỐ

Bài làm tốt nhất

Cho số nguyên dương $N > 2$. Hãy đưa ra cặp số nguyên tố p, q đầu tiên tìm được có tổng đúng bằng N. Ví dụ $N = 6$ ta có cặp số nguyên tố đầu tiên là $3 + 3 = 6$.

Trong trường hợp không tìm thấy đáp án thì không in ra gì cả.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số chẵn N.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $4 \leq N \leq 10000$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng. Nếu không có cặp số nào thỏa mãn thì không in ra gì cả.

Input:	Output:
2	3 71
74	3 1021
1024	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

30 Số Fibonacci thứ N

SỐ FIBONACCI THỨ N

Dãy số Fibonacci được định nghĩa theo công thức như sau:

$$F_1 = 1$$

$$F_2 = 1$$

$$F_n = F_{n-1} + F_{n-2} \text{ với } n > 2$$

Viết chương trình tính số Fibonacci thứ n (với n không quá 92)

Dữ liệu vào: Dòng đầu ghi số bộ test. Mỗi bộ test là một số nguyên n.

Kết quả: Với mỗi bộ test, ghi ra số Fibonacci thứ n trên một dòng.

Ví dụ

Input	Output
3	1
2	5
5	6765
20	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

31 Chia dư

CHIA DƯ

Bài làm tốt nhất

Cho hai số nguyên dương a và m. Nhiệm vụ của bạn là tìm x nhỏ nhất trong khoảng $[0, m-1]$ sao cho $a * x \equiv 1 \pmod{m}$. Ví dụ a = 3, m=11 ta tìm được x = 4 vì $4 * 3 \% 11 = 1$.

Input:

- Đòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là bộ đôi a, m được viết cách nhau một vài khoảng trắng.
- T, a, m thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq a \leq m \leq 100$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng. Nếu chương trình đồng dư không có nghiệm, hãy đưa ra -1

Input:	Output:
2	4
3 11	12
10 17	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

32 Chia dư từ 1 đến N

CHIA DƯ TỪ 1 ĐẾN N

Bài làm tốt nhất

Cho hai số nguyên không âm N và K. Nhiệm vụ của bạn là tìm $S = 1\%K + 2\%K + \dots + N\%K$. Ví dụ với $N = 10$, $K=2$ ta có $S = 1\%2 + 2\%2 + 3\%2 + 4\%2 + 5\%2 + 6\%2 + 7\%2 + 8\%2 + 9\%2 + 10\%2 = 5$. Yêu cầu độ phức tạp thuật toán là hằng số

Input:

- Đòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là bộ đôi N, K được viết cách nhau một vài khoảng trắng.
- T, N, K thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq N \leq 1000$; $0 \leq K \leq 10^{12}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	55
10 55	1
1 11	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

33 Tổng chia dư cho K

TỔNG CHIA DƯ CHO K

Bài làm tốt nhất

Cho hai số nguyên không âm N và K. Nhiệm vụ của bạn là kiểm tra xem $K = 1\%K + 2\%K + \dots + N\%K$ hay không. Đưa ra 1 hoặc 0 nếu N, K thỏa mãn hoặc không thỏa mãn yêu cầu bài toán. Ví dụ với $N = 10, K=55$ ta có kết quả là 1 vì $55 = 1\%55 + 2\%55 + 3\%55 + \dots + 10\%55$. Ngược lại, $N=4, K=11$ có kết quả là 0 vì $11 \neq 1\%11 + 2\%11 + 3\%11 + 4\%11$.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là bộ đôi N, K được viết cách nhau một vài khoảng trắng.
- T, N, K thỏa mãn ràng buộc : $1 \leq T \leq 100; 0 \leq N \leq 1000; 0 \leq K \leq 10^{12}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
10 55	0
1 11	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

34 Chênh lệch nhỏ nhất

CHÊNH LỆCH NHỎ NHẤT

Bài làm tốt nhất

Cho dãy số A[] gồm có N phần tử. Bạn cần tìm chênh lệch nhỏ nhất giữa hai phần tử bất kì trong dãy số đã cho.

Input:

- Dòng đầu tiên là số lượng bộ test T ($T \leq 10$).
- Mỗi test gồm số nguyên N ($1 \leq N \leq 100\,000$).
- 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 là đáp án tìm được.

Ví dụ:

Input:	Output
3	1
6	4
1 5 3 19 18 25	1
4	
30 5 20 9	
7	
1 19 2 31 38 25 100	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

35 Khoảng cách nhỏ nhất

Cho mảng A[] gồm n số chưa được sắp xếp. Hãy tìm $\text{Min}(A[i]-A[j]) : i \neq j$ và $i, j = 0, 1, 2, \dots, n-1$. Ví dụ với $A[] = \{1, 5, 3, 19, 18, 25\}$ ta có kết quả là $1 = 19-18$. với $A[] = \{1, 19, -4, 31, 28, 35, 100\}$ ta có kết quả là $3 = 31-28$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n; dòng tiếp theo là n số $A[i]$ của mảng A[]; các số được viết cách nhau một vài khoảng trắng.
- T, n, $A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $-10^3 \leq A[i] \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
5	6
2 4 5 7 9	
10	
87 32 99 75 56 43 21 10 68 49	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

36 Số nhỏ nhất chưa xuất hiện

Cho mảng A[] gồm n số nguyên bao gồm cả số 0. Nhiệm vụ của bạn là tìm số nguyên dương nhỏ nhất không có mặt trong mảng. Ví dụ với mảng $A[] = \{5, 8, 3, 7, 9, 1\}$, ta có kết quả là 2.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trắng.
- T, n, $A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^6$; $-10^6 \leq A[i] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	6
5	2
1 2 3 4 5	
5	
0 -10 1 3 -20	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

37 Dãy tam giác

Cho mảng A[] gồm n số được thiết lập theo nguyên tắc nửa đầu tăng dần nửa sau giảm dần. Hãy tìm số lớn nhất của mảng. Ví dụ với mảng $A[] = \{1, 2, 3, 4, 5, 2, 1\}$, ta có kết quả 5.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trắng.
- T, n, $A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $0 \leq A[i] \leq 10^7$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

38 Phản tử lớn nhất

PHẢN TỬ LỚN NHẤT

Bài làm tốt nhất

Cho mảng $A[]$ gồm n phần tử. Hãy tìm phản tử lớn nhất của mảng. Ví dụ với mảng $A[] = \{7, 10, 4, 3, 20, 15\}$ ta nhận được kết quả là 20.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phản tử của mảng $A[]$; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trắng.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^5$; $1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	20
6	12
7 10 4 3 20 15	
6	
9 7 12 8 6 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

39 Quay vòng dãy số 1

QUAY VÒNG DÃY SỐ 1

Bài làm tốt nhất

Cho mảng $A[]$ gồm n phần tử và số d . Hãy thực hiện phép quay vòng d phản tử của mảng $A[]$. Ví dụ với mảng $A[] = \{1, 2, 3, 4, 5\}$, $d = 2$ thì ta có kết quả $A[] = \{3, 4, 5, 1, 2\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là hai số n và d ; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trắng.
- $T, n, d, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 10$; $1 \leq d < n \leq 10^6$; $1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

40 Phản tử nhỏ nhất thứ K

PHẢN TỬ NHỎ NHẤT THỨ K

Bài làm tốt nhất

Cho mảng $A[]$ gồm n số và số k . Hãy tìm phản tử nhỏ nhất thứ k của mảng. Ví dụ với mảng $A[] = \{7, 10, 4, 3, 20, 15\}$, $k=3$ ta nhận được số nhỏ nhất thứ k là 7.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phản tử của mảng $A[]$ và số k ; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trắng.
- $T, n, k, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq k \leq n \leq 10^5$; $1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7
6 3	8
7 10 4 3 20 15	
6 4	
9 7 12 8 6 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

41 Tính tổng trong khoảng

TÍNH TỔNG TRONG KHOẢNG

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử và Q câu hỏi. Mỗi câu hỏi Q là bộ đôi hai số L và R. Nhiệm vụ của bạn là tìm tổng các phần tử của mảng A[] của mỗi câu hỏi Q. Ví dụ với mảng A[] = {1, 2, 1, 3, 4, 5, 2, 8}, các câu hỏi Q: [1, 5], [2, 4], [3, 5] ta sẽ có các câu trả lời: 8, 4, 6.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba phần: phần thứ nhất đưa vào n, Q là số phần tử của mảng A[] và số lượng câu hỏi Q; phần tiếp theo đưa vào n số A[i] của mảng; phần cuối cùng đưa vào Q câu hỏi, mỗi câu hỏi là một bộ đôi L, R; các số được viết cách nhau một vài khoảng trắng.
- T, n, Q, L, R, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq L \leq R \leq n$, $Q \leq 10^4$; $1 \leq A[i] \leq 10^3$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	8
9 3	4
1 1 2 1 3 4 5 2 8	6
1 5	
2 4	
3 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

42 Biến đổi nhị phân

BIẾN ĐỔI NHỊ PHÂN

Bài làm tốt nhất

Cho ma trận A[N][M] chỉ bao gồm các số 0 và 1. Hãy sửa đổi các phần tử của ma trận A[][] theo nguyên tắc: nếu phần tử $A[i][j] = 1$ ta thay tất cả các phần tử của hàng i, cột j bởi 1. Ví dụ với ma trận dưới đây sẽ minh họa cho phép biến đổi:

1	0	0	1	1	1	1	1
0	0	1	0	1	1	1	1
0	0	0	0	1	0	1	1

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: Dòng đầu tiên đưa vào hai số N, M ; dòng tiếp là $N \times M$ các phần tử của ma trận A[][], các phần tử được viết cách nhau một vài khoảng trắng.
- T, N, M thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N, M \leq 100$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	0 0 1
2 3	1 1 1
0 0 0	1 1 1 1
0 0 1	1 1 1 1
3 4	1 0 1 1
1 0 0 1	
0 0 1 0	
0 0 0 0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

43 Biến của ma trận

Cho ma trận vuông A[N][N]. Hãy in các phần tử thuộc vùng biên.

Input :	1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
Output :	1 2 3 4 5 8 1 4 5 6 7 8

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N là cấp của ma trận A[N][N]; dòng tiếp theo đưa vào N×N số A[i][j] ; các số được viết cách nhau một vài khoảng trắng.
- T, N, A[i][j] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 100$; $1 \leq A[i][j] \leq 150$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

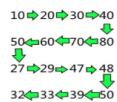
Input:	Output:
2	1 2 3 4
4	5 8
1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8	1 4
3	5 6 7 8
45 48 54 21 89 87 70 78 15	45 48 54 21 87 70 78 15

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

44 In ma trận – 1

Cho ma trận vuông A[N][N]. Hãy in các phần tử thuộc theo hình con rắn.



Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N là cấp của ma trận A[N][N]; dòng tiếp theo đưa vào N×N số A[i][j] ; các số được viết cách nhau một vài khoảng trắng.
- T, N, A[i][j] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 100$; $1 \leq A[i][j] \leq 150$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	45 48 54 87 89 21 70 78 15
3	25 27 21 23
45 48 54 21 89 87 70 78 15	
2	
25 27 23 21	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

45 Ma trận nhị phân

Cho ma trận A[] có N hàng và 3 cột, trong đó các vị trí là các giá trị nhị phân (0 hoặc 1). Hãy đếm xem có bao nhiêu hàng mà số lượng số 1 nhiều hơn số lượng số 0.

Input

Dòng đầu ghi số nguyên dương N (không quá 1000).

N dòng tiếp theo, mỗi dòng ghi 3 giá trị nhị phân.

Output

Ghi ra số dòng mà số lượng số 1 nhiều hơn số lượng số 0.

Ví dụ

Input	Output
3	2
1 1 0	
1 1 1	
1 0 0	
2	1
1 0 0	
0 1 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

46 Liệt kê số khác nhau

Cho dãy số A[] chỉ bao gồm các số nguyên dương không quá 1000. Hãy liệt kê các số khác nhau trong dãy theo thứ tự tăng dần.

Input

Dòng đầu ghi số n là số phần tử của dãy (không quá 100).

Dòng thứ 2 ghi n số của mảng.

Output

Ghi ra các số khác nhau trong dãy A[] theo thứ tự tăng dần, mỗi số cách nhau một khoảng trống.

Ví dụ

Input	Output
10	3 6 7 9 13
3 7 9 3 6 13 7 7 9 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

47 Tích ma trận

Viết chương trình tính tích hai ma trận A cỡ $n*m$ và ma trận B cỡ $m*p$.

Với $1 < n,m,p < 50$. Các giá trị trong ma trận đều nguyên dương và không vượt quá 1000.

Input

Dòng đầu ghi 3 số n,m,p

n dòng tiếp theo ghi ma trận A

m dòng tiếp theo ghi ma trận B

Output

Ghi ra ma trận tích

Ví dụ

Input	Output
3 4 3	30 30 30
1 2 3 4	21 21 21
4 2 3 1	25 25 25
2 4 1 3	
1 1 1	
2 2 2	
3 3 3	
4 4 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

48 Đếm số phần tử lặp lại

Cho mảng $A[]$ gồm N phần tử. Hãy đếm số phần tử bị lặp lại ít nhất 1 lần. Ví dụ với mảng $A[] = \{5, 6, 1, 2, 1, 4\}$ thì ta có đáp án là 2 vì có 2 phần tử 1.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào số phần tử của mảng N ; dòng tiếp theo là N số $A[i]$ là các phần tử của mảng $A[]$.
- $T, N, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 10^6$, $1 \leq A[i] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
5	4
4 5 1 2 1	
6	
10 20 30 30 20 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

49 Ký tự không lặp

KÝ TỰ KHÔNG LẮP

Bài làm tốt nhất

Cho xâu ký tự S. Nhiệm vụ của bạn là in ra tất cả các ký tự không lặp khác nhau trong S. Ví dụ S = "ABCDEABC" ta nhận được kết quả là "DE".

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S được viết trên một dòng.
- T, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{Length}(S) \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng. In ra theo thứ tự chữ cái xuất hiện trong xâu ban đầu.

Input:	Output:
2 ABCDEABC ABC	DE ABC

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

50 Đếm từ

ĐẾM TỪ

Bài làm tốt nhất

Một từ được hiểu là dãy các ký tự liên tiếp không chứa ký tự space, '\t', '\n'. Cho xâu ký tự S có các ký tự space, '\t', '\n', hãy đếm số các từ của S.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S được viết trên một dòng.
- T, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{length}(S) \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 Print the number of words Print the number of words present in the string	5 9

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

51 Xâu Pangram

XÂU PANGRAM

Bài làm tốt nhất

Một xâu ký tự được gọi là Pangram nếu nó chứa đầy đủ các ký tự từ 'a', ..'z'. Cho xâu ký tự S và số K. Hãy xác định xem có thể thực hiện nhiều nhất K phép biến đổi các ký tự để S trở thành Pangram hay không? Mỗi phép biến đổi là một phép thay thế ký tự này bằng một ký tự khác.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào xâu ký tự S; dòng tiếp theo đưa vào số K.
- T, S, K thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq K \leq 62$; $1 \leq \text{Length}(S) \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 qwqqwqeqqwdsdadsdasadsfsdsdasasas	0 1
4 qwqqwqeqqwdsdadsdasadsfsdsdasasas	
24	

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

52 Loại bỏ từ trong xâu

LOẠI BỎ TỪ TRONG XÂU

Bài làm tốt nhất

Viết chương trình cho phép nhập vào một chuỗi và từ cần loại bỏ khỏi chuỗi. Thực hiện loại bỏ từ và in ra kết quả

Trong đó:

INPUT

- Hàng thứ nhất là chuỗi ban đầu
- Hàng tiếp theo là từ cần loại bỏ

OUTPUT

- Chuỗi kết quả

Input	Output
Tôi Yêu PTIT	Yêu PTIT
Tôi	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

53 Chúc mừng năm mới

CHÚC MỪNG NĂM MỚI

Bài làm tốt nhất

Ti năm nay đã lên lớp 1 rồi, Tết đến Ti rất vui vì nhận được rất nhiều lời chúc.

Vì mỗi tập viết nên Ti đã ghi lại tất cả các lời chúc đó. Cũng vì rất trân trọng các lời chúc nên Ti đã ghi tất cả các lời chúc bằng chữ IN HOA, tuy nhiên do mỗi tập viết nên Ti ghi không có dấu. Giờ ngồi lật lại cuốn nhật ký ghi các lời chúc, Ti thấy mình đã ghi được n lời chúc.

Ti muốn biết có bao nhiêu lời chúc khác nhau (hai lời chúc được gọi là khác nhau nếu chúng có độ dài khác nhau hoặc tồn tại ít nhất một vị trí mà ký tự ở vị trí đó của hai lời chúc là khác nhau, hay nói cách khác, đó là hai xâu ký tự khác nhau). Bạn hãy lập chương trình giúp Ti đếm xem có bao nhiêu lời chúc khác nhau nhé.

Input:

Dòng đầu chứa số nguyên dương n là số lời chúc Ti ghi được;

n dòng tiếp theo, mỗi dòng chứa một xâu ký tự S là một lời chúc.

n, S thỏa mãn ràng buộc: $1 \leq n \leq 10^4$; Các lời chúc S có độ dài không quá 30 ký tự gồm các chữ cái la tinh IN HOA 'A'... 'Z' và dấu cách.

Output:

Một số nguyên dương duy nhất là số lời chúc khác nhau.

Ví dụ:

Input:	Output:
4	3
CHUC MUNG NAM MOI	
HAPPY NEW YEAR	
CHUC MUNG TUOI MOI	
CHUC MUNG NAM MOI	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

54 Số đẹp

SỐ ĐẸP

Bài làm tốt nhất

Một số được coi là đẹp nếu đó là số thuận nghịch và chỉ toàn các chữ số chẵn. Viết chương trình đọc vào các số nguyên dương có không quá 500 chữ số và kiểm tra xem số đó có đẹp hay không.

Input:

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng số nguyên dương n không quá 500 chữ số.

Output:

Mỗi bộ test viết ra trên một dòng chữ YES nếu đó là số đẹp, chữ NO nếu ngược lại

Ví dụ

Input	Output
4	NO
123456787654321	YES
86442824468	YES
8006000444422220000222244440006008	NO
235365789787654324567856578654356786556	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

55 Nhỏ nhất – lớn nhất

NHỎ NHẤT - LỚN NHẤT

Bài làm tốt nhất

Cho số tự nhiên m và số nguyên s không âm. Nhiệm vụ của bạn là tìm số bé nhất và lớn nhất có m chữ số và tổng chữ số bằng s.

Input

Dòng đầu gồm 2 số m và s ($1 \leq m \leq 100, 0 \leq s \leq 900$).

Output

In ra kết quả của bài toán.

Số đầu tiên là số bé nhất, số thứ hai là số lớn nhất. Nếu không có đáp án in ra “-1 -1”.

Example

Input:

2 15

Output:

69 96

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

56 Chia hết cho 11

CHIA HẾT CHO 11

Bài làm tốt nhất

Cho số tự nhiên N, hãy kiểm tra xem N có phải là số chia hết cho 11 hay không? Đưa ra 1 nếu N chia hết cho 11, trái lại đưa ra 0.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một số tự nhiên N.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100; 0 \leq N \leq 10^{1000}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
76945	0
363588395960667043875487	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

57 Chia hết cho 5

CHIA HẾT CHO 5

Bài làm tốt nhất

Cho số tự nhiên N được biểu diễn dưới dạng nhị phân. Hãy kiểm tra xem N có phải là số chia hết cho 5 hay không? Đưa ra “Yes” nếu N chia hết cho 5, trái lại đưa ra “No”.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một biểu diễn nhị phân của N.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100; 0 \leq N \leq 10^{1000}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	Yes
1010	Yes
10000101001	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

58 Địa chỉ email – 1

ĐỊA CHỈ EMAIL - 1

Bài làm tốt nhất

Địa chỉ email của cán bộ PTIT được cấp theo nguyên tắc ghép tên với chữ cái đầu tiên của họ và tên đệm. Viết chương trình cho phép tạo các địa chỉ email theo tên cán bộ

Input

Chỉ có một xâu ký tự họ tên độ dài không quá 100.

Output

Ghi ra địa chỉ email theo mẫu trong ví dụ.

Ví dụ

Input	Output
Nguyen vAn nAM	namnv@ptit.edu.vn

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

59 Chuẩn hóa họ tên

CHUẨN HÓA HỌ TÊN

Bài làm tốt nhất

Tên người dùng sẽ được chuẩn hóa theo nguyên tắc tên được viết sau cùng, phân tách với phần tên đệm và tên bởi dấu phẩy. Các chữ cái nằm trong tên đều được viết hoa; chữ cái đầu tiên của tên đệm và họ được viết hoa, các chữ cái còn lại viết thường.

Input

Chỉ có một xâu họ tên độ dài không quá 100

Output

Ghi ra xâu họ tên đã chuẩn hóa theo mẫu trong ví dụ

Ví dụ

Input	Output
ngUyeN vAN Nam	Nguyen Van, NAM

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

60 Dấu cuối giống nhau

ĐẦU CUỐI GIỐNG NHAU

Bài làm tốt nhất

Cho xâu ký tự S. Hãy đếm tất cả các xâu con của S có ký tự đầu và ký tự cuối giống nhau. Ví dụ với xâu “aba” ta có 4 xâu con bao gồm: “a”, “b”, “a”, “aba”.

Input:

- Đưa đầu tiên dưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S được viết trên một dòng.
- T, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq k \leq 26$; $1 \leq \text{Length}(S) \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7
abcab	4
aba	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

61 Dãy ưu thế

DÃY UY THẾ

Bài làm tốt nhất

Cho dãy A[] chỉ bao gồm các số nguyên dương không quá 10^5 nhưng không biết trước số phần tử của dãy. Người ta gọi dãy A[] là dãy ưu thế nếu thỏa mãn 1 trong 2 điều kiện sau đây:

- Dãy gọi là ưu thế chẵn nếu số phần tử của dãy là chẵn và số lượng số chẵn trong dãy nhiều hơn số lượng số lẻ.
- Dãy gọi là ưu thế lẻ nếu số phần tử của dãy là lẻ và số lượng số lẻ trong dãy nhiều hơn số lượng số chẵn.

Hãy kiểm tra xem dãy A[] có phải là dãy ưu thế hay không.

Dữ liệu vào

- Dòng đầu ghi số bộ test, không quá 10
- Mỗi bộ test là một dãy các số nguyên dương (không quá 10^4) và có không quá 200 số, các số cách nhau 1 khoảng trắng, không biết trước số lượng phần tử.

Kết quả

- Nếu dãy A[] thỏa mãn là dãy ưu thế thì in ra YES, nếu không in ra NO

Ví dụ

Input	Output
2	YES
11 22 33 44 55 66 77	NO
23 34 45 56 67 78 89 90 121 131 141 151 161 171	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

62 Mã hóa

MÃ HÓA

Bài làm tốt nhất

Cho một xâu ký tự độ dài không quá 100 chỉ bao gồm các chữ cái in hoa. Người ta thực hiện mã hóa bằng cách đếm các ký tự cạnh nhau giống nhau và viết số lượng phía sau các chữ cái đó.

Ví dụ xâu AAEEEEEEGD thì được mã hóa thành A2E1C4G3D1

Với giả thiết không có ký tự nào xuất hiện nhiều hơn 9 lần liên tiếp. Hãy viết chương trình mã hóa xâu ký tự theo cách như trên.

Input

Dòng đầu ghi số bộ test. Mỗi bộ test ghi xâu chữ cái in hoa không quá 100 ký tự. Không có ký tự nào xuất hiện nhiều hơn 9 lần liên tiếp.

Output

Với mỗi test ghi ra kết quả mã hóa.

Ví dụ

Input	Output
2	A8
AAAAAAA	A2E1C4G3D1
AAECCCCGGGD	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

63 Loại bỏ nguyên âm

Cho một xâu ký tự S chỉ bao gồm các ký tự chữ cái và không có khoảng trắng. Hãy loại bỏ các nguyên âm trong S.

Kết quả được viết ra dưới dạng chữ cái viết thường của các phụ âm có mặt trong S, trước mỗi phụ âm ghi một ký tự dấu chấm ‘.’

Các nguyên âm bao gồm: ‘A’, ‘E’, ‘I’, ‘O’, ‘U’, ‘Y’ (cả viết hoa và viết thường).

Input

Chỉ có xâu S, độ dài không quá 100.

Output

Ghi ra xâu kết quả

Ví dụ

Input	Output
HocVienCNBCVT	.h.c.v.n.c.n.b.c.v.t

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

64 Sắp xếp - 1 – 2

SẮP XẾP 0 -1 - 2

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử. Các phần tử của mảng A[] chỉ bao gồm các số 0, 1, 2. Hãy sắp xếp mảng A[] theo thứ tự tăng dần. Ví dụ với A[] = {0, 2, 1, 2, 0} ta kết quả A[] = {0, 0, 1, 2, 2}.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng; dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng tiếp theo là n số A[i] của mảng A[] các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq A[i] \leq 2$; $1 \leq n \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	0 0 1 2 2
5	0 0 1
0 2 1 2 0	
3	
0 1 0	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

65 Sắp xếp xén kẽ - 1

SẮP XẾP XEN KẼ - 1

Bài làm tốt nhất

Cho mảng A[] gồm n số nguyên khác nhau. Hãy đưa ra các phần tử của mảng theo khuôn dạng lớn nhất, nhỏ nhất, lớn thứ hai, nhỏ thứ 2, ... Ví dụ với A[] = {9, 7, 12, 8, 6, 5} ta đưa ra : 12, 5, 9, 6, 8, 7.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n; dòng tiếp theo là n số A[i] của mảng A[]; các số được viết cách nhau một vài khoảng trắng.
- T, n thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7 1 6 2 5 3 4
7	9 1 8 2 7 3 6 4
7 1 2 3 4 5 6	
8	
1 6 9 4 3 7 8 2	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

66 Sắp xếp chữ số

SẮP XẾP CHỮ SỐ

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử. Nhiệm vụ của bạn là đưa ra mảng đã được sắp xếp bao gồm các chữ số của mỗi phần tử trong A[]. Ví dụ A[] = {110, 111, 112, 113, 114 } ta có kết quả là {0, 1, 2, 3, 4}.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng tiếp theo là n số A[i]; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $0 \leq A[i] \leq 10^{16}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 3 4 8
3	1 2 3 4 6
131 11 48	
4	
111 222 333 446	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

67 Tính tích

Cho mảng A[] gồm n phần tử và mảng B[] gồm m phần tử. Nhiệm vụ của bạn là tìm tích giữa phần tử lớn nhất của mảng A[] và phần tử nhỏ nhất của mảng B[]. Ví dụ A[] = {5, 7, 112, 9, 3, 6, 2}, B[] = {1, 2, 6, -1, 0, 9} ta có kết quả là $-9 = 9 * (-1)$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào n, m tương ứng với số phần tử của mảng A[] và B[]; dòng tiếp theo là n số A[i]; dòng cuối cùng là m số B[i]; các số được viết cách nhau một vài khoảng trắng.
- T, n, m, A[i], B[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n, m \leq 10^6$; $-10^8 \leq A[i] \leq 10^8$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	-9
6 6	20
5 7 9 3 6 2	
1 2 6 -1 0 9	
6 6	
1 4 2 3 10 2	
4 2 6 5 2 9	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

68 Đếm cặp phần tử có tổng bằng K

Cho mảng A[] gồm n phần tử và số k. Đếm tất cả các cặp phần tử của mảng có tổng bằng k. Ví dụ A[] = {1, 5, 3, 4, 2}, k = 7 ta có kết quả là 2 cặp (3, 4), (5, 2).

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[] và k; dòng tiếp theo là n số A[i] của mảng A[]; các số được viết cách nhau một vài khoảng trắng.
- T, n, k, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 100$; $0 \leq k \leq 100$, $0 \leq A[i] \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

69 Hợp và giao của hai dãy số - 1

Cho mảng A[] gồm n phần tử, mảng B[] gồm m phần tử khác nhau. Các phần tử của mảng A[] và B[] đã được sắp xếp. Hãy tìm mảng hợp và giao được sắp giữa A[] và B[]. Ví dụ với A[] = {1, 3, 4, 5, 7}, B[] = {2, 3, 5, 6} ta có mảng hợp Union = {1, 2, 3, 4, 5, 6, 7}, mảng giao Intersection = {3, 5}. In ra đáp án theo giá trị phần tử từ nhỏ đến lớn.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào n, m là số phần tử của mảng A[] và B[]; dòng tiếp theo là n số A[i] của mảng A[]; dòng tiếp theo là m số B[i] của mảng B[]; các số được viết cách nhau một vài khoảng trắng.
- T, n, m, A[i], B[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n, m, A[i], B[i] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

70 Sắp đặt dãy số

SẮP ĐẶT DÃY SỐ

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng sao cho A[i] = i. Nếu phần tử A[j] của có giá trị khác j, hãy đưa ghi vào -1. Ví dụ với mảng A[] = {-1, -1, 6, 1, 9, 3, 2, -1, 4, -1} ta có kết quả A[] = {-1, 1, 2, 3, 4, -1, 1, -1, 9}.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $1 \leq A[i] \leq 10^{18}$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	-1 1
10	2 3 4 5 6 7 8 9 -1 -1
-1 -1 6 1 9 3 2 -1 4 -1	
6	0 1 2 3 4 5 6 7 8 9
0 -3 1 -2 3 -4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

71 Số đảo của dãy

SỐ 0 Ở CUỐI DÃY

Bài làm tốt nhất

Cho mảng $A[]$ gồm n phần tử. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng sao cho các số 0 để ở cuối cùng, các phần tử khác không được bảo toàn thứ tự trước sau. Ví dụ với mảng $A[] = \{1, 2, 0, 0, 0, 3, 6\}$ ta có kết quả $A[] = \{1, 2, 3, 6, 0, 0, 0\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng $A[]$; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trắng.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $0 \leq A[i] \leq 10^{18}$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2 3 6 0 0 0
7	1 2 3 0 0 0
1 2 0 0 0 3 6	
6	
0 1 0 2 0 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

72 Ghép dãy số

GHÉP DÃY SỐ

Bài làm tốt nhất

Cho k mảng mỗi mảng gồm n phần tử đã được sắp xếp. Hãy đưa ra kết quả là một dãy đã được sắp xếp. Ví dụ với $k = 3$, $n=4$ và mảng

```
A[] = { {1, 3, 5, 7},  
        {2, 4, 6, 8}  
        {0, 9, 10, 11}  
};  
sẽ cho ta kết quả  $A[] = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ .
```

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất dòng thứ nhất đưa là hai số k, n ; k dòng tiếp, mỗi dòng gồm n số của mảng $A[k][n]$. Các số được viết cách nhau một vài khoảng trắng.
- $T, n, k, A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq k \leq 10$; $1 \leq A[i][j] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	0 1 2 3 4 5 6 7 8 9 10 11
3 4	
1 3 5 7	
2 4 6 8	
0 9 10 11	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

73 Trộn hai dãy và sắp xếp

Cho mảng A[] gồm n phần tử và mảng B[] gồm m phần tử. Nhiệm vụ của bạn là hợp nhát hai mảng A[] và B[] để được một mảng mới đã được sắp xếp. Ví dụ A[] = {10, 5, 15}, B[] = {20, 3, 2} ta có kết quả là C[] = {2, 3, 5, 10, 15, 20}.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào n, m tương ứng với số phần tử của mảng A[] và B[]; dòng tiếp theo là n số A[i] ; dòng cuối cùng là m số B[i]; các số được viết cách nhau một vài khoảng trống.
- T, n, m, A[i], B[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n, m \leq 10^6$; $-10^8 \leq A[i] \leq 10^8$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 3 3 10 5 15 20 3 2	2 3 5 10 15 20

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

74 Bổ sung phần tử

Cho mảng A[] gồm n số nguyên dương. Gọi L, R là max và min các phần tử của A[]. Nhiệm vụ của bạn là tìm số phần tử cần thiết cần thêm vào mảng để mảng có đầy đủ các số trong khoảng [L, R]. Ví dụ A[] = {5, 7, 9, 3, 6, 2 } ta nhận được kết quả là 2 tương ứng với các số còn thiếu là 4, 8.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n, tương ứng với số phần tử của mảng A[]; dòng tiếp theo là n số A[i] ; các số được viết cách nhau một vài khoảng trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n, A[i] \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 5 4 5 3 8 6 3 2 1 3	1 0

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

75 Tìm kiếm tuần tự

TÌM KIẾM TUẦN TỤ

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử. Hãy tìm vị trí của phần tử đầu tiên có giá trị X trong mảng A[]. Nếu không tìm thấy X hãy đưa ra -1.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào n là số các phần tử của mảng A[] và số X cần tìm; dòng tiếp theo đưa vào n số A[i] ($1 \leq i \leq n$) các số được viết cách nhau một vài khoảng trắng.
- T, n, A, X thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N$, $X, A[i] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 5 16 9 7 2 16 4 7 98 1 22 57 47 34 18 66	4 -1

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

76 Tìm kiếm nhị phân

TÌM KIẾM NHỊ PHÂN

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử đã được sắp xếp. Hãy đưa ra 1 nếu X có mặt trong mảng A[], ngược lại đưa ra -1.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào n, X là số các phần tử của mảng A[] và số X cần tìm; dòng tiếp theo đưa vào n số A[i] ($1 \leq i \leq n$) các số được viết cách nhau một vài khoảng trắng.
- T, n, A, X thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N$, $X, A[i] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 5 16 9 7 2 16 4 7 98 1 22 57 47 34 18 66	1 -1

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

77 Số nhỏ nhất còn thiếu

Cho mảng $A[]$ gồm $n-1$ phần tử bao gồm các khác nhau từ 1, 2, .., n. Hãy tìm số nguyên dương nhỏ nhất không có mặt trong mảng $A[]$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào n; dòng tiếp theo đưa vào $n-1$ số $A[i]$; các số được viết cách nhau một vài khoảng trắng.
- T, n, A thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N, A[i] \leq 10^7$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
5	9
1 2 3 5	
10	
1 2 3 4 5 6 7 8 10	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

78 Tìm kiếm trong dãy sắp xếp vòng**TÌM KIẾM TRONG DÃY SẮP XẾP VÒNG****Bài làm tốt nhất**

Một mảng được sắp được chia thành hai đoạn tăng dần được gọi là mảng sắp xếp vòng. Ví dụ mảng $A[] = \{ 5, 6, 7, 8, 9, 10, 1, 2, 3, 4 \}$ là mảng sắp xếp vòng. Cho mảng $A[]$ gồm n phần tử, hãy tìm vị trí của phần tử x trong mảng $A[]$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào n và x; dòng tiếp theo đưa vào n số $A[i]$; các số được viết cách nhau một vài khoảng trắng.
- $T, n, A[i], x$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N, x, A[i] \leq 10^7$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	9
10 3	3
5 6 7 8 9 10 1 2 3 4	
10 3	
1 2 3 4 5 6 7 8 9 10	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

79 Số nhỏ nhất và nhỏ thứ hai

SỐ NHỎ NHẤT VÀ NHỎ THỨ HAI

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử, hãy đưa ra số nhỏ nhất và số nhỏ thứ hai của mảng. Nếu không có số nhỏ thứ hai, hãy đưa ra -1.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào n là số phần tử của mảng A[]; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N, A[i] \leq 10^7$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 10 5 6 7 8 9 10 1 2 3 4 5 1 1 1 1 1	1 2 -1

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

80 Tổng gần 0 nhất

TỔNG GẦN 0 NHẤT

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử, hãy tìm cặp phần tử có tổng gần nhất so với 0.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào n là số phần tử của mảng A[]; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq N \leq 10^3$, $-10^6 \leq A[i] \leq 10^6$.

Output:

- Đưa ra tổng gần nhất với 0 của cặp phần tử.

Input:	Output:
2	-68
3	-14
-8 -66 -60	
6	
-21 -67 -37 -18 4 -65	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

81 Liệt kê K phần tử lớn nhất

Cho mảng A[] gồm n phần tử, hãy tìm k phần tử lớn nhất của mảng. Các phần tử được đưa ra theo thứ tự giảm dần.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào N và K; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trắng.
- T, N, K, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq K < N \leq 10^3$, $1 \leq A[i] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi bộ test trên một dòng.

Input:	Output:
2	12 10 9
5 3	12 9
10 7 9 12 6	
6 2	
9 7 12 8 6 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

82 Đếm số lần xuất hiện**ĐẾM SỐ LẦN XUẤT HIỆN**

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử đã được sắp xếp. Hãy tìm số lần xuất hiện số X trong mảng. Nếu số lần xuất hiện số x trong mảng là 0 hãy đưa ra -1.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào N và X; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trắng.
- T, N, X, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 10^3$, $1 \leq A[i], X \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
7 2	-1
1 1 2 2 2 2 3	
7 4	
1 1 2 2 2 2 3	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

83 Phần tử đầu tiên lặp lại

Cho mảng A[] gồm N phần tử. Hãy tìm phần tử lặp lại đầu tiên của mảng. Ví dụ với mảng A[] = {5, 6, 1, 2, 1, 4} thì ta có 1 là phần tử đầu tiên lặp lại trong mảng. Nếu không tồn tại đáp án, in ra -1.

Input:

- Đòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào số phần tử của mảng N; dòng tiếp theo là N số A[i] là các phần tử của mảng A[].
- T, N, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 10^6$, $1 \leq A[i] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
5	30
4 5 1 2 1	
6	
10 20 30 30 20 5 7	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

84 Cấu trúc điểm

Viết chương trình khai báo cấu trúc điểm trong không gian hai chiều với hai tọa độ x và y (kiểu số thực double).

```
struct Point{
    double x, y;
}
```

Nhập hai điểm A và B. Tính khoảng cách giữa hai điểm A, B với độ chính xác 4 chữ số sau dấu phẩy.

Input

- Đòng đầu ghi số bộ test, không quá 20.
- Mỗi bộ test có 4 số thực lần lượt là tọa độ của 2 điểm A và B, giá trị tuyệt đối không quá 1000.

Ouput

Với mỗi bộ test, viết ra khoảng cách giữa 2 điểm với 4 chữ số sau dấu phẩy.

Ví dụ

Input	Output
2	5.0000
0 0 0 5	193.0648
0 199 5 6	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    struct Point A, B;
    int t;
    cin>>t;
    while(t--){
        input(A); input(B);
        cout << fixed << setprecision(4) << distance(A,B) << endl;
    }
    return 0;
}
```

85 Cấu trúc thí sinh

CẤU TRÚC THÍ SINH

Bài làm tốt nhất

Viết chương trình khai báo cấu trúc Thí Sinh gồm các thông tin: Họ tên, Điểm môn 1, Điểm môn 2, Điểm môn 3 và Tổng điểm.

Đọc thông tin 1 thí sinh từ bàn phím và in ra màn hình 3 thông tin: Họ tên, Ngày sinh, Tổng điểm.

Input

Gồm 5 dòng lân lượt, mỗi dòng ghi 1 thông tin: Họ tên, Ngày sinh, Điểm môn 1, Điểm môn 2, Điểm môn 3. Họ tên không quá 50 chữ cái, Ngày sinh viết đúng chuẩn dd/mm/yyyy. Các giá trị điểm là số thực (float).

Output

Ghi ra Họ tên, Ngày sinh và Tổng điểm. Mỗi thông tin cách nhau một khoảng trắng. Điểm được ghi ra với 1 số sau dấu phẩy.

Ví dụ

Input	Output
Nguyen Hoang Ha 11/10/2001 4.5 10.0 5.5	Nguyen Hoang Ha 11/10/2001 20.0

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++  
int main(){  
    struct ThiSinh A;  
    nhap(A);  
    in(A);  
    return 0;  
}
```

86 Cấu trúc phân số

CẤU TRÚC PHÂN SỐ

Bài làm tốt nhất

Viết chương trình xây dựng cấu trúc Phân số gồm hai phần tử là tử số và mẫu số. Các giá trị đều nguyên dương và không quá 18 chữ số.

Sau đó thực hiện nhập vào một phân số và in ra phân số đó ở dạng tối giản.

Input

Có hai số nguyên dương lân lượt là tử số và mẫu số.

Output

Ghi ra phân số tối giản như trong ví dụ

Ví dụ

Input	Output
123 456	41/152

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

87 Cấu trúc sinh viên

CẤU TRÚC SINH VIÊN

Bài làm tốt nhất

Viết chương trình khai báo cấu trúc Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float).

Đọc thông tin 1 sinh viên từ bàn phím (không có mã sinh viên) và in ra màn hình. Trong đó Mã SV được gán là **B20DCCN001**. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy.

Input

Gồm 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

Output

Ghi thông tin sinh viên trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trắng.

Ví dụ

Input	Output
Nguyen Hoa Bin D20CQCN04-B 2/2/2002 2	B20DCCN001 Nguyen Hoa Bin D20CQCN04-B 02/02/2002 2.00

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++  
int main(){  
    struct SinhVien a;  
    nhap(a);  
    in(a);  
    return 0;  
}
```

88 Cấu trúc nhân viên

Một nhân viên làm việc trong công ty được lưu lại các thông tin sau:

- Mã nhân viên: được gán giá trị là 00001
- Họ tên: Xâu ký tự không quá 40 chữ cái.
- Giới tính: Nam hoặc Nữ
- Ngày sinh: đúng theo chuẩn dd/mm/yyyy
- Địa chỉ: Xâu ký tự không quá 100 chữ cái
- Mã số thuế: Dãy số có đúng 10 chữ số
- Ngày ký hợp đồng: đúng theo chuẩn dd/mm/yyyy

Viết chương trình nhập một nhân viên (không nhập mã) và in ra màn hình thông tin của nhân viên đó.

Input

Gồm 6 dòng lần lượt ghi các thông tin theo thứ tự đã ghi trong đề bài. Không có mã nhân viên.

Output

Ghi ra đầy đủ thông tin nhân viên trên một dòng, các thông tin cách nhau đúng một khoảng trắng.

Ví dụ

Input
Nguyen Van Hoa Nam 22/11/1982 Mo Lao-Ha Dong-Ha Noi 8333123456 31/12/2013
Output
00001 Nguyen Van Hoa Nam 22/11/1982 Mo Lao-Ha Dong-Ha Noi 8333123456 31/12/2013

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main(){
    struct NhanVien a;
    nhap(a);
    in(a);
    return 0;
}
```

Viết chương trình xây dựng cấu trúc Phân số gồm hai phần tử là tử số và mẫu số. Các giá trị đều nguyên dương và không quá 9 chữ số.

Sau đó thực hiện nhập vào hai phân số p và q. Tính tổng p + q, rút gọn và in ra kết quả.

Input

Có bốn số nguyên dương lần lượt là tử số và mẫu số của p rồi đến q.

Output

Ghi ra phân số tổng p + q ở dạng tối giản như trong ví dụ

Ví dụ

Input	Output
123 456 12 34	1609/2584

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main() {
    struct PhanSo p,q;
    nhap(p); nhap(q);
    PhanSo t = tong(p,q);
    in(t);
    return 0;
}
```

Một nhân viên làm việc trong công ty được lưu lại các thông tin sau:

- Mã nhân viên: được gán tự động tăng, bắt đầu từ 00001
- Họ tên: Xâu ký tự không quá 40 chữ cái.
- Giới tính: Nam hoặc Nữ
- Ngày sinh: đúng theo chuẩn dd/mm/yyyy
- Địa chỉ: Xâu ký tự không quá 100 chữ cái
- Mã số thuế: Dãy số có đúng 10 chữ số
- Ngày ký hợp đồng: đúng theo chuẩn dd/mm/yyyy

Viết chương trình nhập danh sách nhân viên (không nhập mã) và in ra màn hình danh sách nhân viên vừa nhập.

Input

Dòng đầu ghi số N là số nhân viên (không quá 40). Mỗi nhân viên ghi trên 6 dòng lần lượt ghi các thông tin theo thứ tự đã ghi trong đề bài. Không có mã nhân viên.

Output

Ghi ra danh sách đầy đủ nhân viên theo đúng thứ tự nhập, mỗi nhân viên trên một dòng, các thông tin cách nhau đúng một khoảng trắng.

Ví dụ

Input
3
Nguyen Van A
Nam
10/22/1982
Mo Lao-Ha Dong-Ha Noi
8333012345
31/12/2013
Ly Thi B
Nu
10/15/1988
Mo Lao-Ha Dong-Ha Noi
8333012346
22/08/2011
Hoang Thi C
Nu
04/02/1981
Mo Lao-Ha Dong-Ha Noi
8333012347
22/08/2011
Output
00001 Nguyen Van A Nam 10/22/1982 Mo Lao-Ha Dong-Ha Noi 8333012345 31/12/2013
00002 Ly Thi B Nu 10/15/1988 Mo Lao-Ha Dong-Ha Noi 8333012346 22/08/2011
00003 Hoang Thi C Nu 04/02/1981 Mo Lao-Ha Dong-Ha Noi 8333012347 22/08/2011

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main(){
    struct NhanVien ds[50];
    int N,i;
    cin >> N;
    for(i = 0; i < N; i++) nhap(ds[i]);
```

91 Khai báo lớp sinh viên – 1

KHAI BÁO LỚP SINH VIÊN - 1

Bài làm tốt nhất

Viết chương trình khai báo lớp Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float). Hàm khởi tạo không có tham số, gán các giá trị thuộc tính ở trạng thái mặc định (xâu ký tự rỗng, giá trị số bằng 0).
Đọc thông tin 1 sinh viên từ bàn phím (không có mã sinh viên) và in ra màn hình. Trong đó Mã SV được gán là **B20DCCN001**. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy.

Input

Gồm 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

Output

Ghi thông tin sinh viên trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trắng.

Ví dụ

Input	Output
Nguyen Van A D20CQCN04-B 2/2/2002 2	B20DCCN001 Nguyen Van A D20CQCN04-B 02/02/2002 2.00

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main(){
    SinhVien a;
    a.nhap();
    a.xuat();
    return 0;
}
```

92 Khai báo lớp sinh viên – 2

Viết chương trình khai báo lớp Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float). Hàm khởi tạo không có tham số, gán các giá trị thuộc tính ở trạng thái mặc định (xâu ký tự rỗng, giá trị số bằng 0). Yêu cầu sử dụng chòng toán tử nhập và xuất để nhập đổi tượng sinh viên với cin và in ra đổi tượng sinh viên với cout. Đọc thông tin 1 sinh viên từ bàn phím (không có mã sinh viên) và in ra màn hình. Trong đó Mã SV được gán là **B20DCCN001**. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy.

Input

Gồm 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

Output

Ghi thông tin sinh viên trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trắng.

Ví dụ

Input	Output
Nguyen Hoa Bin D20CQCN04-B 2/2/2002 2	B20DCCN001 Nguyen Hoa Bin D20CQCN04-B 02/02/2002 2.00

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    SinhVien a;
    cin >> a;
    cout << a;
    return 0;
}
```

93 Khai báo lớp sinh viên – 3

Viết chương trình khai báo lớp Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float). Hàm khởi tạo không có tham số, gán các giá trị thuộc tính ở trạng thái mặc định (xâu ký tự rỗng, giá trị số bằng 0).

Yêu cầu sử dụng chèn toán tử nhập và xuất để nhập đổi tượng sinh viên với cin và in ra đổi tượng sinh viên với cout.

Đọc thông tin 1 sinh viên từ bàn phím (không có mã sinh viên) và in ra màn hình. Trong đó Mã SV được gán là **B20DCCN001**. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy. Tên được đưa về dạng chuẩn.

Input

Gồm 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

Output

Ghi thông tin sinh viên trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trắng.

Ví dụ

Input	Output
NguyEn hoa BiNH D20CQCN04-B 2/2/2002 2	B20DCCN001 Nguyen Hoa Binh D20CQCN04-B 02/02/2002 2.00

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

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

94 Khai báo lớp phân số

Viết chương trình xây dựng class Phân số.

Sau đó thực hiện nhập vào một phân số và in ra phân số đó ở dạng tối giản.

Input

Có hai số nguyên dương lần lượt là tử số và mẫu số. Các giá trị không quá 18 chữ số.

Output

Ghi ra phân số tối giản như trong ví dụ

Ví dụ

Input	Output
123 456	41/152

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main() {
    PhanSo p(1,1);
    cin >> p;
    p.rutgon();
    cout << p;
    return 0;
}
```

95 Khai báo lớp nhân viên

KHAI BÁO LỚP NHÂN VIÊN

Bài làm tốt nhất

Một nhân viên làm việc trong công ty được lưu lại các thông tin sau:

- Mã nhân viên: được gán giá trị là 00001
- Họ tên: Xâu ký tự không quá 40 chữ cái.
- Giới tính: Nam hoặc Nữ
- Ngày sinh: đúng theo chuẩn mm/dd/yyyy
- Địa chỉ: Xâu ký tự không quá 100 chữ cái
- Mã số thuế: Dãy số có đúng 10 chữ số
- Ngày ký hợp đồng: đúng theo chuẩn dd/mm/yyyy

Viết chương trình nhập một nhân viên (không nhập mã) trong đó có sử dụng chòng toán tử nhập/xuất và in ra màn hình thông tin của nhân viên đó.

Input

Gồm 6 dòng lần lượt ghi các thông tin theo thứ tự đã ghi trong đề bài. Không có mã nhân viên.

Output

Ghi ra đầy đủ thông tin nhân viên trên một dòng, các thông tin cách nhau đúng một khoảng trống.

Ví dụ

Input
Nguyen Van Hoa
Nam
11/22/1982
Mo Lao-Ha Dong-Ha Noi
8333123456
31/12/2013
Output
00001 Nguyen Van Hoa Nam 11/22/1982 Mo Lao-Ha Dong-Ha Noi 8333123456 31/12/2013

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

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

96 Tính tổng hai đối tượng phân số

Viết chương trình xây dựng class Phân số.

Sau đó thực hiện nhập vào hai phân số p và q. Tính tổng p + q, rút gọn và in ra kết quả.

Input

Có bốn số nguyên dương lần lượt là tử số và mẫu số của p rồi đến q. Các giá trị không quá 9 chữ số.

Output

Ghi ra phân số tổng p + q ở dạng tối giản như trong ví dụ

Ví dụ

Input	Output
123 456 12 34	1609/2584

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main() {
    PhanSo p(1,1), q(1,1);
    cin >> p >> q;
    cout << p + q;
    return 0;
}
```

97 Liệt kê xâu nhị phân

Cho số tự nhiên N. Hãy đưa ra các xâu nhị phân có độ dài N.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số N được viết trên 1 dòng.
- T, N thỏa mãn ràng buộc : $1 \leq T, N \leq 20$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	00 01 10 11
2	000 001 010 011 100 101 110 111
3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

98 Hoán vị liền kề phía trước

Cho số tự nhiên N và một hoán vị $X[]$ của $1, 2, \dots, N$. Nhiệm vụ của bạn là đưa ra hoán vị trước đó của $X[]$. Ví dụ $N=5, X[] = \{1, 2, 3, 4, 5\}$ thì hoán vị trước đó của $X[]$ là $\{5, 4, 3, 2, 1\}$.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là số N; dòng tiếp theo đưa vào hoán vị $X[]$ của $1, 2, \dots, N$.
- $T, N, X[]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq N \leq 10^3$.
- Input đảm bảo không có trường hợp hoán vị đã cho là đầu tiên (tức là luôn có hoán vị trước nó)

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ví dụ:

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

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

99 Tích chữ số các chữ số

Axe chơi một trò chơi với Lina.

Họ định nghĩa hàm $F(x)$ với số x nguyên dương là tích giải thừa các chữ số của x .

Ví dụ $F(135) = 1! * 3! * 5! = 720$.

Đầu tiên, họ chọn một số a có n chữ số và có ít nhất một chữ số lớn hơn 1, có thể có chữ số không ở đầu. Sau đó họ tìm một số nguyên dương x lớn nhất thỏa mãn:

1. X không chứa chữ số 0 hoặc 1
2. $F(x) = F(a)$

Hãy giúp Axe và Lina tìm ra được số đó.

Input

Dòng đầu tiên chứa số bộ test T ($T < 100$).

Mỗi test gồm một dòng chứa số n và số a ($1 \leq n \leq 15$).

Output

In ra kết quả mỗi test trên một dòng là số lớn nhất tìm được.

Example

Input:

1

4 1234

Output:

33222

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

100 Tính lũy thừa

Cho ba số nguyên dương x, y, p . Nhiệm vụ của bạn là tính $(x^y) \% p$. Ví dụ với $x = 2, y = 3, p = 5$ thì $(2^3) \% 5 = 3$.

Input:

- Dòng đầu tiên đưa vào số lượng test T .
- Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là bộ ba x, y, p được viết cách nhau một vài khoảng trắng.
- T, x, y, p thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq x, y \leq 10^6; 1 \leq p \leq 10^9 + 7$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

101. Đảo từ

ĐÀO TỪ

Bài làm tốt nhất

Cho xâu ký tự S. Nhiệm vụ của bạn là đảo ngược các từ trong S. Ví dụ S = "I like this program very much", ta nhận được kết quả là "much very program this like I".

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S.
- T, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{Length}(S) \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 I like this program very much much very program this like I	much very program this like I I like this program very much

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

102. Tính số Fibonacci lớn

TÍNH SỐ FIBONACCI LỚN

Bài làm tốt nhất

Dãy số Fibonacci được định nghĩa $F_n = F_{n-1} + F_{n-2}$, $n \geq 1$ và $F_0 = 0$, $F_1 = 1$. Dưới đây là một số số Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21...

Nhiệm vụ của bạn là tìm số Fibonacci thứ n.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên dương n.
- T, n thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 1000$.

Output:

- Đưa ra kết quả mỗi test theo modulo $10^9 + 7$ theo từng dòng.

Input	Output
2	1
2	5
5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

103. Sao chép tệp tin

Cho tệp dữ liệu đầu vào có tên: PTIT.in đặt cùng thư mục với tệp mã nguồn.

Nhiệm vụ của bạn là sao chép nội dung tệp PTIT.in tới tệp PTIT.out

Ví dụ:

PTIT.in	PTIT.out
Ngon ngu lap trinh C++	Ngon ngu lap trinh C++

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

104. Cấu trúc sinh viên

Viết chương trình khai báo cấu trúc Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float).

Đọc thông tin 1 sinh viên từ bàn phím (không có mã sinh viên) và in ra màn hình. Trong đó Mã SV được gán là N20DCCN001.

Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy.

Input

Gồm 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với nhiều nhất 2 số sau dấu phẩy.

Output

Ghi thông tin sinh viên trên 1 dòng, mỗi thông tin cách nhau 1 Tab.

Ví dụ

Input	Output
Nguyen Van A D20CQCN01-N 2/2/2000 2.5	N20DCCN001 Nguyen Van A D20CQCN01-N 02/02/2000 2.50

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main(){
    struct SinhVien a;
    nhapThongTinSV(a);
    inThongTinSV(a);
    return 0;
}
```

105. PTIT25R1C – Tổng hiệu

Cho dãy số $A[]$ có N phần tử. Hãy tính tổng hiệu $|A[i] - A[j]|$ với mọi cặp (i, j) thỏa mãn $1 \leq i < j \leq N$. Hãy giá trị biểu thức:

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^N |A_i - A_j|$$

Input:

Dòng đầu tiên là số lượng phần tử N ($2 \leq N \leq 200000$).

Dòng tiếp theo gồm N số nguyên $A[i]$, các số có giá trị tuyệt đối không vượt quá 10^9 .

Output:

In ra một số nguyên là giá trị biểu thức tìm được.

Test ví dụ:

Input:	Output:
3	8
5 1 2	
5 1 2 3 4 5	20

Giải thích test 1: $|5 - 1| + |5 - 2| + |1 - 2| = 8$.

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

106. PTIT25R1D – Xóa đi một kí tự

Cho xâu S có độ dài N . Kí hiệu S_i là xâu S còn lại sau khi xóa bỏ kí tự thứ i .

Yêu cầu: Hãy đếm số cặp chỉ số (i, j) với $1 \leq i < j \leq N$ và $S_i = S_j$.

Input:

Dòng đầu tiên là số nguyên dương N ($2 \leq N \leq 300000$).

Dòng tiếp theo là xâu S gồm N kí tự.

Output:

In ra một số nguyên là số cặp chỉ số (i, j) thỏa mãn yêu cầu.

Test ví dụ:

Input:	Output:
7 abbbcca	4
4 aaaa	6
2 ab	0

Giải thích test 1: Các xâu thu được sau khi xóa bỏ 1 kí tự lần lượt là: bbbbca, abbc, abbc, abbc, abbc, abbc, abbc.

Có 4 cặp chỉ số thỏa mãn là $(2, 3), (2, 4), (3, 4)$ và $(5, 6)$.

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

107. PTIT25R1E – Số gần đối xứng

Một số được gọi là gần palindrome (hay số gần đối xứng) nếu trong dạng biểu diễn ở hệ 10 chỉ cần thay đổi không quá một chữ số ta sẽ được dãy ký tự palindrome. Ví dụ, các số 1234321, 1234311, 123421 là những số gần palindrome. Còn các số 1234213 và 12345331 – không phải là gần palindrome.

Yêu cầu: Cho hai số nguyên dương A và B. Hãy xác định số lượng các số là số gần palindrome trong đoạn [A B].

Input:

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

Mỗi test gồm hai số nguyên dương A và B ($1 \leq A \leq B \leq 1000000$).

Output:

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

Test ví dụ:

Input:	Output:
3	10
1 10	1000
1 1000	7601
10000 50000	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

108. Bài class BigNumber (offline) (Lỗi)

109. Số tăng giảm

SỐ TĂNG GIẢM

Bài làm tốt nhất

Một số được gọi là số tăng giảm nếu số đó có các chữ số thỏa mãn hoặc tăng dần, hoặc giảm dần từ trái qua phải. Hãy đếm các số nguyên tố là số tăng giảm với **số chữ số cho trước**.

Dữ liệu vào: Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng số chữ số tương ứng cần kiểm tra (lớn hơn 1 và nhỏ hơn 10)

Kết quả: Ghi ra số lượng các số thỏa mãn điều kiện.

Input	Output
2	20
2	50
4	

Giới hạn thời gian: 4s

Giới hạn bộ nhớ: 65536 Kb

110. Ước số nguyên tố nhỏ nhất

ƯỚC SỐ NGUYÊN TỐ NHỎ NHẤT

Bài làm tốt nhất

Cho số tự nhiên N. Nhiệm vụ của bạn là in ra ước số nguyên tố nhỏ nhất của các số từ 1 đến N. Trong bài toán này, ta coi ước số nguyên tố nhỏ nhất của 1 là 1. Ước số nguyên tố nhỏ nhất của các số chẵn là 2. Ước số nguyên tố nhỏ nhất của các số nguyên tố là chính nó.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số N được ghi trên một dòng.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 10000$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2 3 2 5 2
6	1 2 3 2 5 2 7 2 3 2
10	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

111. Số Sphenic

SỐ SPHENIC

Bài làm tốt nhất

Số nguyên dương N được gọi là số Sphenic nếu N được phân tích duy nhất dưới dạng tích của ba số khác nhau. Ví dụ N=30 là số Sphenic vì $30 = 2 \times 3 \times 5$; N = 60 không phải số Sphenic vì $60 = 2 \times 2 \times 3 \times 5$. Cho số tự nhiên N, nhiệm vụ của bạn là kiểm tra xem N có phải số Spheic hay không?

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên dương N.
- T, N thỏa mãn ràng buộc : $1 \leq T \leq 100$; $1 \leq N \leq 10000$.

Output:

- Đưa ra 1 hoặc 0 tương ứng với N là số Sphenic hoặc không của mỗi test theo từng dòng.

Input:	Output:
2	1
30	0
60	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

112. Ước số chia hết cho 2

ƯỚC SỐ CHIA HẾT CHO 2

Bài làm tốt nhất

Cho số nguyên dương N.

Nhiệm vụ của bạn là hãy xác định xem có bao nhiêu ước số của N chia hết cho 2?

Input:

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

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

Output:

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

Ví dụ:

Input:	Output:
2	0
9	3
8	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

113. Chia hết cho A và B

CHIA HẾT CHO A VÀ B

Bài làm tốt nhất

Cho bốn số M, N, A, B. Nhiệm vụ của bạn là hãy đếm tất cả các số trong khoảng [M, N] chia hết cho các số A, B.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 4 số M, N, A, B.
- T, M, N, A, B thỏa mãn ràng buộc $1 \leq T \leq 100$; $1 \leq M, N \leq 10^5$; $1 \leq A, B \leq 500$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
5 11 4 6	289
3 1000 5 9	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

114. Ước số nguyên tố nhỏ nhất

ƯỚC SỐ NGUYỄN TỐ NHỎ NHẤT

Bài làm tốt nhất

Cho số tự nhiên N. Nhiệm vụ của bạn là hãy đưa ra ước số nguyên tố nhỏ nhất của các số từ 1 đến N.

Ví dụ n=10, ta có được kết quả : 1 2 3 2 5 2 7 2 3 2.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số N.
- T, N thỏa mãn ràng buộc $1 \leq T \leq 100$; $1 \leq N \leq 10^9$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2 3 2 5
5	1 2 3 2 5 2 7 2 3 2
10	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

115. Đếm số có ba ước số

ĐẾM SỐ CÓ BA ƯỚC SỐ

Bài làm tốt nhất

Cho số tự nhiên N. Nhiệm vụ của bạn là hãy đếm tất cả các số từ 1 tới N mà có đúng ba ước số. Ví dụ N=100, ta có các số 4, 9, 25, 49.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số N.
- T, N thỏa mãn ràng buộc $1 \leq T \leq 100$; $1 \leq N \leq 10^{12}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
50	6
200	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

116. Kiểm tra số Fibonacci

Cho số nguyên dương n. Hãy kiểm tra xem n có phải là số Fibonacci hay không?

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên dương n.
- T, n thỏa mãn ràng buộc : $1 \leq T \leq 100; 0 \leq n \leq 10^{18}$.

Output:

- Đưa ra "YES" hoặc "NO" tương ứng với n là số Fibonacci hoặc không phải số Fibonacci của mỗi test theo từng dòng.

Input	Output
2	YES
8	NO
15	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

117. Nguyên tố cùng nhau

Juggernaut được cô giáo Disruptor dạy toán, cô giáo định nghĩa một hàm $f(x)$ như sau:

Với t là số lượng các số tự nhiên k ($1 \leq k \leq x$) thỏa mãn nguyên tố cùng nhau với x , nếu t là nguyên tố thì $f(x) = 1$, ngược lại $f(x) = 0$.

Disruptor cho Juggernaut một số nguyên dương x , yêu cầu anh cho biết giá trị của hàm $f(x)$, nếu trả lời sai thì Jug sẽ bị cô trả về nhà, Jug không muốn về nhà, các bạn hãy giúp Jug giải bài toán này.

Input

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

Mỗi test gồm một dòng chứa số x ($1 \leq x \leq 10^5$).

Output

In ra kết quả mỗi test trên một dòng là giá trị của hàm $f(x)$.

Example

Input:

2
2
3

Output:

0
1

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

118. Chia hết cho ba số nguyên

Cho bốn số nguyên dương X, Y, Z và N. Hãy tìm số nguyên dương nhỏ nhất có N chữ số chia hết đồng thời cho X, Y, Z. Ví dụ với

$X = 2, Y = 3, Z = 5, N = 4$ ta tìm được số nguyên dương nhỏ nhất có 4 chữ số là 1020 chia hết cho cả 2, 3, 5.

Input:

- Dòng đầu tiên đưa vào số lượng test T ($T \leq 100$).
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là bộ bốn số X, Y, Z, N. Các số X, Y, Z, N thỏa mãn ràng buộc dưới đây:

$1 \leq X, Y, Z \leq 10^5; N \leq 18$.

Output:

- Đưa ra theo từng dòng kết quả mỗi test là số nguyên nhỏ nhất có N chữ số chia hết đồng thời cho X, Y, Z. Trong trường hợp không có số nguyên N chữ số thỏa mãn yêu cầu bài toán đưa ra giá trị -1.

Input	Output
3	1020
2 3 5 4	120
4 5 6 3	-1
3 5 7 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

119. Số lặp lại

SỐ LẶP LẠI

Bài làm tốt nhất

Cho ba số a, x, y. Nhiệm vụ của bạn là tìm ước số chung lớn nhất của hai số P và Q, trong đó P lặp lại x lần số a và Q lặp lại y lần số a. Ví dụ a = 2, x = 3, y = 2 thì P=222, Q=22.

Input:

- Dòng đầu tiên đưa vào T là số lượng bộ test.
- T dòng tiếp đưa các bộ test. Mỗi bộ test được viết trên một dòng là bộ ba số a, x, y phân biệt nhau bởi một vài khoảng trắng.
- Các số T, a, x, y thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq a, x, y \leq 10^{18}$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	2
2 2 3	123
123 5 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

120. Đếm số nguyên tố

ĐẾM SỐ NGUYÊN TỐ

Bài làm tốt nhất

Cho một câu hỏi Q là bộ đôi hai số L và R. Nhiệm vụ của bạn là xác định xem có bao nhiêu số nguyên tố trong khoảng [L, R]. Ví dụ với Q = [1, 10] ta có câu trả lời là 4 vì có {2, 3, 5, 7} là các số nguyên tố.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một bộ đôi L, R. các số được viết cách nhau một vài khoảng trắng.
- T, L, R thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq L \leq R \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
1 10	2
5 10	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

121. Hiệu lớn nhất của cặp phần tử đúng thứ tự

Cho mảng A[] gồm n số nguyên. Hãy tìm hiệu lớn nhất của bất kể hai phần tử nào của mảng dây con thỏa mãn ràng buộc số lớn hơn xuất hiện sau số nhỏ hơn. Nếu không tìm được cặp phần tử của mảng hãy đưa ra -1. Ví dụ với mảng A[] = {2, 3, 10, 6, 4, 8, 1} ta nhận được kết quả là 8 = 10-2. Với mảng A[] = {7, 9, 5, 6, 3, 2} ta có kết quả là 2 = 9-7.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	8
7	-1
2 3 10 6 4 8 1	
3	
3 2 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

122. Tính giá trị đa thức

Tính toán giá trị đa thức $P(n, x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_0$.

Kết quả có thể rất lớn nên chia dư cho $10^9 + 7$

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test gồm hai dòng: dòng thứ nhất đưa vào hai số n, x; dòng tiếp theo đưa vào n số $a_{n-1}, a_{n-2}, \dots, a_0$ là hệ số của đa thức P. Các số được viết cách nhau một vài khoảng trống.
- T, n, x, P[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq n \leq 2000$; $0 \leq x, P[i] \leq 1000$.
- **Chú ý:**Các hệ số của đa thức P được viết theo thứ tự từ bậc 0 đến bậc n-1

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	20
4 2	
1 2 0 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

123. Kiểm tra dãy Fibonacci

KIỂM TRA DÃY FIBONACCI

Bài làm tốt nhất

Cho mảng A[] gồm n số nguyên không âm. Hãy tìm dãy con lớn nhất chỉ toàn các số Fibonacci. Số 0 được coi là số Fibonacci đầu tiên.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào n là số phần tử của mảng A[]; dòng tiếp theo đưa vào n số các phần tử của mảng A[]; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 100$; $1 \leq A[i] \leq 1000$.

Output:

- Đưa ra dãy con lớn nhất bao gồm các số Fibonacci của mỗi test theo từng dòng.

Input	Output
2	1 3 13
7	0 2 8 5 2 1 13
1 4 3 9 10 13 7	
9	
0 2 8 5 2 1 4 13 23	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

124. Dãy Mountain

DÃY MOUNTAIN

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử và một câu hỏi Q. Mỗi câu hỏi Q là bộ đôi hai số L và R. Nhiệm vụ của bạn là xác định xem dãy con của A[] trong khoảng [L, R] có tạo nên một dãy Mountain hay không? Dãy Mountain là dãy được chia thành hai phần: phần thứ nhất tăng phần thứ hai giảm. Ví dụ với mảng A[] = {2, 3, 2, 4, 4, 6, 3, 2}, các câu hỏi Q: [0, 2], [2, 7], [2, 3], [1, 3] ta sẽ có các câu trả lời: Yes, Yes, Yes, No tương ứng với các dãy Mountain [2, 3, 2], [2, 4, 4, 6, 3, 2], [2, 4], [3, 2, 4].

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào n, L, R là số phần tử của mảng A[] và câu hỏi Q; phần tiếp theo đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trắng.
- T, n, L, R, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq L \leq R \leq n \leq 10^4$; $1 \leq A[i] \leq 10^3$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	Yes
8	No
2 3 2 4 4 6 3 2	
0 2	
8	
2 3 2 4 4 6 3 2	
1 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

125. Phần tử nhỏ nhất thứ k của ma trận

PHẦN TỬ NHỎ NHẤT THỨ K CỦA MA TRẬN

Bài làm tốt nhất

Cho ma trận vuông $A[][]$ cấp n . Các phần tử của ma trận $A[][]$ đã được sắp xếp theo hàng, cột. Hãy tìm phần tử nhỏ nhất thứ k của ma trận. Ví dụ với ma trận cấp 4 dưới đây sẽ cho ta số nhỏ nhất thứ 3 là 20, số nhỏ nhất thứ 7 là 30.

10 20 30 40
15 25 35 45
24 29 37 48
32 33 39 50

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là n và k ; phần thứ hai là n^2 các phần tử của ma trận vuông $A[][]$; các số được viết cách nhau một vài khoảng trống.
- $T, n, k, A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq n \leq 100; 1 \leq k, A[i][j] \leq 10^4$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 4 7 10 20 30 40 15 25 35 45 24 29 37 48 32 33 39 50	30

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

126. Quay ma trận

QUAY MA TRẬN

Bài làm tốt nhất

Cho ma trận $A[][]$ gồm các số nguyên dương. Nhiệm vụ của bạn là quay ma trận theo chiều kim đồng hồ. Ví dụ về quay theo chiều kim đồng hồ ma trận $A[][]$ dưới đây.

1 2 3 4 1 2
4 5 6 7 5 3
7 8 9 8 9 6

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n , m tương ứng với số hàng, số cột của ma trận $A[]$; dòng tiếp theo đưa vào $n \times m$ số $A[i][j]$; các số được viết cách nhau một vài khoảng trống.
- $T, n, m, A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq n, m \leq 100; 1 \leq A[i][j] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 2 2 1 2 5 6 3 3 1 2 3 4 5 6 7 8 9	5 1 6 2 4 1 2 7 5 3 8 9 6

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

127. Đếm phần tử giống nhau

Cho ma trận vuông $A[][]$ cấp N . Nhiệm vụ của bạn là đưa ra số các phần tử giống nhau ở tất cả các hàng. Ví dụ với ma trận $A[][]$ dưới đây sẽ cho ta kết quả là 2 tương ứng với số 2, 3 xuất hiện ở tất cả các hàng.

2	1	4	3
1	2	3	2
3	6	2	3
5	2	5	3

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N là cấp của ma trận $A[][]$; dòng tiếp theo đưa vào $N \times N$ số $A[i][j]$; các số được viết cách nhau một vài khoảng trắng.
- $T, N, A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 100$; $1 \leq A[i][j] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

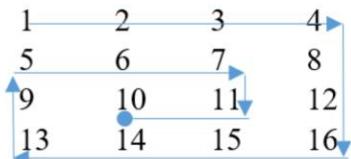
Input:	Output:
1 4 2 1 4 3 1 2 3 2 3 6 2 3 5 2 5 3	2

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

128. Ma trận xoắn ốc – 1

Cho ma trận $A[N][M]$. Nhiệm vụ của bạn là in các phần tử của ma trận theo hình xoắn ốc. Ví dụ về in ma trận theo hình xoắn ốc như dưới đây: 1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10 .



Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N, M là cấp của ma trận $A[][]$; dòng tiếp theo đưa vào $N \times M$ số $A[i][j]$; các số được viết cách nhau một vài khoảng trắng.
- $T, M, N, A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq M, N \leq 100$; $1 \leq A[i][j] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 4 4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 3 4 1 2 3 4 5 6 7 8 9 10 11 12	1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10 1 2 3 4 8 12 11 10 9 5 6 7

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

129. Đếm số miền ma trận

Cho ma trận $A[N][M]$ bao gồm các số 0 và 1. Ta gọi mỗi miền của ma trận $A[][],$ là nhóm các số 1 được bao quanh bởi các số 0.

Hãy tìm số miền của ma trận. Ví dụ số miền của ma trận $A[][],$ là 4.

```
{ {1, 1, 0, 0, 0},  
{0, 1, 0, 0, 1},  
{1, 0, 0, 1, 1},  
{0, 0, 0, 0, 0},  
{1, 0, 1, 1, 0} }
```

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N, M là cấp của ma trận $A[][],$ dòng tiếp theo đưa vào $N \times M$ số $A[i][j]$; các số được viết cách nhau một vài khoảng trắng.
- $T, M, N, A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq M, N \leq 100; 0 \leq A[i][j] \leq 1.$

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

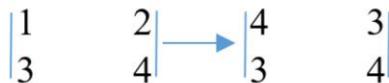
Input:	Output:
2 3 3 1 1 0 0 0 1 1 0 1 4 4 1 1 0 0 0 1 0 0 0 0 1 0 1 0 0	2 2

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

130. Biến đổi ma trận

Cho ma trận vuông $A[N][N]$. Tim số phép biến đổi ít nhất để tông theo các hàng, các cột của ma trận đều bằng nhau. Biết mỗi phép biến đổi bạn chỉ được phép tăng một phần tử bất kỳ của ma trận lên 1 đơn vị. Ví dụ với ma trận



Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N là cấp của ma trận $A[N][N];$ dòng tiếp theo đưa vào $N \times N$ số $A[i][j]$; các số được viết cách nhau một vài khoảng trắng.
- $T, N, A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq N \leq 100; 1 \leq A[i][j] \leq 150.$

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 2 1 2 3 4 3 1 2 3 4 2 3 3 2 1	4 6

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

131 HÌNH CHỮ NHẬT LỚN NHẤT - 1

HÌNH CHỮ NHẬT LỚN NHẤT - 1

Bài làm tốt nhất

Cho ma trận $A[N][M]$ chỉ bao gồm các số 0 và 1. Hãy tìm hình chữ nhật lớn nhất có các phần tử đều bằng 1 bằng cách tráo đổi các cột của ma trận với nhau. Ví dụ với ma trận dưới đây ta sẽ có hình chữ nhật lớn nhất có các phần tử là 1 bằng 6.

0	1	0	1	0	0	0	1	1	0
0	1	0	1	1	0	0	1	1	1
1	1	0	1	0	1	0	1	1	0

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: Dòng đầu tiên đưa vào hai số N, M ; dòng tiếp là $N \times M$ các phần tử của ma trận $A[][]$; các phần tử được viết cách nhau một vài khoảng trắng.
- T, N, M thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N, M \leq 15$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
2 3	2
1 1 1	
0 1 1	
2 2	
1 0	
1 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

132 BIẾN ĐỔI DÃY SỐ

BIẾN ĐỔI DÃY SỐ

Bài làm tốt nhất

Cho mảng $A[]$ gồm n số nguyên dương. Hãy biến mảng $A[]$ thành một mảng đổi xứng sao cho phép thay thế $A[i] = \text{Merge}(A[i], A[i+1])$, $A[i+1]$ được thực hiện ít nhất. Trong đó, $\text{Merge}(A[i], A[i+1]) = A[i] + A[i+1]$. Ví dụ với $A[] = \{3, 2, 3, 3, 5\}$ ta chỉ cần thực hiện 1 phép $\text{Merge}(A[0], A[1])$ để trả thành mảng $A[] = \{5, 3, 3, 5\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n; dòng tiếp theo là n số $A[i]$ của mảng $A[]$; các số được viết cách nhau một vài khoảng trắng.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A[i] \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

133 DÃY SỐ NHỊ PHÂN

DÃY SỐ NHỊ PHÂN

Bài làm tốt nhất

Cho mảng các số nhị phân $A1[]$ và $A2[]$ gồm n 0, 1. Hãy tìm khoảng chung dài nhất thỏa mãn: $j \geq i$ và $\text{span}(i, j) = A1[i] + A1[i+1] + \dots + A1[j] = A2[i] + A2[i+1] + \dots + A2[j]$. Ví dụ với $A1[] = \{0, 1, 0, 0, 0, 0\}$, $A2[] = \{1, 0, 1, 0, 0, 1\}$ ta có kết quả là 4 tương ứng với $A1[1]+A1[2]+A1[3]+A1[4] = A2[1]+A2[2]+A2[3]+A2[4] = 1$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên là số phần tử của mảng n ; dòng tiếp theo là n số $A1[i]$ của mảng $A1[]$; dòng tiếp theo là n số $A2[i]$ của mảng $A2[]$; các số được viết cách nhau một vài khoảng trắng.
- T, n thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 6 0 1 0 0 0 0 1 0 1 0 0 1	4

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

134 MA TRẬN XOẮN ỐC - 4

MA TRẬN XOẮN ỐC - 4

Bài làm tốt nhất

Cho ma trận vuông A cỡ $N \times N$ chỉ bao gồm các số nguyên dương không quá 1000. Hãy sắp đặt các giá trị trong ma trận A sao cho các số được điền lần lượt theo kiểu xoắn ốc tăng dần, theo chiều kim đồng hồ.

Input

Dòng đầu ghi số N ($2 < N < 20$).

N dòng tiếp theo ghi ma trận A , các giá trị nguyên dương và không quá 1000.

Output

Ghi ra ma trận kết quả

Ví dụ

Input	Output
3	1 3 4
3 6 1	9 12 5
8 7 9	8 7 6
4 12 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

135 Cửa sổ trượt

Cho ma trận vuông A cỡ $N \times N$. Một ma trận vuông B nhỏ hơn cỡ $M \times M$ có thể dùng làm “cửa sổ trượt” trên ma trận A nếu M là ước số của N .

Hãy thực hiện tính tích chập của ma trận B với từng “khung cửa sổ” tương ứng trên ma trận A. Tích chập được hiểu là tính giá trị tích từng vị trí tương ứng trên 2 ma trận kích thước bằng nhau.

Xem ví dụ để hiểu rõ hơn.

Input

Dòng đầu ghi số N ($3 < N < 100$). Tiếp theo là N dòng ghi ma trận A. Các giá trị đều nguyên dương và không quá 1000.

Tiếp theo là một dòng ghi số M ($1 < M < 10$). Tiếp theo là M dòng ghi ma trận B. Các giá trị lớn hơn hoặc bằng 0 và không quá 20.

Dữ liệu vào đảm bảo M là ước số của N .

Output

Ghi ra N dòng mô tả ma trận kết quả.

Ví dụ

Input	Output
4	1 0 3 0
1 2 3 4	0 12 0 16
5 6 7 8	9 0 11 0
9 10 11 12	0 28 0 32
13 14 15 16	
2	
1 0	
0 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

136 Điểm cân bằng

ĐIỂM CÂN BẰNG

Bài làm tốt nhất

Cho dãy số $A[]$ gồm có N phần tử nguyên dương. Phần tử thứ i được gọi là điểm cân bằng của dãy số nếu như tổng các số bên trái bằng tổng các số bên phải của nó.

Nhiệm vụ của bạn là điểm cân bằng đầu tiên của dãy $A[]$ cho trước. Nếu không có đáp án, in ra -1.

Input:

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

Mỗi test gồm số nguyên N ($1 \leq N \leq 100\,000$), số lượng phần tử trong dãy số ban đầu.

Dòng tiếp theo gồm N số nguyên $A[i]$ ($-1000 \leq A[i] \leq 1000$).

Output:

Với mỗi test, in ra trên một dòng vị trí của điểm cân bằng tìm được.

Ví dụ:

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

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

137 Hiệu của hai tập từ

HIỆU CỦA HAI TẬP TỬ

Bài làm tốt nhất

Cho hai xâu ký tự S1 và S2. Hãy viết chương trình tìm các từ chỉ xuất hiện trong S1 mà không xuất hiện trong S2. Chú ý: mỗi từ chỉ liệt kê 1 lần.

Dữ liệu vào: Dòng 1 ghi số bộ test. Mỗi bộ test gồm 2 dòng, mỗi dòng ghi một xâu ký tự độ dài không quá 200, chỉ bao gồm các ký tự viết thường và các khoảng trắng.

Kết quả: Với mỗi bộ test ghi ra các từ có trong S1 mà không có trong S2. Các từ được ghi theo thứ tự từ điển.

Ví dụ:

Input	Output
2	abcd
abc ab ab ab abcd	aaa ab abc zzz
ab abc	
aaa xyz ab zzz abc dd dd abc	
xyz dd ttt sas cdc	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

138 Tổng lớn nhất và nhỏ nhất

TỔNG LỚN NHẤT VÀ NHỎ NHẤT

Bài làm tốt nhất

Cho hai số nguyên dương X1, X2. Ta chỉ được phép thay đổi chữ số 5 thành 6 và ngược lại chữ số 6 thành chữ số 5 của các số X1 và X2. Hãy đưa ra tổng nhỏ nhất và tổng lớn nhất các số X1 và X2 được tạo ra theo nguyên tắc kể trên.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là cặp các số X1, X2.
- T, X1, X2 thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq X1, X2 \leq 10^{18}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1100 1312
645 666	10010 11132
5466 4555	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

139 Sắp đặt xâu ký tự – 1

Cho xâu ký tự S bao gồm các ký tự 'a'...,'z'. Các ký tự trong S có thể lặp lại. Nhiệm vụ của bạn sắp đặt lại các ký tự trong S sao cho các ký tự kề nhau đều khác nhau.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S.
- T, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq \text{Length}(S) \leq 10^3$.

Output:

- Đưa ra 1 hoặc 0 nếu có thể hoặc không thể sắp đặt lại các ký tự trong xâu S thỏa mãn yêu cầu bài toán.

Input:	Output:
3	1
geeksforgeeks	1
bbbabaacd	0
bbbbbb	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

140 Đổi chỗ chữ số

Cho số tự nhiên N. Bạn chỉ được phép sử dụng nhiều nhất một phép đổi chỗ giữa 2 chữ số để nhận được số lớn nhất nhỏ hơn N. Ví dụ với số N=12345, sử dụng một phép đổi chỗ ta nhận được số lớn nhất nhỏ hơn N là 12345. Mặc dù $12354 > 12345$ nhưng ta không thể tạo ra số 12345 với chỉ một phép hoán vị. Với số N=12345 ta không có phép đổi chỗ.

Input:

- Dòng đầu tiên đưa vào T là số lượng bộ test.
- T dòng tiếp đưa các bộ test. Mỗi bộ test được viết trên một dòng là một xâu ký tự số không có ký tự '0' đầu tiên.
- Các số T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{length}(N) \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng, trong đó -1 được xem là test không có phép đổi chỗ.

Input	Output
2	12345
12435	-1
12345	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

141 Số may mắn – 2

Một số nguyên không âm n được gọi là số may mắn nếu tổng các chữ của n bằng 9 hoặc tổng các chữ số của n là số may mắn.

Ví dụ các số 9, 108, 279 là các số may mắn, còn các số 19, 289 không phải là số may mắn.

Yêu cầu: Cho số nguyên không âm n , hãy kiểm tra xem n có phải là số may mắn hay không?

Input

Dữ liệu vào gồm nhiều bộ dữ liệu tương ứng với nhiều test. Dòng đầu tiên chứa một số nguyên dương không vượt quá 100 là số lượng các bộ dữ liệu. Các dòng tiếp theo chứa các bộ dữ liệu.

Mỗi bộ dữ liệu gồm một dòng duy nhất chứa một số nguyên không âm n ($n \leq 10^{100}$).

Output

Với mỗi bộ dữ liệu, ghi ra trên một dòng câu trả lời, ghi số 1 nếu n là số may mắn, ghi số 0 trong trường hợp ngược lại.

Example

Input:

```
3
888
666
289
```

Output:

```
0
1
0
```

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

142 Biển số đẹp

Biển số xe máy được quy định gồm các thành phần:

- Hai chữ số đầu là mã quản lý theo tỉnh – thành phố (ví dụ mã của Hà Nội là 29 đến 31)
- Sau đó là một chữ cái in hoa (từ A đến Z) và một chữ số. Cặp chữ cái và chữ số này được cấp theo khu vực quận – huyện. Tiếp theo là dấu gạch ngang,
- Tiếp theo là một dãy 5 số gồm 2 cụm: 3 chữ số đầu và hai chữ số sau, phân tách bởi dấu chấm.

Thông thường, người ta chỉ quan tâm đến 5 chữ số cuối. Giả sử các trường hợp sau được coi là đẹp:

- Cá 5 chữ số được sắp theo thứ tự tăng chất từ trái qua phải.
- Cá 5 chữ số đều bằng nhau
- Ba chữ số đầu bằng nhau và hai chữ số cuối bằng nhau
- Cá 5 chữ số đều là 6 và/hoặc 8 (số lộc phát).

Theo quy tắc trên, các biển số sau được coi là đẹp:

- 29T1-123.79
- 29T1-555.55
- 29T1-222.33
- 29T1-686.88

Và các biển số sau không đẹp:

- 29T1-123.33
- 29T1-555.54
- 29T1-606.88

Viết chương trình kiểm tra xem các biển số xe có đẹp hay không.

Dữ liệu vào

- Dòng đầu ghi số bộ test, không quá 50
- Mỗi bộ test là một biển số. Dữ liệu vào đảm bảo biển số được viết đúng quy định.

Kết quả

- In ra kết quả kiểm tra với từng bộ test

Ví dụ

Input	Output
7	YES
29T1-123.45	YES
29T1-555.55	YES
29T1-222.33	YES
29T1-686.88	NO
29T1-123.33	NO
29T1-555.54	NO
29T1-606.88	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

143 Số đầy đủ

Cho một số nguyên dương lớn có nhiều hơn 20 chữ số nhưng không quá 1000 chữ số. Hãy kiểm tra xem số đó có đầy đủ tất cả các chữ số từ 0 đến 9 hay không.

Dữ liệu vào

- Đòng đầu ghi số bộ test, không quá 10
- Mỗi bộ test là một dãy ký tự có độ dài không quá 1000, không có khoảng trắng

Kết quả

- Nếu dữ liệu vào không phải là một số nguyên hợp lệ (có ký tự không phải số hoặc bắt đầu bằng chữ số 0) thì in ra INVALID
- Nếu dữ liệu vào thỏa mãn đầy đủ thì in ra YES, nếu không in ra NO

Ví dụ

Input	Output
3	INVALID
01234aa32432432432534545b987978	YES
1234444444444456666666667890	NO
324562783924723543243242354354333234324	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

144 Hiệu hai số nguyên lớn

Cho hai số rất lớn X và Y được biểu diễn như hai xâu ký tự. Nhiệm vụ của bạn là tìm $|X-Y|$?

Input:

- Đòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test gồm hai dòng: dòng thứ nhất đưa xâu X; dòng tiếp theo đưa vào xâu Y.
- T, X, Y thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq \text{length}(X), \text{length}(Y) \leq 10^3$.

Output:

- Đưa ra số kết quả mỗi test theo từng dòng.

Ví dụ:

Input:	Output:
2	11999
978	0999900
12977	
100	
1000000	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

145 Tổng hai số nguyên lớn

Cho hai số rất lớn X và Y được biểu diễn như hai xâu ký tự. Nhiệm vụ của bạn là tìm $X+Y$?

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test gồm hai dòng: dòng thứ nhất đưa xâu X; dòng tiếp theo đưa vào xâu Y.
- T, X, Y thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq \text{length}(X), \text{length}(Y) \leq 10^3$.

Output:

- Đưa ra số kết quả mỗi test theo từng dòng.

Ví dụ:

Input:	Output:
1	198123
12	
198111	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

146 Phép chia dư của số nguyên lớn

Cho số nguyên dương N rất lớn được biểu diễn như một xâu và số M. Hãy tìm K = N%M. Ví dụ N=123456789873123456778976, M = 100 thì K=76.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp mỗi dòng đưa vào các test. Mỗi test là bộ đôi N, M được viết trên hai dòng khác nhau.
- T, N, M thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq \text{length}(N) \leq 1000$; $2 \leq M \leq 10^{12}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	5
12345	
10	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

147 Phép chia dư của lũy thừa số nguyên lớn

Cho số nguyên dương N rất lớn được biểu diễn như một xâu và số M. Hãy tìm K = N%M. Ví dụ N=123456789873123456778976, M = 100 thì K=76.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp mỗi dòng đưa vào các test. Mỗi test là bộ đôi N, M được viết trên hai dòng khác nhau.
- T, N, M thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq \text{length}(N) \leq 1000$; $2 \leq M \leq 10^{12}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	5
12345	
10	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

148 Tính tổng các số trong xâu

TÍNH TỔNG CÁC SỐ TRONG XÂU

Bài làm tốt nhất

Cho xâu ký tự S bao gồm các ký tự 'a'...,'z' và các chữ số. Nhiệm vụ của bạn là hãy tính tổng các số có mặt trong xâu.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S.
- T, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq \text{Length}(S) \leq 10^5$.
- Input đảm bảo đáp án không vượt quá 10^9 .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
4	24
1abc23	4
geeks4geeks	100
1abc2x30yz67	123
123abc	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

149 Tìm số lớn nhất trong xâu

TÌM SỐ LỚN NHẤT TRONG XÂU

Bài làm tốt nhất

Cho xâu ký tự S bao gồm các ký tự 'a'...,'z' và các chữ số. Nhiệm vụ của bạn là hãy tìm số lớn nhất có mặt trong xâu.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S.
- T, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq \text{Length}(S) \leq 10^5$.
- Input đảm bảo đáp số không vượt quá 10^9 .

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
3	564
100klh564abc365bg	9
abvhdf9sdnkjdfs	0
abchsd0sdhs	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

150 Đếm xâu con

ĐẾM XÂU CON

Bài làm tốt nhất

Cho xâu ký tự S và số k. Nhiệm vụ của bạn là đếm số xâu con của S có đúng k ký tự khác nhau. Các xâu con không nhất thiết phải khác nhau. Ví dụ với xâu S="abc" và k = 2, ta có kết quả là 2 bao gồm các xâu con: "ab", "bc".

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S và số k được viết trên một dòng.
- T, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq k \leq 26$; $1 \leq \text{Length}(S) \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
abc 2	3
aba 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

151 Tách chữ số

TÁCH CHỮ SỐ

Bài làm tốt nhất

Cho xâu ký tự S bao gồm các ký tự 'A'..., 'Z' và các chữ số '0'..., '9'. Nhiệm vụ của bạn in các ký tự từ 'A'..., 'Z' trong S theo thứ tự anphabet và nối với tổng các chữ số trong S ở cuối cùng. Ví dụ S ="ACCBA10D2EW30" ta nhận được kết quả: "AABCCDEW6".

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S.
- T, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{Length}(S) \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	ABCEW5
AC2BEW3	AABCCDEW6
ACCBA10D2EW30	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

152 Chuẩn hóa họ tên

Khi viết tên người Việt Nam sang dạng chuẩn Tiếng Anh không dấu, người ta có thể lựa chọn 1 trong 2 cách viết:

- Cách 1: tên có 1 từ.

Ví dụ: Nguyen Manh Son được viết thành Son Nguyen Manh

- Cách 2: họ có 1 từ.

Ví dụ: Nguyen Manh Son được viết thành Manh Son Nguyen

Hãy viết chương trình chuẩn hóa xâu ký tự họ tên theo một trong hai dạng trên.

Input

Dòng đầu ghi số bộ test.

Mỗi bộ test gồm 2 dòng. Dòng đầu ghi số 1 hoặc số 2 cho biết cách chuẩn hóa được chọn. Dòng thứ 2 ghi xâu họ tên có độ dài không quá 60 ký tự.

Output

Ghi ra danh sách các xâu họ tên đã chuẩn hóa.

Chú ý: có thể phải loại bỏ khoảng trắng, chuyển đổi chữ hoa chữ thường cho đúng với cách viết họ tên chuẩn trước khi áp dụng một trong hai cách viết nói trên.

Ví dụ

Input	Output
2	Nam Nguyen Van
1	Le Nhat Thong Chi Hoang
NguyeN vAn Nam	
2	
hoAng le Nhat THONG chi	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

153 Địa chỉ email – 2

Địa chỉ email của các cán bộ, giảng viên PTIT được tạo ra bằng cách viết đầy đủ tên và ghép với các chữ cái đầu của họ và tên đệm.

Nếu có nhiều người cùng email thì từ người thứ 2 sẽ thêm số thứ tự vào email đó.

Cho trước các xâu họ tên (có thể không chuẩn). Hãy tạo ra các địa email tương ứng.

Input:

- Dòng 1 ghi số N là xâu họ tên trong danh sách
- N dòng tiếp theo ghi lần lượt các xâu họ tên (không quá 50 ký tự)

Output: Ghi ra các email được tạo ra.

Ví dụ:

Input	Output
4	vinhnq@ptit.edu.vn
nGUYEn quaNG vInH	huongtt@ptit.edu.vn
tRan thi THU huOnG	vinhnq2@ptit.edu.vn
nGO quoC VINH	anhlt@ptit.edu.vn
LE tuAn aNH	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

154 Điện thoại cục gạch

Một thời không quá xa, điện thoại di động với chỉ các tính năng nghe, gọi, nhắn tin vẫn còn chiếm的地位 đa số thiết bị cầm tay tại Việt Nam. Khi nhắn tin, người nhắn sẽ bấm các phím số một lần bốn lần liên tiếp tương ứng với ký tự đi kèm ghi trên đó.

Cụ thể: các số và chữ cái tương ứng gồm:

2: ABC, 3: DEF, 4: GHI, 5: JKL

6: MNO, 7: PQRS, 8: TUV, 9: WXYZ

Cho trước dãy ký tự mô tả tin nhắn (không tính các ký tự khác ngoài danh sách nêu trên). Hãy kiểm tra xem dãy số được nhắn ứng với dãy ký tự đó có phải số thuận nghịch hay không (chi xét tương ứng giữa số và ký tự, không tính số đó được nhắn bao nhiêu lần, ví dụ tất cả A,B,C,a,b,c đều chỉ là một chữ số 2).

Input

Dòng đầu tiên là số bộ test, không quá 1000.

Mỗi test là dãy ký tự mô tả tin nhắn.

Output

Ghi ra kết quả kiểm tra, YES nếu dãy số là thuận nghịch, NO nếu ngược lại.

Ví dụ

Input	Output
2	YES
BOHIMA	NO
IamACoder	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

155 Xử lý văn bản

Cho dữ liệu vào là luồng văn bản bất kỳ, gồm các ký tự viết hoa, viết thường, các ký tự số và các dấu câu, không có các ký tự đặc biệt khác. Người ta muốn tách văn bản thành các câu. Mỗi câu in trên một dòng.

Một câu được định nghĩa là dãy ký tự có *ít nhất 1 ký tự chữ cái hoặc chữ số*, không chứa các dấu ngắt câu gồm: dấu chấm (.), dấu chấm hỏi (?) và dấu chấm cảm (!).

Dấu phẩy (,) và dấu hai chấm (:) không được coi là dấu ngắt câu.

Nhiệm vụ của bạn là in ra mỗi câu trên một dòng, ký tự đầu câu viết hoa, các ký tự khác viết thường, các từ cách nhau đúng một khoảng trống. Không có khoảng trống ở đầu và cuối câu, và không in ra các dấu ngắt câu.

Input

Gồm một luồng văn bản không quá 100 dòng, mỗi dòng không quá 200 ký tự.

Output

In ra các câu, mỗi câu trên một dòng theo quy tắc đã cho.

Ví dụ

Input
ky thi LAP TRINH ICPC PTIT bat dau to chuc tu nam 2010. nhu vay, nam nay la tron 10 nam! vay CO PHAI NAM NAY LA KY THI LAN THU 10? khong phai! nam nay la KY THI LAN THU 11.
Output
Ky thi lap trinh icpc ptit bat dau to chuc tu nam 2010 Nhu vay, nam nay la tron 10 nam Vay co phai nam nay la ky thi lan thu 10 Khong phai Nam nay la ky thi lan thu 11

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

156 Sắp xếp lại dãy con

Cho mảng A[] gồm n phần tử. Hãy tìm dãy con liên tục của mảng A[R], ..., A[L] sao cho khi sắp xếp lại dãy con ta nhận được một mảng được sắp xếp. Ví dụ với A[] = {10, 12, 20, 30, 25, 40, 32, 31, 35, 50, 60} ta chỉ cần sắp xếp lại dãy con từ A[4]..., A[9]: {30, 25, 40, 32, 31, 35} để có mảng được sắp.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng tiếp theo là n số A[i] của mảng A[] các số được viết cách nhau một vài khoảng trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^6$; $0 \leq A[i] \leq 10^7$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4 9
11	3 6
10 12 20 30 25 40 32 31 35 50 60	
9	
0 1 15 25 6 7 30 40 50	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

157 Hợp và giao của hai dãy số – 2

Cho mảng A[] gồm n phần tử, mảng B[] gồm m phần tử khác nhau. Các phần tử của mảng A[] và B[] chưa được sắp xếp. Hãy tìm mảng hợp và giao được sắp giữa A[] và B[]. Ví dụ với A[] = {7, 1, 5, 2, 3, 6}, B[] = {3, 8, 6, 20, 7} ta có mảng hợp Union = {1, 2, 3, 5, 6, 7, 8, 20}, mảng giao Intersection = {3, 6}.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào n, m là số phần tử của mảng A[] và B[]; dòng tiếp theo là n số A[i] của mảng A[]; dòng tiếp theo là m số B[i] của mảng B[]; các số được viết cách nhau một vài khoảng trắng.
- T, n, m, A[i], B[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n, m, A[i], B[i] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	1 2 3 5 6 7 8 20
6 5	3 6 7
7 1 5 2 3 6	
3 8 6 20 7	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

158 Sắp xếp theo khoảng cách

Cho mảng A[] gồm n phần tử và số X. Hãy đưa sắp xếp các phần tử của mảng theo trị tuyệt đối của $|X - A[i]|$. Ví dụ với A[] = {10, 5, 3, 9, 2} và X = 7 ta đưa ra mảng được sắp xếp theo nguyên tắc kê trên: A[] = {5, 9, 10, 3, 2} vì $|7-10|=3$, $|7-5|=2$, $|7-3|=4$, $|7-9|=2$, $|7-2|=5$. Trong trường hợp có nhiều phần tử có giá trị tuyệt đối như nhau, ưu tiên theo thứ tự số xuất hiện trước trong mảng ban đầu.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n và X; dòng tiếp theo là n số A[i] của mảng A[]; các số được viết cách nhau một vài khoảng trắng.
- T, n, X thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n, X, A[i] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	5 9 10 3 2
5 7	5 4 3 2 1
10 5 3 9 2	
5 6	
1 2 3 4 5	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

159 Số nhỏ hơn K (SD cửa sổ trượt)

Cho mảng $A[]$ gồm n số nguyên dương và số k . Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng sao cho các số nhỏ hơn hoặc bằng k đứng cạnh nhau. Ví dụ với mảng $A[] = \{2, 1, 5, 6, 3\}$, $k = 3$ ta chỉ cần thực hiện 1 phép đổi chỗ để có mảng $A[] = \{2, 1, 3, 6, 5\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng $A[]$ và số k ; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trắng.
- $T, n, k, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $1 \leq A[i], k \leq 10^7$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
5 3	2
2 1 5 6 3	
7 5	
2 7 9 5 8 7 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

160 Sắp xếp chẵn lẻ

Cho mảng $A[]$ gồm n số nguyên dương. Nhiệm vụ của bạn là hãy sắp xếp lại các phần tử của mảng sao cho $A[i] \geq A[i-1]$ nếu i chẵn, $A[i] \leq A[i-1]$ nếu i lẻ. Ví dụ với mảng $A[] = \{1, 2, 2, 1\}$ ta được mảng được sắp $A[] = \{1, 2, 1, 2\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng $A[]$; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trắng.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A[i] \leq 10^3$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2 1 2
4	1 3 2
1 2 2 1	
3	
1 3 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

161 Sắp xếp xen kẽ – 2

Cho mảng $A[]$ gồm n số nguyên dương. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng sao theo nguyên tắc số lớn nhất ở đầu, số nhỏ nhất thứ 2, số lớn thứ nhì ở vị trí tiếp theo, số nhỏ thứ nhì ở vị trí kế tiếp Ví dụ với mảng $A[] = \{1, 7, 3, 5, 9\}$ ta được mảng được sắp $A[] = \{9, 1, 7, 3, 5\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng $A[]$; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trắng.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq n \leq 10^3; 1 \leq A[i] \leq 10^3$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7 1 5 2
4	9 1 8 2 7 3
1 5 2 7	
6	
1 3 2 7 9 8	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

162 Nhân đôi cặp số bằng nhau

Cho mảng $A[]$ gồm n số nguyên không âm. Ta gọi phần tử $A[i]$ là hợp lệ nếu $A[i] \neq 0$ ngược lại là không hợp lệ. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng theo nguyên tắc nếu số $A[i+1]$ ($i=0, \dots, n-2$) là số hợp lệ và $A[i] = A[i+1]$ thì nhân đôi $A[i]$ và thiết lập $A[i+1] = 0$. Sau khi thay đổi, hãy di chuyển các số không hợp lệ vào cuối mảng. Các số hợp lệ phải bảo toàn quan hệ trước sau. Ví dụ với mảng $A[] = \{2, 2, 0, 4, 0, 8\}$ thực hiện theo nguyên tắc trên ta được kết quả $A[] = \{4, 4, 8, 0, 0, 0\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng $A[]$; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trắng.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq n \leq 10^5; 1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4 4 8 0 0 0
6	4 2 12 8 0 0 0 0 0 0
2 2 0 4 0 8	
10	
0 2 2 2 0 6 6 0 0 8	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

163 Sửa đèn

Tuyến đường ven biển của thành phố Highland có N chiếc đèn. Không may cơn bão vừa rồi đã làm hỏng B chiếc đèn.

Để khắc phục sự cố và nhanh chóng khôi phục lại hoạt động du lịch, chính quyền thành phố đã quyết định sửa tạm thời một số đèn đường bị hỏng sao cho có ít nhất một khu vực có K chiếc đèn liên tiếp hoạt động.

Các bạn hãy xác định xem số đèn đường cần phải sửa chữa ít nhất là bao nhiêu?

Input

Dòng đầu tiên gồm 3 số nguyên dương N, K và B ($1 \leq B, K \leq N \leq 100\,000$).

B dòng tiếp theo, mỗi dòng chứa vị trí của một chiếc đèn bị hỏng.

Output

In ra số đèn đường cần sửa ít nhất sao cho có một khu vực có nhiều hơn hoặc bằng K chiếc đèn hoạt động.

Ví dụ:

Input	Output
10 6 5	1
2	
10	
1	
5	
9	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

164 Đếm cặp phần tử có hiệu nhỏ hơn K (2 con trỏ)

ĐẾM CẶP PHẦN TỬ CÓ HIỆU NHỎ HƠN K

Bài làm tốt nhất

Cho mảng A[] gồm n số nguyên dương và số k. Nhiệm vụ của bạn là đếm số các cặp phần tử có hiệu nhỏ hơn k. Ví dụ A[] = {1, 10, 4, 2}, k=3 ta nhận được kết quả là 2 tương ứng với hiệu các cặp (1, 2), (4, 2).

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n, tương ứng với số phần tử của mảng A[] và số k; dòng tiếp theo là n số A[i]; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^5$; $1 \leq k \leq 10^3$; $1 \leq A[i] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
4 3	3
1 10 4 2	
3 5	
2 3 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

165 Sắp xếp theo chữ số

SẮP XẾP THEO CHỮ SỐ

Bài làm tốt nhất

Cho mảng A[] gồm n số nguyên không âm. Hãy sắp đặt lại các phần tử trong mảng sao cho khi nối các số lại với nhau ta được một số lớn nhất. Ví dụ với mảng A[] = {54, 546, 548, 60} thực hiện theo nguyên tắc trên ta được số lớn nhất là 6054854654.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^5$; $1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	6054854654
4	998764543431
54 546 548 60	
8	
1 34 3 98 9 76 45 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

166 Biến đổi dãy số – 2

BIẾN ĐỔI DÃY SỐ - 2

Bài làm tốt nhất

Cho mảng A[] gồm n số nguyên không âm. Hãy sắp đặt lại các phần tử trong mảng theo nguyên tắc:

- Phần tử đầu tiên được thay bằng tích của nó và phần tử kế tiếp.
- Phần tử cuối cùng được thay bằng tích của nó và sau nó.
- Các phần tử còn lại được thay bằng tích của phần tử sau nó và phần tử trước nó.

Ví dụ với mảng A[] = {2, 3, 4, 5, 6} thực hiện theo nguyên tắc trên ta được mảng A[] = {6, 8, 15, 24, 30}.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^5$; $1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	6 8 15 24 30
5	63 108 56 72 40 30
2 3 4 5 6	
6	
9 7 12 8 6 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

167 Khoảng cách bằng X

Cho mảng A[] gồm N phần tử và số X. Nhiệm vụ của bạn là tìm cặp phần tử $A[i] - A[j] = X$. Nếu tồn tại $A[i] - A[j] = X$ đưa ra 1, ngược lại đưa ra -1.

Input:

- Đòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất là cặp số N, X; dòng tiếp theo là N số A[i] là các phần tử của mảng A[].
- T, N, X, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 10^5$, $1 \leq X, A[i] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
6 78	-1
5 20 3 2 5 80	
5 45	
90 70 20 80 50	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

168 Phần tử gần nhất

Cho mảng A[] gồm N phần tử đã được sắp xếp. Hãy tìm K phần tử gần nhất của X khác X. Ví dụ với mảng A[] = {1, 3, 5, 7, 9, 11}, X = 7, K=2 thì ta có 2 phần tử gần nhất của 7 là 5 và 9. K là số chẵn và cần lấy một nửa lớn, một nửa nhỏ hơn X.

Input:

- Đòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm ba dòng: dòng thứ nhất đưa vào số phần tử của mảng N; dòng tiếp theo là N số A[i] là các phần tử của mảng A[]; dòng tiếp theo đưa vào hai số K và X.
- T, N, K, X, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N, K, X \leq 10^6$, $1 \leq A[i] \leq 10^6$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

169 Nhỏ nhất thứ K

Cho mảng A[] gồm N phần tử. Hãy đưa ra các phần tử nhỏ nhất thứ k của mảng.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào N,k là số phần tử của mảng A[] và số k; các dòng tiếp theo đưa vào N số A[i].
- T, N, A[i], k thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq k < N \leq 10^6$, $0 \leq A[i] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 6 3 10 5 15 50 40 80	15

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

170 Tam giác vuông

TAM GIÁC VUÔNG

Bài làm tốt nhất

Theo định lý Pytago, ta đã biết một bộ 3 số (a, b, c) thỏa mãn $a^2 + b^2 = c^2$ thì đó là ba cạnh của một tam giác vuông.

Cho dãy số A[] gồm có N phần tử. Nhiệm vụ của bạn là kiểm tra xem **có tồn tại bộ ba số thỏa mãn là ba cạnh của tam giác vuông hay không.**

Đữ liệu vào:

- Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).
- Mỗi test gồm số nguyên N ($1 \leq N \leq 5000$).
- Dòng tiếp theo gồm N số nguyên A[i] ($1 \leq A[i] \leq 10^9$).

Kết quả:

Với mỗi test, in ra trên một dòng "YES" nếu tìm được, và "NO" trong trường hợp ngược lại.

Ví dụ:

Input	Output
2	YES
5	NO
3 1 4 6 5	
3	
1 1 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

171 Phép toán với phân số

Phân số là sự biểu diễn số hữu tỷ dưới dạng tóm tắt của hai số nguyên, trong đó số ở trên được gọi là tử số, còn số ở dưới được gọi là mẫu số. Cho hai phân số A và B có tử số và mẫu số được nhập từ bàn phím.

Viết chương trình thực hiện hai nhiệm vụ sau:

- Tính $C = (A + B)^2$ và rút gọn kết quả.
- Tính $D = A \times B \times C$ và rút gọn kết quả.

Input:

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

T dòng tiếp theo, mỗi dòng gồm 4 số lần lượt là tử và mẫu số của phân số A và phân số B với $-10^2 \leq$ tử số, mẫu số $\leq 10^2$. Mẫu số là số khác 0.

Output:

Kết quả của hai phép tính theo định dạng phân số.

Ví dụ

Input:	Output:
2 1 2 3 4 2 3 4 5	25/16 75/128 484/225 3872/3375

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main() {
    int t;
    cin >> t;
    while (t--) {
        PhanSo A;
        PhanSo B;
        cin >> A.tu >> A.mau >> B.tu >> B.mau;
        process(A, B);
    }
}
```

172 Danh sách sinh viên – 1

Viết chương trình khai báo cấu trúc Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (đang số thực float).

Đọc thông tin N sinh viên từ bàn phím (không có mã sinh viên) và in ra lần lượt màn hình mỗi dòng 1 sinh viên theo đúng thứ tự ban đầu. Trong đó Mã SV được tự tạo ra theo quy tắc thêm mã **B20DCCN** sau đó là giá trị nguyên tự động tăng tính từ 001 (tối đa là 099). Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy

Input

Dòng đầu tiên ghi số sinh viên N ($0 < N < 50$).

Mỗi sinh viên ghi trên 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

Output

Ghi ra danh sách lần lượt các sinh viên có đầy đủ Mã sinh viên, Họ tên, Lớp, Ngày sinh (đã chuẩn hóa về dạng dd/mm/yyyy), Điểm GPA (với đúng 2 số sau dấu phẩy).

Mỗi sinh viên ghi trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trắng.

Ví dụ

Input	Output
1 Nguyen Van An D20CQCN01-B 2/12/2002 3.19	B20DCCN001 Nguyen Van An D20CQCN01-B 02/12/2002 3.19

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main(){
    struct SinhVien ds[50];
    int N;
    cin >> N;
    nhap(ds, N);
    in(ds, N);
    return 0;
}
```

173 Danh sách sinh viên – 2

Viết chương trình khai báo cấu trúc Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp, Ngày sinh và Điểm GPA (dạng số thực float).

Đọc thông tin N sinh viên từ bàn phím (không có mã sinh viên) và in ra lần lượt màn hình mỗi dòng 1 sinh viên theo đúng thứ tự ban đầu. Trong đó Mã SV được tự tạo ra theo quy tắc thêm mã **B20DCCN** sau đó là giá trị nguyên tự động tăng tính từ 001 (tối đa là 099). Họ tên được xử lý đưa về dạng chuẩn. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy

Input

Dòng đầu tiên ghi số sinh viên N ($0 < N < 50$).

Mỗi sinh viên ghi trên 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

Output

Ghi ra danh sách lần lượt các sinh viên có đầy đủ Mã sinh viên, Họ tên, Lớp, Ngày sinh (đã chuẩn hóa), điểm GPA (với đúng 2 số sau dấu phẩy).

Mỗi sinh viên ghi trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trắng.

Ví dụ

Input	Output
1 nGuyEn vaN biNH D20CQCN01-B 2/12/2002 3.1	B20DCCN001 Nguyen Van Binh D20CQCN01-B 02/12/2002 3.10

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    struct SinhVien ds[50];
    int N;
    cin >> N;
    nhap(ds, N);
    in(ds, N);
    return 0;
}
```

174 Danh sách sinh viên – 3

Viết chương trình khai báo cấu trúc Sinh Viên gồm các thông tin: Mã SV, Họ tên, Ngày sinh, Lớp và Điểm GPA (dạng số thực float).

Đọc thông tin N sinh viên từ bàn phím (không có mã sinh viên) sau đó sắp xếp theo điểm GPA giảm dần và in ra lần lượt màn hình mỗi dòng 1 sinh viên.

Trong đó Mã SV được tự tạo ra theo quy tắc thêm mã **B20DCCN** sau đó là giá trị nguyên tử động tăng tính từ 001 (tối đa là 099). Họ tên được xử lý đưa về dạng chuẩn. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy

Input

Dòng đầu tiên ghi số sinh viên N ($0 < N < 50$).

Mỗi sinh viên ghi trên 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiêu nhất 2 số sau dấu phẩy.

Dữ liệu đảm bảo không có hai sinh viên nào có điểm GPA bằng nhau.

Output

Ghi ra danh sách lần lượt các sinh viên có đầy đủ Mã sinh viên, Họ tên, Lớp, Ngày sinh (đã chuẩn hóa), điểm GPA (với đúng 2 số sau dấu phẩy) đã được sắp xếp theo điểm GPA giảm dần.

Mỗi sinh viên ghi trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trắng.

Ví dụ

Input	Output
2	B20DCCN002 Nguyen Quang Hai D20DCCN02-B 01/09/1994 3.00
nguyen Van Nam	B20DCCN001 Nguyen Van Nam D20DCCN01-B 02/12/1994 2.17
D20DCCN01-B	
2/12/1994	
2.17	
Nguyen Quang hAi	
D20DCCN02-B	
1/9/1994	
3.0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    struct SinhVien ds[50];
    int N;
    cin >> N;
    nhap(ds, N);
    sapxep(ds,N);
    in(ds, N);
    return 0;
}
```

175 Sắp xếp danh sách mặt hàng

Xây dựng cấp trúc mặt hàng gồm các thông tin: Mã mặt hàng (là một số nguyên, tự động tăng, tính từ 1); Tên mặt hàng, nhóm hàng; là các xâu ký tự; Giá mua, giá bán: là các số thực (không quá 9 chữ số)

Hãy sắp xếp danh sách các mặt hàng theo lợi nhuận giảm dần.

Input:

Dòng đầu chứa số mặt hàng. Mỗi mặt hàng viết trên 4 dòng: Dòng 1: Tên mặt hàng. Dòng 2: Nhóm hàng. Dòng 3: Giá mua.

Dòng 4: Giá bán

Output: Ghi ra danh sách mặt hàng đã sắp xếp theo lợi nhuận giảm dần (lợi nhuận tính bằng giá bán trừ đi giá mua). Mỗi mặt hàng viết trên một dòng gồm: mã, tên, nhóm hàng và lợi nhuận. Các thông tin cách nhau đúng 1 khoảng trắng. Lợi nhuận viết với 2 chữ số sau dấu phẩy.

Ví dụ:

Input	Output
3	2 Tu lanh Side by Side Dien lanh 7699.00
May tinh SONY VAIO	1 May tinh SONY VAIO Dien tu 1299.00
Dien tu	3 Banh Chocopie Tieu dung 9.50
16400	
17699	
Tu lanh Side by Side	
Dien lanh	
18300	
25999	
Banh Chocopie	
Tieu dung	
27.5	
37	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

176 Sắp xếp danh sách nhân viên

Một nhân viên làm việc trong công ty được lưu lại các thông tin sau:

- Mã nhân viên: được gán tự động tăng, bắt đầu từ 00001
- Họ tên: Xâu ký tự không quá 40 chữ cái
- Giới tính: Nam hoặc Nữ
- Ngày sinh: đúng theo chuẩn dd/mm/yyyy
- Địa chỉ: Xâu ký tự không quá 100 chữ cái
- Mã số thuế: Dãy số có đúng 10 chữ số
- Ngày ký hợp đồng: đúng theo chuẩn dd/mm/yyyy

Viết chương trình nhập danh sách nhân viên (không nhập mã), sau đó sắp xếp theo thứ tự ngày sinh từ già nhất đến trẻ nhất và in ra màn hình danh sách nhân viên đã sắp xếp.

Input

Dòng đầu ghi số N là số nhân viên (không quá 40). Mỗi nhân viên ghi trên 6 dòng lần lượt ghi các thông tin theo thứ tự đã ghi trong đề bài. Không có mã nhân viên.

Output

Ghi ra danh sách đầy đủ nhân viên đã sắp xếp, mỗi nhân viên trên một dòng, các thông tin cách nhau đúng một khoảng trắng.

Ví dụ

Input
<pre>3 Nguyen Van A Nam 10/22/1982 Mo Lao-Ha Dong-Ha Noi 8333012345 31/12/2013 Ly Thi B Nu 10/15/1988 Mo Lao-Ha Dong-Ha Noi 8333012346 22/08/2011 Hoang Thi C Nu 04/02/1981 Mo Lao-Ha Dong-Ha Noi 8333012347 22/08/2011</pre>
Output
<pre>00003 Hoang Thi C Nu 04/02/1981 Mo Lao-Ha Dong-Ha Noi 8333012347 22/08/2011 00001 Nguyen Van A Nam 10/22/1982 Mo Lao-Ha Dong-Ha Noi 8333012345 31/12/2013 00002 Ly Thi B Nu 10/15/1988 Mo Lao-Ha Dong-Ha Noi 8333012346 22/08/2011</pre>

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    struct NhanVien ds[50];
    int N,i;
    cin >> N;
    for(i = 0; i < N; i++) nhap(ds[i]);
    sapxep(ds, N);
    inds(ds, N);
    return 0;
}
```

177 Bảng điểm thành phần – 1

Cho dữ liệu bảng điểm thành phần trong đó thông tin của mỗi sinh viên gồm:

- Mã sinh viên (xâu ký tự độ dài không quá 15, không có khoảng trắng)
- Tên sinh viên (xâu ký tự, độ dài không quá 50)
- Lớp (xâu ký tự độ dài không quá 15, không có khoảng trắng)
- Điểm 1, Điểm 2, Điểm 3: mỗi điểm là một số thực (hệ 10)

Hãy sắp xếp lại bảng điểm thành phần theo mã sinh viên (thứ tự từ điển tăng dần).

Input

Dòng đầu ghi số sinh viên (không quá 100).

Mỗi sinh viên ghi trên 6 dòng lần lượt là: Mã SV, Họ tên, Lớp, Điểm 1, Điểm 2, Điểm 3.

Output

Ghi ra danh sách sinh viên đã sắp xếp theo mã sinh viên.

Mỗi sinh viên ghi trên 1 dòng gồm các thông tin: thứ tự, mã sv, họ tên, lớp, điểm 1, điểm 2, điểm 3. Các thông tin cách nhau đúng một khoảng trắng.

Các giá trị điểm ghi ra với đúng 1 chữ số phần thập phân.

Ví dụ

Input	Output
3	
B20DCCN999	1 B20DCAT001 Le Van Nam D20CQAT02-B 6.0 6.0 4.0
Nguyen Van An	2 B20DCCN111 Tran Hoa Binh D20CQCN04-B 9.0 8.0 6.0
D20CQCN04-B	3 B20DCCN999 Nguyen Van An D20CQCN04-B 10.0 9.0 8.0
10.0	
9.0	
8.0	
B20DCAT001	
Le Van Nam	
D20CQAT02-B	
6.0	
6.0	
4.0	
B20DCCN111	
Tran Hoa Binh	
D20CQCN04-B	
9.0	
5.0	
6.0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main(){
    int n;
    cin >> n;
    struct SinhVien *ds = new SinhVien[n];
    for(int i = 0; i < n; i++) {
        nhap(ds[i]);
    }
    sap_xep(ds, n);
    in_ds(ds,n);
    return 0;
}
```

178 Bảng điểm thành phần – 2

BÀNG ĐIỂM THÀNH PHẦN - 2

Bài làm tốt nhất

Cho dữ liệu bảng điểm thành phần trong đó thông tin của mỗi sinh viên gồm:

- Mã sinh viên (xâu ký tự độ dài không quá 15, không có khoảng trắng)
- Họ tên sinh viên (xâu ký tự, độ dài không quá 50)
- Lớp (xâu ký tự độ dài không quá 15, không có khoảng trắng)
- Điểm 1, Điểm 2, Điểm 3: mỗi điểm là một số thực (hệ 10)

Hãy sắp xếp lại bảng điểm thành phần theo họ tên (thứ tự từ điển tăng dần - so sánh cả xâu ký tự họ tên để sắp xếp, không cần tách tên).

Input

Dòng đầu ghi số sinh viên (không quá 100).

Mỗi sinh viên ghi trên 6 dòng lần lượt là: Mã SV, Họ tên, Lớp, Điểm 1, Điểm 2, Điểm 3.

Output

Ghi ra danh sách sinh viên đã sắp xếp theo họ tên.

Mỗi sinh viên ghi trên 1 dòng gồm các thông tin: thứ tự, mã sv, họ tên, lớp, điểm 1, điểm 2, điểm 3. Các thông tin cách nhau đúng một khoảng trắng. Các giá trị điểm ghi ra với đúng 1 chữ số phần thập phân.

Ví dụ

Input	Output
3	1 B20DCAT001 Le Van An D20CQAT02-B 6.0 6.0 4.0
B20DCCN999	2 B20DCCN111 Nguyen Van Binh D20CQCN01-B 9.0 5.0 6.0
Nguyen Van Nam	3 B20DCCN999 Nguyen Van Nam D20CQCN04-B 10.0 9.0 8.0
D20CQCN04-B	
10.0	
9.0	
8.0	
B20DCAT001	
Le Van An	
D20CQAT02-B	
6.0	
6.0	
4.0	
B20DCCN111	
Nguyen Van Binh	
D20CQCN01-B	
9.0	
5.0	
6.0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

179 Trẻ nhất – già nhất

Cho một danh sách tên người và ngày tháng năm sinh.

Hãy tìm ra người trẻ nhất và người già nhất.

Input

Dòng 1 ghi số N là số người (không quá 100).

N dòng tiếp theo, mỗi dòng ghi tên (xâu ký tự không có khoảng trắng và không quá 15 ký tự, sau đó là dãy ký tự mô tả ngày tháng năm sinh theo chuẩn dd/mm/yyyy).

Output

Dòng đầu ghi ra tên người trẻ nhất.

Dòng thứ 2 ghi ra tên người già nhất.

Đữ liệu đảm bảo không có 2 người nào trùng ngày sinh.

Ví dụ

Input	Output
5	Binh
Nam 01/10/1991	Tam
An 30/12/1990	
Binh 15/08/1993	
Tam 18/09/1990	
Truong 20/09/1990	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

180 Sắp xếp thời gian

Giá trị độ đo thời gian được biểu diễn bởi ba thành phần: giờ, phút, giây.

Cho N giá trị thời gian, hãy sắp xếp danh sách theo thứ tự tăng dần.

Input

Dòng đầu ghi số nguyên dương N (không quá 5000) là số lượng giá trị thời gian cần sắp xếp.

Mỗi giá trị thời gian biểu diễn trên một dòng bằng ba số nguyên dương, lần lượt là số giờ, số phút, số giây. Trong đó số giờ đảm bảo nhỏ hơn 100, số phút và số giây đảm bảo đúng quy tắc (tức là không quá 59).

Output

In ra danh sách đã sắp xếp theo thứ tự tăng dần.

Ví dụ

Input	Output
3	11 15 12
11 20 20	11 20 20
14 20 14	14 20 14
11 15 12	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

181 Danh sách thực tập – 1

Sinh viên CNTT PTIT đến năm cuối được cử đi thực tập tại các doanh nghiệp.

Thông tin của mỗi sinh viên trong danh sách thực tập bao gồm:

- Số thứ tự: là 1 số nguyên tự động tăng
- Mã sinh viên: là một xâu ký tự không có khoảng trắng, không quá 12 ký tự
- Họ tên: là một xâu ký tự họ tên đã chuẩn hóa, không quá 50 ký tự
- Lớp: là một xâu ký tự không có khoảng trắng, không quá 10 ký tự
- Email: là một địa chỉ email, không có khoảng trắng, không quá 100 ký tự
- Doanh nghiệp: tên viết tắt của doanh nghiệp, không có khoảng trắng, không quá 15 ký tự.

Hãy viết chương trình đọc vào danh sách thực tập sau đó in danh sách cho từng doanh nghiệp theo yêu cầu.

Input

Dòng đầu ghi số N là sinh viên

Mỗi sinh viên ghi trên 5 dòng gồm mã, họ tên, lớp, email và doanh nghiệp.

Không có số thứ tự, cần tự gắn theo thứ tự tăng dần từ 1.

Sau khi hết danh sách sinh viên sẽ có một số nguyên Q (không quá 5) cho biết danh sách truy vấn.

Tiếp theo là Q dòng, mỗi dòng ghi tên một doanh nghiệp (đúng như trong danh sách, không có trường hợp nào không tồn tại trong danh sách)

Output

Với mỗi doanh nghiệp, liệt kê danh sách sinh viên thực tập ở doanh nghiệp đó theo thứ tự sắp xếp họ tên (so sánh cả xâu họ tên theo thứ tự từ điển, không cần tách riêng phần tên).

Mỗi sinh viên trên một dòng. Mỗi thông tin trong danh sách cách nhau đúng một khoảng trắng.

Ví dụ

Input	Output
6	
B17DCCN016	3 B17DCAT092 Cao Danh Huy D17CQAT04-B test3@stu.ptit.edu.vn FPT
Le Khac Tuan Anh	2 B17DCCN107 Dao Thanh Dat D17CNPM5 test2@stu.ptit.edu.vn FPT
D17HTTT2	5 B17DCCN461 Dinh Quang Nghia D17CNPM2 test5@stu.ptit.edu.vn FPT
test1@stu.ptit.edu.vn	
VIETTEL	
B17DCCN107	
Dao Thanh Dat	
D17CNPM5	
test2@stu.ptit.edu.vn	
FPT	
B17DCAT092	
Cao Danh Huy	
D17CQAT04-B	
test3@stu.ptit.edu.vn	
FPT	
B17DCCN088	
Cao Sy Hai Long	
D17CNPM2	
test4@stu.ptit.edu.vn	
VNPT	
B17DCCN461	
Dinh Quang Nghia	
D17CNPM2	
test5@stu.ptit.edu.vn	
FPT	
B17DCCN554	
Bui Xuan Thai	
D17CNPM1	
test6@stu.ptit.edu.vn	
GAMELOFT	
1	
FPT	

182 Danh sách thực tập – 2

DANH SÁCH THỰC TẬP - 2

Bài làm tốt nhất

Sinh viên CNTT PTIT đến năm cuối được cử đi thực tập tại các doanh nghiệp.

Thông tin của mỗi sinh viên trong danh sách thực tập bao gồm:

- Số thứ tự: là 1 số nguyên tự động tăng
- Mã sinh viên: là một xâu ký tự không có khoảng trắng, không quá 12 ký tự
- Họ tên: là một xâu ký tự họ tên đã chuẩn hóa, không quá 50 ký tự
- Lớp: là một xâu ký tự không có khoảng trắng, không quá 10 ký tự
- Email: là một địa chỉ email, không có khoảng trắng, không quá 100 ký tự
- Doanh nghiệp: tên viết tắt của doanh nghiệp, không có khoảng trắng, không quá 15 ký tự.

Hãy viết chương trình đọc vào danh sách thực tập sau đó in danh sách cho từng doanh nghiệp theo yêu cầu.

Input

Dòng đầu ghi số N là sinh viên

Mỗi sinh viên ghi trên 5 dòng gồm mã, họ tên, lớp, email và doanh nghiệp.

Không có số thứ tự, cần tự gắn theo thứ tự tăng dần từ 1.

Sau khi hết danh sách sinh viên sẽ có một số nguyên Q (không quá 5) cho biết danh sách truy vấn.

Tiếp theo là Q dòng, mỗi dòng ghi tên một doanh nghiệp (đúng như trong danh sách, không có trường hợp nào không tồn tại trong danh sách)

Output

Với mỗi doanh nghiệp, liệt kê danh sách sinh viên thực tập ở doanh nghiệp đó theo thứ tự sắp xếp mã sinh viên (so sánh theo thứ tự từ điển).

Mỗi sinh viên trên một dòng. Mỗi thông tin trong danh sách cách nhau đúng một khoảng trắng.

Ví dụ

Input	Output
6	3 B17DCAT092 Cao Danh Huy D17CQAT04-B test3@stu.ptit.edu.vn FPT
B17DCCN016	2 B17DCCN107 Dao Thanh Dat D17CNPM5 test2@stu.ptit.edu.vn FPT
Le Khac Tuan Anh	5 B17DCCN461 Dinh Quang Nghia D17CNPM2 test5@stu.ptit.edu.vn FPT
D17HTT2	
test1@stu.ptit.edu.vn	
VIETTEL	
B17DCCN107	
Dao Thanh Dat	
D17CNPM5	
test2@stu.ptit.edu.vn	
FPT	
B17DCAT092	
Cao Danh Huy	
D17CQAT04-B	
test3@stu.ptit.edu.vn	
FPT	
B17DCCN388	
Cao Sy Hai Long	
D17CNPM2	
test4@stu.ptit.edu.vn	
VNPT	
B17DCCN461	
Dinh Quang Nghia	
D17CNPM2	
test5@stu.ptit.edu.vn	
FPT	
B17DCCN554	
Bui Xuan Thai	
D17CNPM1	
test6@stu.ptit.edu.vn	
GAMELOFT	
1	
FPT	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

183 Số thuận nghịch giảm dần

Cho dữ liệu vào dạng văn bản, với không quá 1000 số nguyên dương, các số không quá 50 chữ số. Hãy liệt kê các số thuận nghịch khác nhau theo thứ tự giảm dần và số lần xuất hiện của nó.

Input

Luồng vào dạng văn bản với không quá 1000 số nguyên dương. Không biết trước số dòng hoặc số lượng chính xác bao nhiêu số nguyên.

Output

Ghi ra các số thuận nghịch theo thứ tự giá trị giảm dần và số lần xuất hiện của nó, mỗi số trên một dòng. Không tính các số có 1 chữ số.

Ví dụ

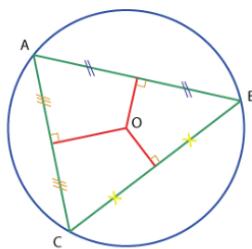
Input	Output
12321 456 12321 34 56	99999999999999999999 1
99999999999999999999	12321 4
12 3 43 34 54 34 54 34 54 657	565 3
5 45 554 12321 12321	
65 76 45 45 34 53	
546 565 65645 6 65 65 35 65	
565 565	

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

184 Diện tích hình tròn ngoại tiếp tam giác

Cho ba điểm A, B, C trong không gian hai chiều Oxy.

Hãy tính diện tích hình tròn ngoại tiếp tam giác tạo bởi 3 điểm trên.



Công thức Heron tính diện tích tam giác với 3 cạnh là a, b, c:

$$S = \frac{1}{4} \sqrt{(a+b+c)(a+b-c)(b+c-a)(c+a-b)}$$

Công thức tính bán kính hình tròn ngoại tiếp:

$$R = \frac{a.b.c}{4S}$$

Khi tính diện tích nên dùng hằng số PI

```
#define PI 3.141592653589793238
```

Input

Dòng đầu ghi số bộ test (không quá 20).

Mỗi bộ test ghi trên 1 dòng 6 số thực lần lượt là tọa độ của 3 điểm A, B, C. Giá trị tọa độ không quá 1000.

Output

Nếu 3 điểm không thể tạo thành tam giác, in ra INVALID

Nếu 3 điểm tạo thành tam giác, in ra diện tích hình tròn ngoại tiếp với độ chính xác 3 số phần thập phân.

Ví dụ

Input	Output
3	INVALID
0 0 0 5 0 199	INVALID
1 1 1 1 1 1	39.270
0 0 0 5 5 0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

185 Danh sách đối tượng sinh viên – 1

Viết chương trình khai báo lớp Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp và Điểm GPA (dạng số thực float).

Hàm khởi tạo không có tham số, gán các giá trị thuộc tính ở trạng thái mặc định (xâu ký tự rỗng, giá trị số bằng 0).

Yêu cầu sử dụng chèn toán tử nhập và xuất để nhập đối tượng sinh viên với cin và in ra đối tượng sinh viên với cout.

Đọc thông tin N thí sinh từ bàn phím (không có mã sinh viên) và in ra lần lượt màn hình mỗi dòng 1 sinh viên theo đúng thứ tự ban đầu. Trong đó Mã SV được tự tạo ra theo quy tắc thêm mã **B20DCCN** sau đó là giá trị nguyên tự động tăng tính từ 001 (tối đa là 099). Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy

Input

Dòng đầu tiên ghi số sinh viên N ($0 < N < 50$).

Mỗi sinh viên ghi trên 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đú 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

Output

Ghi ra danh sách lần lượt các sinh viên có đầy đủ Mã sinh viên, Họ tên, Lớp, Ngày sinh (đã chuẩn hóa về dạng dd/mm/yyyy), Điểm GPA (với đúng 2 số sau dấu phẩy).

Mỗi sinh viên ghi trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trống.

Ví dụ

Input	Output
1 Nguyen Van An D20CQCN01-B 2/12/2002 3.19	B20DCCN001 Nguyen Van An D20CQCN01-B 02/12/2002 3.19

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

C/C++

```
int main(){
    SinhVien ds[50];
    int N, i;
    cin >> N;
    for(i=0;i<N;i++){
        cin >> ds[i];
    }
    for(i=0;i<N;i++){
        cout << ds[i];
    }
    return 0;
}
```

186 Danh sách đối tượng sinh viên – 2

Viết chương trình khai báo lớp Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp và Điểm GPA (dạng số thực float).

Hàm khởi tạo không có tham số, gán các giá trị thuộc tính ở trạng thái mặc định (xâu ký tự rỗng, giá trị số bằng 0).

Yêu cầu sử dụng chòng toán tử nhập và xuất để nhập đối tượng sinh viên với cin và in ra đối tượng sinh viên với cout.

Đọc thông tin N thí sinh từ bàn phím (không có mã sinh viên) và in ra lần lượt màn hình mỗi dòng 1 sinh viên theo đúng thứ tự ban đầu. Trong đó Mã SV được tự tạo ra theo quy tắc thêm mã **B20DCCN** sau đó là giá trị nguyên tử động tăng tính từ 001 (tối đa là 099). Họ tên được xử lý đưa về dạng chuẩn. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy

Input

Dòng đầu tiên ghi số sinh viên N ($0 < N < 50$).

Mỗi sinh viên ghi trên 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

Output

Ghi ra danh sách lần lượt các sinh viên có đầy đủ Mã sinh viên, Họ tên, Lớp, Ngày sinh (đã chuẩn hóa), điểm GPA (với đúng 2 số sau dấu phẩy).

Mỗi sinh viên ghi trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trắng.

Ví dụ

Input	Output
1 nGuyEn vaN binH D20CQCN01-B 2/12/2002 3.1	B20DCCN001 Nguyen Van Binh D20CQCN01-B 02/12/2002 3.10

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    SinhVien ds[50];
    int N, i;
    cin >> N;
    for(i=0;i<N;i++){
        cin >> ds[i];
    }
    for(i=0;i<N;i++){
        cout << ds[i];
    }
    return 0;
}
```

187 Danh sách đối tượng sinh viên – 3

Viết chương trình khai báo lớp Sinh Viên gồm các thông tin: Mã SV, Họ tên, Lớp và Điểm GPA (dạng số thực float). Hàm khởi tạo không có tham số, gán các giá trị thuộc tính ở trạng thái mặc định (xâu ký tự rỗng, giá trị số bằng 0).

Yêu cầu sử dụng chèn toán tử nhập và xuất để nhập đối tượng sinh viên với cin và in ra đối tượng sinh viên với cout.

Đọc thông tin N thí sinh từ bàn phím (không có mã sinh viên) sau đó sắp xếp theo điểm GPA giảm dần và in ra lần lượt màn hình mỗi dòng 1 sinh viên.

Trong đó Mã SV được tự tạo ra theo quy tắc thêm mã **B20DCCN** sau đó là giá trị nguyên tự động tăng tính từ 001 (tối đa là 099). Họ tên được xử lý đưa về dạng chuẩn. Ngày sinh được chuẩn hóa về dạng dd/mm/yyyy

Input

Dòng đầu tiên ghi số sinh viên N ($0 < N < 50$).

Mỗi sinh viên ghi trên 4 dòng lần lượt là Họ tên, Lớp, Ngày sinh và Điểm GPA.

Trong đó:

- Họ tên không quá 30 chữ cái.
- Lớp theo đúng định dạng thường dùng ở PTIT
- Ngày sinh có đủ 3 phần ngày tháng năm nhưng có thể chưa đúng chuẩn dd/mm/yyyy.
- Điểm GPA đảm bảo trong thang điểm 4 với 2 nhiều nhất 2 số sau dấu phẩy.

Dữ liệu đảm bảo không có hai sinh viên nào có điểm GPA bằng nhau.

Output

Ghi ra danh sách lần lượt các sinh viên có đầy đủ Mã sinh viên, Họ tên, Lớp, Ngày sinh (đã chuẩn hóa), điểm GPA (với đúng 2 số sau dấu phẩy) đã được sắp xếp theo điểm GPA giảm dần.

Mỗi sinh viên ghi trên 1 dòng, mỗi thông tin cách nhau 1 khoảng trống.

Ví dụ

Input	Output
2	B20DCCN002 Nguyen Quang Hai D20DCCN02-B 01/09/2002 3.00
nguyen Van Nam	B20DCCN001 Nguyen Van Nam D20DCCN01-B 02/12/2002 2.17
D20DCCN01-B	
2/12/2002	
2.17	
Nguyen Quang hAi	
D20DCCN02-B	
1/9/2002	
3.0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    SinhVien ds[50];
    int N, i;
    cin >> N;
    for(i=0;i<N;i++){
        cin >> ds[i];
    }
    sapxep(ds, N);
    for(i=0;i<N;i++){
        cout << ds[i];
    }
    return 0;
}
```

188 Danh sách đối tượng nhân viên

Một nhân viên làm việc trong công ty được lưu lại các thông tin sau:

- Mã nhân viên: được gán tự động tăng, bắt đầu từ 00001
- Họ tên: Xâu ký tự không quá 40 chữ cái.
- Giới tính: Nam hoặc Nữ
- Ngày sinh: đúng theo chuẩn dd/mm/yyyy
- Địa chỉ: Xâu ký tự không quá 100 chữ cái
- Mã số thuế: Dãy số có đúng 10 chữ số
- Ngày ký hợp đồng: đúng theo chuẩn dd/mm/yyyy

Viết chương trình nhập danh sách nhân viên (không nhập mã) trong đó có sử dụng chèn toán tử nháp/xuất và in ra màn hình danh sách vừa nhập.

Input

Dòng đầu ghi số N là số nhân viên (không quá 40). Mỗi nhân viên ghi trên 6 dòng lần lượt ghi các thông tin theo thứ tự đã ghi trong đề bài. Không có mã nhân viên.

Output

Ghi ra danh sách đầy đủ nhân viên, mỗi nhân viên trên một dòng, các thông tin cách nhau đúng một khoảng trắng.

Ví dụ

Input
3
Nguyen Van A
Nam
10/22/1982
Mo Lao-Ha Dong-Ha Noi
8333012345
31/12/2013
Ly Thi B
Nu
10/15/1988
Mo Lao-Ha Dong-Ha Noi
8333012346
22/08/2011
Hoang Thi C
Nu
04/02/1981
Mo Lao-Ha Dong-Ha Noi
8333012347
22/08/2011
Output
00001 Nguyen Van A Nam 10/22/1982 Mo Lao-Ha Dong-Ha Noi 8333012345 31/12/2013
00002 Ly Thi B Nu 10/15/1988 Mo Lao-Ha Dong-Ha Noi 8333012346 22/08/2011
00003 Hoang Thi C Nu 04/02/1981 Mo Lao-Ha Dong-Ha Noi 8333012347 22/08/2011

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    NhanVien ds[50];
    int N;
    cin >> N;
    for(i=0;i<N;i++) cin >> ds[i];
    for(i=0;i<N;i++) cout << ds[i];
    return 0;
}
```

189 Sắp xếp danh sách đối tượng nhân viên

Một nhân viên làm việc trong công ty được lưu lại các thông tin sau:

- Mã nhân viên: được gán tự động tăng, bắt đầu từ 00001
- Họ tên: Xâu ký tự không quá 40 chữ cái.
- Giới tính: Nam hoặc Nữ
- Ngày sinh: đúng theo chuẩn dd/mm/yyyy
- Địa chỉ: Xâu ký tự không quá 100 chữ cái
- Mã số thuế: Dãy số có đúng 10 chữ số
- Ngày ký hợp đồng: đúng theo chuẩn dd/mm/yyyy

Viết chương trình nhập danh sách nhân viên (không nhập mã) trong đó có sử dụng chồng toán tử nhập/xuất, sau đó sắp xếp theo thứ tự ngày sinh từ già nhất đến trẻ nhất và in ra màn hình danh sách đối tượng nhân viên đã sắp xếp.

Input

Dòng đầu ghi số N là số nhân viên (không quá 40). Mỗi nhân viên ghi trên 6 dòng lần lượt ghi các thông tin theo thứ tự đã ghi trong đề bài. Không có mã nhân viên.

Output

Ghi ra danh sách đầy đủ nhân viên đã sắp xếp, mỗi nhân viên trên một dòng, các thông tin cách nhau đúng một khoảng trắng.

Ví dụ

Input
<pre>3 Nguyen Van A Nam 10/22/1982 Mo Lao-Ha Dong-Ha Noi 8333012345 31/12/2013 Ly Thi B Nu 10/15/1988 Mo Lao-Ha Dong-Ha Noi 8333012346 22/08/2011 Hoang Thi C Nu 04/02/1981 Mo Lao-Ha Dong-Ha Noi 8333012347 22/08/2011</pre>

Output
<pre>00003 Hoang Thi C Nu 04/02/1981 Mo Lao-Ha Dong-Ha Noi 8333012347 22/08/2011 00001 Nguyen Van A Nam 10/22/1982 Mo Lao-Ha Dong-Ha Noi 8333012345 31/12/2013 00002 Ly Thi B Nu 10/15/1988 Mo Lao-Ha Dong-Ha Noi 8333012346 22/08/2011</pre>

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

Bài tập này yêu cầu sử dụng hàm main cho sẵn như sau:

```
C/C++
int main(){
    NhanVien ds[50];
    int N,i;
    cin >> N;
    sapxep(ds, N);
    for(i=0;i<N;i++) cout << ds[i];
    return 0;
}
```

190 Sắp xếp sinh viên theo lớp

Thông tin về mỗi sinh viên gồm:

- Mã sinh viên: dây ký tự không có khoảng trắng (không quá 15). Đảm bảo không trùng nhau.
- Họ và tên: độ dài không quá 100
- Lớp: dây ký tự không có khoảng trắng (không quá 15)
- Email: dây ký tự không có khoảng trắng (không quá 15)

Hãy nhập danh sách sinh viên và sắp xếp theo lớp tăng dần (thứ tự từ điển)

Input

Dòng đầu ghi số sinh viên.

Mỗi sinh viên ghi trên 4 dòng lần lượt là: mã, họ tên, lớp, email.

Có không quá 1000 sinh viên trong danh sách.

Output

Ghi ra danh sách sinh viên đã sắp xếp theo lớp. Mỗi sinh viên trên một dòng, các thông tin cách nhau một khoảng trắng.

Nếu 2 sinh viên có cùng lớp thi sắp xếp theo mã tăng dần (thứ tự từ điển)

Ví dụ

Input	Output
4	
B16DCCN011	B15DCCN215 To Ngoc Hieu D15CNPM3 sv2@stu.ptit.edu.vn
Nguyen Trong Duc Anh	B15DCKT150 Nguyen Ngoc Son D15CQKT02-B sv3@stu.ptit.edu.vn
D16CNPM1	B15DCKT199 Nguyen Trong Tung D15CQKT03-B sv4@stu.ptit.edu.vn
sv1@stu.ptit.edu.vn	B16DCCN011 Nguyen Trong Duc Anh D16CNPM1 sv1@stu.ptit.edu.vn
B15DCCN215	
To Ngoc Hieu	
D15CNPM3	
sv2@stu.ptit.edu.vn	
B15DCKT150	
Nguyen Ngoc Son	
D15CQKT02-B	
sv3@stu.ptit.edu.vn	
B15DCKT199	
Nguyen Trong Tung	
D15CQKT03-B	
sv4@stu.ptit.edu.vn	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

191 Sắp xếp theo mã sinh viên

Thông tin về mỗi sinh viên gồm:

- Mã sinh viên: dãy ký tự không có khoảng trắng (không quá 15). Đảm bảo không trùng nhau.
- Họ và tên: độ dài không quá 100
- Lớp: dãy ký tự không có khoảng trắng (không quá 15)
- Email: dãy ký tự không có khoảng trắng (không quá 15)

Hãy nhập danh sách sinh viên và sắp xếp theo mã sinh viên tăng dần (thứ tự từ điển)

Input

Mỗi sinh viên ghi trên 4 dòng lân lượt là: mã, họ tên, lớp, email.

Không cho biết số sinh viên nhưng dữ liệu đảm bảo là chẵn lẻ 4 dòng.

Có không quá 1000 sinh viên trong danh sách.

Output

Ghi ra danh sách sinh viên đã sắp xếp theo mã. Mỗi sinh viên trên một dòng, các thông tin cách nhau một khoảng trắng.

Ví dụ

Input	Output
B16DCCN011	B15DCCN215 To Ngoc Hieu D15CNPM3 sv2@stu.ptit.edu.vn
Nguyen Trong Duc Anh	B15DCKT150 Nguyen Ngoc Son D15CQKT02-B sv3@stu.ptit.edu.vn
D16CNPM1	B15DCKT199 Nguyen Trong Tung D15CQKT03-B sv4@stu.ptit.edu.vn
sv1@stu.ptit.edu.vn	B16DCCN011 Nguyen Trong Duc Anh D16CNPM1 sv1@stu.ptit.edu.vn
B15DCCN215	
To Ngoc Hieu	
D15CNPM3	
sv2@stu.ptit.edu.vn	
B15DCKT150	
Nguyen Ngoc Son	
D15CQKT02-B	
sv3@stu.ptit.edu.vn	
B15DCKT199	
Nguyen Trong Tung	
D15CQKT03-B	
sv4@stu.ptit.edu.vn	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

192 Liệt kê sinh viên theo lớp

Thông tin về mỗi sinh viên gồm:

- Mã sinh viên: dây ký tự không có khoảng trắng (không quá 15). Đảm bảo không trùng nhau.
- Họ và tên: độ dài không quá 100
- Lớp: dây ký tự không có khoảng trắng (không quá 15)
- Email: dây ký tự không có khoảng trắng (không quá 15)

Hãy nhập danh sách sinh viên và liệt kê sinh viên theo lớp

Input

Dòng đầu ghi số sinh viên (không quá 1000)

Mỗi sinh viên ghi trên 4 dòng lân lượt là: mã, họ tên, lớp, email.

Sau đó sẽ có giá trị số Q là số truy vấn

Tiếp theo là Q dòng, mỗi dòng ghi một lớp

Output

Với mỗi truy vấn, liệt kê danh sách sinh viên của lớp đó theo mẫu như trong ví dụ. Mỗi sinh viên ghi trên một dòng, các thông tin cách nhau một khoảng trắng. Thứ tự sinh viên vẫn giữ nguyên như thứ tự ban đầu.

Ví dụ

Input	Output
4	DANH SACH SINH VIEN LOP D15CQKT02-B:
B16DCCN011	B15DCKT150 Nguyen Ngoc Son D15CQRT02-B sv3@stu.ptit.edu.vn
Nguyen Trong Duc Anh	B15DCKT199 Nguyen Trong Tung D15CQRT02-B sv4@stu.ptit.edu.vn
D16CNPM1	
sv1@stu.ptit.edu.vn	
B15DCCN215	
To Ngoc Hieu	
D15CNPM3	
sv2@stu.ptit.edu.vn	
B15DCKT150	
Nguyen Ngoc Son	
D15CQKT02-B	
sv3@stu.ptit.edu.vn	
B15DCKT199	
Nguyen Trong Tung	
D15CQRT02-B	
sv4@stu.ptit.edu.vn	
1	
D15CQKT02-B	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

193 Liệt kê sinh viên theo khóa

Thông tin về mỗi sinh viên gồm:

- Mã sinh viên: dãy ký tự không có khoảng trống (không quá 15). Đảm bảo không trùng nhau.
- Họ và tên: độ dài không quá 100
- Lớp: dãy ký tự không có khoảng trống (không quá 15)
- Email: dãy ký tự không có khoảng trống (không quá 15)

Hãy nhập danh sách sinh viên và liệt kê sinh viên theo khóa học. Chú ý: dữ liệu khóa học thể hiện qua hai chữ số thứ 2 và thứ 3 trong lớp sinh viên.

Input

Dòng đầu ghi số sinh viên (không quá 1000)

Mỗi sinh viên ghi trên 4 dòng lân lượt là: mã, họ tên, lớp, email.

Sau đó sẽ có giá trị số Q là số truy vấn

Tiếp theo là Q dòng, mỗi dòng ghi năm bắt đầu khóa học theo định dạng yyyy

Output

Với mỗi truy vấn, liệt kê danh sách sinh viên của khóa đó theo mẫu như trong ví dụ. Mỗi sinh viên ghi trên một dòng, các thông tin cách nhau một khoảng trống. Thứ tự sinh viên vẫn giữ nguyên như thứ tự ban đầu.

Ví dụ

Input	Output
4	DANH SACH SINH VIEN KHOA 2015:
B16DCCN011	B15DCCN215 To Ngoc Hieu D15CNPM3 sv2@stu.ptit.edu.vn
Nguyen Trong Duc Anh	B15DCKT150 Nguyen Ngoc Son D15CQKT02-B sv3@stu.ptit.edu.vn
D16CNPM1	B15DCKT199 Nguyen Trong Tung D15CQKT02-B sv4@stu.ptit.edu.vn
sv1@stu.ptit.edu.vn	
B15DCCN215	
To Ngoc Hieu	
D15CNPM3	
sv2@stu.ptit.edu.vn	
B15DCKT150	
Nguyen Ngoc Son	
D15CQKT02-B	
sv3@stu.ptit.edu.vn	
B15DCKT199	
Nguyen Trong Tung	
D15CQKT02-B	
sv4@stu.ptit.edu.vn	
1	
2015	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

194 Liệt kê sinh viên theo ngành

Thông tin về mỗi sinh viên gồm:

- Mã sinh viên: dãy ký tự không có khoảng trắng (không quá 15). Đảm bảo không trùng nhau.
- Họ và tên: độ dài không quá 100
- Lớp: dãy ký tự không có khoảng trắng (không quá 15)
- Email: dãy ký tự không có khoảng trắng (không quá 50)

Hãy nhập danh sách sinh viên và liệt kê sinh viên theo ngành học. Chú ý: dữ liệu ngành học thể hiện qua 4 chữ cái từ thứ 4 đến thứ 7 trong mã sinh viên.

Input

Dòng đầu ghi số sinh viên (không quá 1000)

Mỗi sinh viên ghi trên 4 dòng lần lượt là: mã, họ tên, lớp, email.

Sau đó sẽ có giá trị số Q là số truy vấn

Tiếp theo là Q dòng, mỗi dòng ghi ngành đào tạo. Chỉ có các ngành đào tạo trong danh sách sau (trong Input sẽ không có dấu):

- Kế toán** – mã sinh viên có cụm ký tự DCKT
- Công nghệ thông tin** – mã sinh viên có cụm DCCN – trừ đi các sinh viên lớp bắt đầu bằng chữ E
- An toàn thông tin** – mã sinh viên có cụm DCAT – trừ các sinh viên lớp bắt đầu bằng chữ E
- Viễn thông** – mã sinh viên có cụm DCVT
- Điện tử** - mã sinh viên có cụm DCDT

Output

Với mỗi truy vấn, liệt kê danh sách sinh viên của khóa đó theo mẫu như trong ví dụ. Mỗi sinh viên ghi trên một dòng, các thông tin cách nhau một khoảng trắng. Thứ tự sinh viên vẫn giữ nguyên như thứ tự ban đầu.

Ví dụ

Input	Output
4 B16DCCN011 Nguyen Trong Duc Anh D16CNPM1 sv1@stu.ptit.edu.vn B15DCCN215 To Ngoc Hieu D15CNPM3 sv2@stu.ptit.edu.vn B15DCKT150 Nguyen Ngoc Son D15CQKT02-B sv3@stu.ptit.edu.vn B15DCKT199 Nguyen Trong Tung D15CQKT02-B sv4@stu.ptit.edu.vn 1 Ke toan	DANH SACH SINH VIEN NGANH KE TOAN: B15DCKT150 Nguyen Ngoc Son D15CQKT02-B sv3@stu.ptit.edu.vn B15DCKT199 Nguyen Trong Tung D15CQKT02-B sv4@stu.ptit.edu.vn

195 Sắp xếp danh sách giảng viên

TÌM KIẾM GIẢNG VIÊN

Bài làm tốt nhất

Thông tin về giảng viên Khoa CNTT ban đầu chỉ có họ tên và Bộ môn. Mã giảng viên sẽ tự động điền tăng dần, tính từ GV01.

Hãy tìm kiếm giảng viên theo tên (*yêu cầu tìm gần đúng – tức là trong tên giảng viên xuất hiện từ hoặc cụm từ khóa, không phân biệt chữ hoa chữ thường*).

Input

Dòng đầu ghi số giảng viên.

Mỗi giảng viên ghi trên 2 dòng gồm họ tên (không quá 50 ký tự) và Bộ môn (không quá 30 ký tự).

Tiếp theo là một dòng ghi số Q là số truy vấn.

Mỗi truy vấn là một từ khóa cần tìm.

Output

Danh sách các giảng viên tìm được theo từ khóa, trong đó mỗi giảng viên ghi trên một dòng. Mã được tự động điền theo thứ tự nhập, bộ môn được viết tắt theo các chữ cái đầu của từng từ và ở dạng in hoa.

Thứ tự giảng viên của mỗi bộ môn được liệt kê theo đúng thứ tự ban đầu.

Ví dụ

Input	Output
3	DANH SACH GIANG VIEN THEO TU KHOA aN:
Nguyen Manh Son	GV01 Nguyen Manh Son CNPM
Cong nghe phan mem	GV03 Dang Minh Tuan ATTT
Vu Hoai Nam	
Khoa hoc may tinh	
Dang Minh Tuan	
An toan thong tin	
1	
aN	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

196 Danh sách giảng viên theo bộ môn

Thông tin về giảng viên Khoa CNTT ban đầu chỉ có họ tên và Bộ môn. Mã giảng viên sẽ tự động điền tăng dần, tính từ GV01.

Hãy liệt kê danh sách giảng viên của Bộ môn được yêu cầu.

Input

Dòng đầu ghi số giảng viên.

Mỗi giảng viên ghi trên 2 dòng gồm họ tên (không quá 50 ký tự) và Bộ môn (không quá 30 ký tự).

Tiếp theo là một dòng ghi số Q là số truy vấn.

Mỗi truy vấn là tên một bộ môn trên một dòng.

Output

Danh sách các giảng viên của bộ môn theo từng truy vấn, trong đó mỗi giảng viên ghi trên một dòng. Mã được tự động điền theo thứ tự nhập, bộ môn được viết tắt theo các chữ cái đầu của từng từ và ở dạng in hoa.

Thứ tự giảng viên của mỗi bộ môn được liệt kê theo đúng thứ tự ban đầu.

Ví dụ

Input	Ouput
3	DANH SACH GIANG VIEN BO MON CNPM:
Nguyen Manh Son	GV01 Nguyen Manh Son CNPM
Cong nghe phan mem	
Vu Hoai Nam	
Khoa hoc may tinh	
Dang Minh Tuan	
An toan thong tin	
1	
Cong nghe phan mem	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

197 Tìm kiếm giảng viên

Thông tin về giảng viên Khoa CNTT ban đầu chỉ có họ tên và Bộ môn. Mã giảng viên sẽ tự động điền tăng dần, tính từ GV01.

Hãy tìm kiếm giảng viên theo tên (*yêu cầu tìm gần đúng – tức là trong tên giảng viên xuất hiện từ hoặc cụm từ khóa, không phân biệt chữ hoa chữ thường*).

Input

Dòng đầu ghi số giảng viên.

Mỗi giảng viên ghi trên 2 dòng gồm họ tên (không quá 50 ký tự) và Bộ môn (không quá 30 ký tự).

Tiếp theo là một dòng ghi số Q là số truy vấn.

Mỗi truy vấn là một từ khóa cần tìm.

Output

Danh sách các giảng viên tìm được theo từ khóa, trong đó mỗi giảng viên ghi trên một dòng. Mã được tự động điền theo thứ tự nhập, bộ môn được viết tắt theo các chữ cái đầu của từng từ và ở dạng in hoa.

Thứ tự giảng viên của mỗi bộ môn được liệt kê theo đúng thứ tự ban đầu.

Ví dụ

Input	Ouput
3	DANH SACH GIANG VIEN THEO TU KHOA aN:
Nguyen Manh Son	GV01 Nguyen Manh Son CNPM
Cong nghe phan mem	GV03 Dang Minh Tuan ATTT
Vu Hoai Nam	
Khoa hoc may tinh	
Dang Minh Tuan	
An toan thong tin	
1	
aN	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

198 Danh sách doanh nghiệp nhận sinh viên thực tập – 1

Để chuẩn bị cho đợt thực tập tốt nghiệp của sinh viên năm cuối, Khoa CNTT1 trao đổi với các doanh nghiệp đối tác và chốt số lượng sinh viên có thể nhận thực tập.

Hãy sắp xếp các doanh nghiệp theo số lượng sinh viên có thể nhận giảm dần.

Input

Dòng đầu ghi số doanh nghiệp.

Mỗi doanh nghiệp ghi trên 3 dòng:

- Mã doanh nghiệp (xâu ký tự không có dấu cách, độ dài không quá 10)
- Tên doanh nghiệp (xâu ký tự độ dài không quá 150)
- Số sinh viên có thể nhận: giá trị nguyên không quá 1000

Output

Ghi ra danh sách đã được sắp xếp theo số lượng giảm dần, mỗi thông tin ghi trên một dòng. Trong trường hợp cùng số lượng thì sắp xếp theo mã doanh nghiệp (thứ tự từ điển tăng dần).

Ví dụ

Input
4
VIETTEL
TAP DOAN VIEN THONG QUAN DOI VIETTEL
40
FSOFT
CONG TY TNHH PHAN MEM FPT – FPT SOFTWARE
300
VNPT
TAP DOAN BUU CHINH VIEN THONG VIET NAM
200
SUN
SUN*
50
Output
FSOFT CONG TY TNHH PHAN MEM FPT – FPT SOFTWARE 300
VNPT TAP DOAN BUU CHINH VIEN THONG VIET NAM 200
SUN SUN* 50
VIETTEL TAP DOAN VIEN THONG QUAN DOI VIETTEL 40

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

199 Danh sách doanh nghiệp nhận sinh viên thực tập – 2

Đề chuẩn bị cho đợt thực tập tốt nghiệp của sinh viên năm cuối, Khoa CNTT1 trao đổi với các doanh nghiệp đối tác và chốt số lượng sinh viên có thể nhận thực tập.

Hãy lọc ra các doanh nghiệp nhận số lượng sinh viên trong đoạn [a,b].

Input

Dòng đầu ghi số doanh nghiệp.

Mỗi doanh nghiệp ghi trên 3 dòng:

- Mã doanh nghiệp (xâu ký tự không có dấu cách, độ dài không quá 10)
- Tên doanh nghiệp (xâu ký tự độ dài không quá 150)
- Số sinh viên có thể nhận: giá trị nguyên không quá 1000

Tiếp theo là một dòng ghi số truy vấn Q. Mỗi truy vấn là 2 số nguyên a, b viết trên một dòng trong đó $a < b$ và dữ liệu đảm bảo luôn có ít nhất 1 doanh nghiệp nhận số lượng sinh viên trong đoạn [a,b].

Output

Ghi ra danh sách đã lọc trong đoạn [a,b] và được sắp xếp theo số lượng giảm dần, mỗi thông tin ghi trên một dòng. Trong trường hợp cùng số lượng thì sắp xếp theo mã doanh nghiệp (thứ tự từ điển tăng dần).

Ví dụ

Input
<pre>4 VIETTEL TAP DOAN VIEN THONG QUAN DOI VIETTEL 40 FSOFT CONG TY TNHH PHAN MEM FPT - FPT SOFTWARE 300 VNPT TAP DOAN BUU CHINH VIEN THONG VIET NAM 200 SUN SUN* 50 1 30 50</pre>

Output
<pre>DANH SACH DOANH NGHIEP NHAN TU 30 DEN 50 SINH VIEN: SUN SUN* 50 VIETTEL TAP DOAN VIEN THONG QUAN DOI VIETTEL 40</pre>

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

200 Liệt kê tổ hợp

LIỆT KÊ TỔ HỢP

Bài làm tốt nhất

Cho số tự nhiên N và số K. Hãy đưa ra các tổ hợp chập K của 1, 2, ..., N.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là bộ đôi N, K được viết trên 1 dòng.
- T, N, K thỏa mãn ràng buộc : $1 \leq T, K, N \leq 20$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
1	123 124 125 134 135 145 234 235 245 345
5 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

201 Liệt kê hoán vị

LIỆT KÊ HOÁN VỊ

Bài làm tốt nhất

Cho số tự nhiên N. Hãy đưa ra các hoán vị của 1, 2, ..., N.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số N được viết trên 1 dòng.
- T, N thỏa mãn ràng buộc : $1 \leq T, N \leq 20$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
1	123 132 213 231 312 321
3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

202 Dãy con tăng dài nhất

DÃY CON TĂNG DÀI NHẤT

Bài làm tốt nhất

Cho mảng A[] gồm n số được sinh ra ngẫu nhiên. Hãy tìm độ dài dãy tăng dài nhất các phần tử của mảng. Chú ý, dãy con của mảng không nhất thiết là liên tục. Hai phần tử giống nhau của mảng ta chỉ xem là 1 trong độ dài dãy tăng. Ví dụ với mảng A[] = {5, 8, 3, 7, 9, 1}, ta có kết quả là 3.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $0 \leq A[i] \leq 10^3$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	6
16	3
0 8 4 12 2 10 6 14 1 9 5 13 3 11 7 15	
6	
5 8 3 7 9 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

203. DÃY SỐ CATALAN (*)

Bạn được cho một dãy số được định nghĩa truy hồi như sau:

$$C_n = C_{n-1}C_0 + C_{n-2}C_1 + \dots + C_0C_{n-1}$$

Với $C_0 = C_1 = 1$ thì dãy trên chính là dãy Catalan, có thể được tính theo cách khác:

$$C_n = \frac{(2n)!}{n! * (n+1)!}$$

Nhiệm vụ của bạn là tính số S_n biết:

$$S_n = \sum_{k=0}^n C_k C_{n-k}$$

Input

Dòng duy nhất chứa số n ($0 \leq n \leq 5000$).

Output

Dòng duy nhất chứa kết quả tìm được.

Example**Input:**

3

Output:

14

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

204. TÍNH SỐ TỔ HỢP (*)

TÍNH SỐ TỔ HỢP

Bài làm tốt nhất

Cho hai số n và r , hãy tìm $C(n, r) \% P$. Trong đó, $P = 10^9 + 7$.

Input:

- Đòng đầu tiên đưa vào số lượng test T .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test là bộ đôi n, r . Các số được viết cách nhau một vài khoảng trắng.
- T, n, r thỏa mãn ràng buộc: $1 \leq T \leq 100; 0 \leq n \leq 10^3; 1 \leq r \leq 800$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3
3 2	6
4 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

205. PHÉP CHIA DƯ CỦA TÍCH HAI SỐ

PHÉP CHIA DƯ CỦA TÍCH HAI SỐ

Bài làm tốt nhất

Cho ba số a, b, c có thể lớn cỡ 10^{16} . Nhiệm vụ của bạn là tính $(a * b) \% c$.

Input:

- Đòng đầu tiên đưa vào số lượng test T .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi test trên một dòng đưa vào ba số a, b, c .
- T, a, b, c thỏa mãn ràng buộc: $1 \leq T \leq 100; 0 \leq a, b, c \leq 10^{16}$.

Output:

- Đưa ra số kết quả mỗi test theo từng dòng.

Input:	Output:
1	2
8 4 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

206. TỔNG LỚN NHẤT CỦA DÃY CON KHÔNG KỀ NHAU (*)

TỔNG LỚN NHẤT CỦA DÃY CON KHÔNG KẾ NHAU

Bài làm tốt nhất

Cho mảng $A[]$ gồm n số nguyên dương. Hãy tìm tổng lớn nhất của dãy con thỏa mãn ràng buộc không có hai phần tử kề nhau thuộc một dãy con. Ví dụ với mảng $A[] = \{3, 2, 7, 10\}$ ta nhận được kết quả là $13 = 10 + 3$. Với mảng $A[] = \{3, 2, 5, 10, 7\}$ ta có kết quả là $15 = 3 + 5 + 7$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng $A[]$; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trắng.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^5$; $1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	110
6	20
5 5 10 100 10 5	
3	
1 20 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

207. TỔNG LỚN NHẤT CỦA DÃY CON LIÊN TỤC

TỔNG LỚN NHẤT CỦA DÃY CON LIÊN TỤC

Bài làm tốt nhất

Cho mảng $A[]$ gồm n phần tử. Nhiệm vụ của bạn là tìm tổng lớn nhất các dãy con liên tục của mảng $A[]$. Ví dụ với $A[] = \{-2, -3, 4, -1, -2, 1, 5, -3\}$ ta có câu trả lời là 7 tương ứng với tổng lớn nhất của dãy con liên tục $\{4, -1, -2, 1, 5\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n ; dòng tiếp theo là n số $A[i]$ của mảng $A[]$; các số được viết cách nhau một vài khoảng trắng.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^6$; $10^{-6} \leq A[i] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	9
5	-1
1 2 3 -2 5	
4	
-1 -2 -3 -4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

208. DÃY CON TRUNG BÌNH LỚN NHẤT

Cho mảng A[] gồm n số và số nguyên dương k. Hãy tìm dãy con liên tục độ dài k có giá trị trung bình các phần tử lớn nhất. Ví dụ với A[] = { 1, 12, -5, -6, 50, 3} và k = 4 ta có câu trả lời là {12, -5, -6, 50} có trung bình các phần tử lớn nhất là $(12-5-6+30)/4=7.75$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n và số k; dòng tiếp theo là n số A[i] của mảng A[]; các số được viết cách nhau một vài khoảng trắng.
- T, n, k, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq k \leq n \leq 10^3$; $-10^3 \leq A[i] \leq 10^3$.

Output:

- Dưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	15 20
5 2	34 56
10 4 5 15 20	
4 2	
-12 34 56 7	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

209. CỘNG 1 VÀ NHÂN ĐÔI

Cho mảng A[] gồm n số nguyên dương. Mảng A[] được gọi là mảng mục tiêu. Hãy tạo một mảng bắt đầu S[] gồm n phần tử có các phần tử ban đầu được thiết lập là 0. Chỉ được phép thực hiện hai thao tác dưới đây:

- Thao tác 1 (Increment Operation): tăng giá trị của 1 phần tử bất kỳ lên 1 đơn vị.
- Thao tác 2 (Double Operation): nhân toàn bộ phần tử trong mảng với 2.

Hãy tìm số các ít nhất để dịch chuyển mảng bắt đầu S[] thành mảng mục tiêu A[]. Ví dụ với A[] = { 16, 16, 16} ta cần thực hiện ít nhất 7 thao tác như sau:

- Sử dụng 3 thao tác Increment để biến : S[0]=1, S[1]=1, S[2]=1.
- Sử dụng 4 thao tác Double để biến : S[0]=16, S[1]=16, S[2]=16.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n; dòng tiếp theo là n số A[i] của mảng A[]; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A[i] \leq 10^3$.

Output:

- Dưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7
3	4
16 16 16	
2	
2 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

210. GIẢI MÃ TĂNG GIẢM

Cho mảng A[] chỉ bao gồm các ký tự I hoặc D. Ký tự I được hiểu là tăng (Increasing) ký tự D được hiểu là giảm (Decreasing). Sử dụng các số từ 1 đến 9, hãy đưa ra số nhỏ nhất được đoán nhận từ mảng A[]. Chủ ý, các số không được phép lặp lại. Dưới đây là một số ví dụ mẫu:

- A[] = "T" : số tăng nhỏ nhất là 12.
- A[] = "D" : số giảm nhỏ nhất là 21
- A[] = "DD" : số giảm nhỏ nhất là 321
- A[] = "DDIDDDII": số thỏa mãn 321654798

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ID
- T, Length(A) thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{Length}(A) \leq 9$.

Output:

- Dưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
4	12
I	21
D	321
DD	321654798
DDIDDDII	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

211. TÍCH LỚN NHẤT CỦA DÃY CON LIÊN TỤC

TÍCH LỚN NHẤT CỦA DÂY CON LIỀN TỤC

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử gồm các số âm và dương. Hãy tìm giá trị lớn nhất tích các phần tử của tất cả các dây con liên tục trong mảng A[]. Ví dụ với mảng A[] = {6, -3, -10, 0, 2} ta có kết quả là 180 trong ứng với tích các phần tử của dây dài con {6, -3, -10}.

Input:

- Đòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào số lượng phần tử của mảng N; phần thứ hai đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 100$; $1 \leq A[i] \leq 200$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
3	180
5	120
6 -3 -10 0 2	288
6	
2 3 4 5 -1 0	
10	
8 -2 -2 0 8 0 -6 -8 -6 -1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

212. LOẠI BỎ 100

LOẠI BỎ 100

Bài làm tốt nhất

Cho xâu ký tự S chỉ bao gồm các ký tự '0' và '1'. Nhiệm vụ của bạn là loại bỏ các xâu con "100" trong S và đưa ra độ dài lớn nhất xâu con bị loại bỏ. Ví dụ S = "1011110000" ta nhận được kết quả là 6 vì ta cần loại bỏ xâu "110000" có độ dài 6.

Input:

- Đòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự nhị phân S được viết trên một dòng.
- T, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{Length}(S) \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3
010010	6
1011110000	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

213. TÍNH TỔNG

TÍNH TỔNG

Bài làm tốt nhất

Cho file dữ liệu dạng văn bản DATA.in có thẻ chứa các số và ký tự.

Hãy lọc ra các số nguyên int trong file và tính tổng các số đó.

Chú ý: file dữ liệu có rất nhiều dòng với rất nhiều số và ký tự xen kẽ nhau. Chỉ tính tổng các số thỏa mãn điều kiện là số kiểu int.

Input

File văn bản DATA.in có không quá 1000 dòng.

Output

Ghi ra giá trị tổng các số tính được.

Ví dụ

DATA.in	Output
12 3 4 5 6 7	58
Aaa 1 1 Bbb XXX yyy 5 5	
99999999999999999999999999999999	
9	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

214. SỐ KHÁC NHAU TRONG FILE

Cho file văn bản DATA.in có không quá 100000 số nguyên dương, giá trị các số nhỏ hơn 1000.

Hãy liệt kê các số khác nhau xuất hiện trong file và số lần xuất hiện của từng số đó.

Input

File DATA.in có không quá 100000 số nguyên dương.

Output

Ghi ra các số khác nhau và số lần xuất hiện theo thứ tự tăng dần

Ví dụ

DATA.in	Output
17 20 25 20 15 10 24 17 25 17 22 11 23 18	10 2
14 25 12 10 12 17 21 25	11 1
	12 2
	14 1
	15 1
	17 4
	18 1
	20 2
	21 1
	22 1
	23 1
	24 1
	25 4

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

215. LIỆT KÊ TỪ KHÁC NHAU

Cho file văn bản VANBAN.in.

Một từ được định nghĩa là một dãy ký tự liên tiếp không có khoảng trắng, dấu tab hay dấu xuống dòng. Tạm thời chưa xét đến các dấu câu trong bài toán này.

Hãy chuyển tất cả các từ về dạng chữ thường sau đó liệt kê các từ khác nhau xuất hiện trong file VANBAN.in theo thứ tự từ điển.

Input

File VANBAN.in có không quá 200 dòng.

Output

Ghi ra danh sách các từ khác nhau xuất hiện trong file. Mỗi từ trên một dòng theo thứ tự từ điển.

Ví dụ

VANBAN.in	Output
lap trinh Huong doi tuong	doi
lap trinh Huong thanh phan	huong
	lap
	phan
	thanh
	trinh
	tuong

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

216. HỢP VÀ GIAO CỦA HAI FILE VĂN BẢN

Cho hai file DATA1.in và DATA2.in.

Một từ được định nghĩa là một dãy ký tự liên tiếp không có khoảng trắng, dấu tab hay dấu xuống dòng. Tạm thời chưa xét đến các dấu câu trong bài toán này.

Hãy viết chương trình liệt kê hợp và giao của hai tập từ khác nhau trong hai file.

Các từ được liệt kê theo thứ tự từ điển.

Input

Hai file văn bản DATA1.in và DATA2.in, có không quá 200 dòng.

Output

Hợp và giao của hai tập từ khác nhau trong hai file ban đầu. Mỗi tập trên một dòng, các từ liệt kê theo thứ tự từ điển và cách nhau đúng một khoảng trắng.

Ví dụ

DATA1.in	Output
lap trinh huong	ban c++ co doi huong lap ngon ngu phan thanh trinh tuong
doi tuong	huong lap trinh
ngon ngu lap	
trinh C++	
DATA2.in	
lap trinh co	
ban	
lap trinh huong	
thanh phan	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 100000 Kb

217. LỚP INTSET

Trong lý thuyết tập hợp, một tập hợp chỉ được phép chứa các giá trị phân biệt và luôn luôn lưu các giá trị theo thứ tự tăng dần.

Khai báo lớp IntSet và viết các phương thức để thực hiện các thao tác trên tập hợp số nguyên. Sử dụng lớp IntSet để in ra tập hợp các số nguyên là giao của hai tập số trong 2 dãy ban đầu.

Input - file văn bản DATA.in

Dòng đầu ghi 2 số n và m ($1 < n, m < 100$).

Dòng thứ 2 ghi n số của a[].

Dòng thứ 3 ghi m số của b[].

Các số đều dương và nhỏ hơn 1000.

Output

Ghi tập giao của A và B trên một dòng theo thứ tự từ nhỏ đến lớn.

Ví dụ

DATA.in	Output
5 6	3 4 5
1 2 3 4 5	
3 4 5 6 7 8	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

218. THỐNG KÊ PHẦN TỬ

Cho 1 danh sách tuyến tính ds theo khai báo sau chứa n số nguyên:

```
const int MAXLIST = 10000
```

```
typedef struct list
```

```
{ int n;
```

```
int nodes[MAXLIST]; };
```

```
list ds;
```

Viết chương trình con Thống kê số lần xuất hiện của từng số trong danh sách ds, và in ra màn hình mỗi số trên 1 dòng theo ví dụ sau:

Dãy số ds: 5 5 3 4 3 2 5

Kết quả in trên màn hình :

5 3

3 2

4 1

2 1

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

219. PHÂN TÍCH THỪA SỐ NGUYÊN TỐ – VER 2

Cho số nguyên dương N. Hãy phân tích N thành tích của các thừa số nguyên tố.

Input:

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

Mỗi test gồm một số nguyên dương N ($2 \leq N \leq 10^{14}$).

Output:

Với mỗi test, liệt kê các thừa số và số mũ theo thứ tự tăng dần. Sau mỗi test, in ra một dấu xuống dòng.

Example:

Input	Output
2	2
4	
168	2 3
	3 1
	7 1

Giới hạn thời gian: 1s
Giới hạn bộ nhớ: 524288 Kb

220. GHÉP CẤP**GHÉP CẤP**

Bài làm tốt nhất

Cho số nguyên dương N, đếm số cách chia các số từ 1 đến 2N thành N nhóm, mỗi nhóm gồm 2 số mà hiệu hai số trong một nhóm bằng hiệu hai số trong nhóm khác.

Input:

Dòng đầu tiên chứa số lượng bộ test T.

Mỗi test gồm 1 số nguyên dương N.

Giới hạn:

Subtask 1 (50%): $T, N \leq 10000$

Subtask 2 (50%): $T \leq 5 \cdot 10^5, N \leq 10^6$.

Output:

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

Example:

Input	Output
2	1
1	2
2	

Giải thích test 2: Có 2 cách chia nhóm là:

(1, 2) và (3, 4)

(1, 3) và (2, 4)

Giới hạn thời gian: 1s
Giới hạn bộ nhớ: 200000 Kb

221. PTTT2025R1B – NGUYÊN TỐ CÙNG NHAU

Cho dãy số nguyên A[] có N phần tử và số nguyên dương M. Nhiệm vụ của bạn là hãy tìm các giá trị K trong khoảng [1, M] sao cho K nguyên tố cùng nhau với mọi phần tử của dãy số A.

Hay nói cách khác, với mỗi giá trị K trong khoảng [1, M], hãy kiểm tra xem $\text{GCD}(A[i], K) = 1$ với mọi chỉ số của dãy số A[] (tức $1 \leq i \leq N$) hay không?

Input:

Dòng đầu tiên gồm 2 số nguyên dương N và M.

Dòng tiếp theo gồm N số nguyên A[i] ($1 \leq i \leq N, A[i] \leq 100000$).

Output:

Dòng đầu tiên in ra số lượng các số thỏa mãn.

Dòng tiếp theo liệt kê các số K tìm được theo thứ tự tăng dần.

Test ví dụ:

Input:	Output:
9 12	3
6 1 5	1 7 11

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 524288 Kb

222. ƯỚC SỐ CỦA GIAI THỪA

Cho số tự nhiên N và số nguyên tố P. Nhiệm vụ của bạn là tìm số x lớn nhất để $N!$ chia hết cho p^x . Ví dụ với $N=7$, $p=3$ thì $x=2$ là số lớn nhất để $7!$ chia hết cho 3^2 .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là cặp số N, p được viết cách nhau một vài khoảng trắng.
- T, N, p thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 10^5$; $2 \leq p \leq 5000$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
3	9
62 7	73
76 2	0
3 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

223. LIỆT KÊ SỐ CÓ BA ƯỚC SỐ TRONG KHOẢNG

Cho hai số L, R. Nhiệm vụ của bạn là hãy đếm tất cả các số có đúng ba ước số trong khoảng [L, R]. Ví dụ L = 1, R = 10, ta có kết quả là 2 vì chỉ có số 3 và 9 là có đúng 3 ước số.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là cặp số L, R.
- T, N thỏa mãn ràng buộc $1 \leq T \leq 100$; $1 \leq L, R \leq 10^{12}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
1 10	78498
1 100000000000	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

224. SỐ SMITH

Cho số tự nhiên N. Nhiệm vụ của bạn là hãy kiểm tra N có phải là số Smith hay không. Một số được gọi là số Smith nếu N không phải là số nguyên tố và có tổng các chữ số của N bằng tổng các chữ số của các ước số nguyên tố của N. Ví dụ N = 666 có các ước số nguyên tố là 2, 3, 3, 37 có tổng các chữ số là 18.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên N.
- T, N thỏa mãn ràng buộc $1 \leq T \leq 100$; $1 \leq N \leq 100000$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	YES
4	YES
666	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

225. ĐẾM SỐ NGHIỆM CỦA PHƯƠNG TRÌNH ĐỒNG DƯ'

ĐIỂM SỐ NGHIỆM CỦA PHƯƠNG TRÌNH ĐỒNG ĐƯ**Bài làm tốt nhất**

- Tìm số nghiệm của phương trình đồng dư $x^2 \equiv 1 \pmod{p}$ trong khoảng $[1, b]$. Ví dụ với $b=5, p=7$ ta tìm được $x = 1 \in [1, 5]$ để $x^2 \equiv 1 \pmod{7}=1$. Với $b=8, p=6$ ta tìm được $x = \{1, 5, 7\}$ để $x^2 \equiv 1 \pmod{6}$.
- $6\%1 = 38\%1 = 34\%1 = 0; 6\%2 = 38\%2 = 34\%2 = 0; 6\%4 = 38\%4 = 34\%4 = 2;$
- Input:
- Dòng đầu tiên đưa vào số lượng test T.
 - Những dòng kế tiếp đưa vào các bộ test. Mỗi test là bộ đôi b, p. Các số được viết cách nhau một vài khoảng trắng.
 - T, b, p thỏa mãn ràng buộc: $1 \leq T \leq 100; 0 \leq b \leq 10^5; 1 \leq p \leq 10^5$.
- Output:
- Đưa ra số các số kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
5 7	3
8 6	

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

226. KHOẢNG CÁCH XA NHẤT**KHOẢNG CÁCH XA NHẤT****Bài làm tốt nhất**

- Cho mảng A[] gồm n số nguyên dương. Hãy tìm hiệu lớn nhất của i-j thỏa mãn ràng buộc $A[i] <= A[j]$. Ví dụ với mảng A[] = {34, 8, 10, 3, 2, 80, 30, 33, 1} ta nhận được kết quả là 6 vì $A[1] < A[7]$ và $7-1 = 6$ là lớn nhất.
- Input:
- Dòng đầu tiên đưa vào số lượng bộ test T.
 - Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trắng.
 - T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq n \leq 10^7; 1 \leq A[i] \leq 10^8$.
- Output:
- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	6
9	

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

227. ĐỒNG ĐƯ VỚI K**ĐỒNG ĐƯ VỚI K****Bài làm tốt nhất**

- Cho mảng các số nguyên dương A[] gồm n số. Hãy tìm tất cả các số nguyên dương K sao cho tất cả các phần tử của mảng A[] lấy phần dư với K đều bằng nhau. Ví dụ với mảng A[] = {6, 38, 34} ta tìm được các số K = {1, 2, 4} vì:
- $6\%1 = 38\%1 = 34\%1 = 0; 6\%2 = 38\%2 = 34\%2 = 0; 6\%4 = 38\%4 = 34\%4 = 2;$

- Input:
- Dòng đầu tiên đưa vào số lượng test T.
 - Những dòng kế tiếp đưa vào các bộ test. Mỗi test gồm hai dòng: dòng thứ nhất đưa vào số n; dòng tiếp theo đưa vào n số của mảng A[]. Các số được viết cách nhau một vài khoảng trắng.
 - T, A[i], n thỏa mãn ràng buộc: $1 \leq T \leq 100; 0 \leq n \leq 10^5; 1 \leq A[i] \leq 10^5$.

- Output:
- Đưa ra số các số K kết quả mỗi test theo từng dòng.

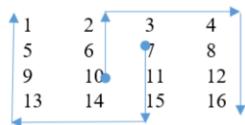
Input:	Output:
2	3
3	1
6 38 34	
2	
3 2	

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

228. IN MA TRẬN – 2

IN MA TRẬN - 2**Bài làm tốt nhất**

Cho số N biểu diễn cho ma trận vuông $A[4*N][4*N]$ được điền các con số từ 1 đến $4*N*4*N$ theo thứ tự từ nhỏ đến lớn, từ trái qua phải. Nhiệm vụ của bạn là in các phần tử của ma trận theo hai hình cuộn dây. Ví dụ với $N = 2$ ta có ma trận 4×4 và hai cuộn dây sau:



Cuộn 1: 10 6 2 3 4 8 12 16

Cuộn 2: 7 11 15 14 13 9 5 1

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một số N được viết trên 1 dòng.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 10$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	10 6 2 3 4 8 12 16
1	7 11 15 14 13 9 5 1

Giới hạn thời gian: 2s

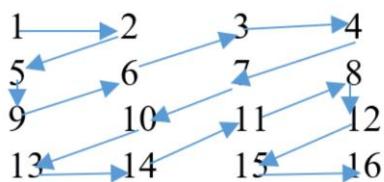
Giới hạn bộ nhớ: 65536 Kb

229. IN MA TRẬN – 3

IN MA TRẬN - 3**Bài làm tốt nhất**

Cho ma trận $A[N][M]$. Nhiệm vụ của bạn là in các phần tử của ma trận theo đường chéo. Ví dụ về in ma trận theo đường chéo: 1 2

5 9 6 3 4 7 10 13 14 11 8 12 15 16.

**Input:**

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N, M là cấp của ma trận $A[][],$ dòng tiếp theo đưa vào $N \times M$ số $A[i][j]$; các số được viết cách nhau một vài khoảng trắng.
- T, M, N, $A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq M, N \leq 100$; $1 \leq A[i][j] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2 4 7 5 3 6 8 9
3 3	1 2 3 4
1 2 3	
4 5 6	
7 8 9	
2 2	
1 2	
3 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

230. MA TRẬN VUÔNG LỚN NHẤT

Cho ma trận vuông $A[N][N]$ có các phần tử hoặc là ký tự 'O' hoặc là ký tự 'X'. Hãy tìm cấp của ma trận vuông lớn nhất có các phần tử 'X' bao quanh các phần tử 'O'. Ví dụ với ma trận dưới đây ta sẽ có kết quả là 3.

X	O	X	X	X
X	X	X	X	X
X	X	O	X	O
X	X	X	X	X
X	X	X	O	O

Input:

- Đòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: Đòng đầu tiên đưa vào số N ; dòng tiếp là $N \times N$ các phần tử của ma trận $A[][]$; các phần tử được viết cách nhau một vài khoảng trắng.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 20$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
2	3
XX	
XX	
4	
XXXO	
XOXX	
XXXO	
XOXX	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

231 SẮP ĐẶT HAI DÃY SỐ

Cho mảng $A1[]$ và $A2[]$ gồm n, m phần tử theo thứ tự. Hãy sắp xếp lại các phần tử trong $A1[]$ theo quan hệ thứ tự trong $A[2]$. Phần tử xuất hiện trước trong $A2[]$ và có mặt trước trong $A1[]$ đứng trước; các phần tử xuất hiện trong $A1[]$ nhưng không xuất hiện trong $A2[]$ đứng sau theo thứ tự tăng dần. Ví dụ với mảng $A1[] = \{2, 1, 2, 5, 7, 1, 9, 3, 6, 8, 8\}$, $A2[] = \{2, 1, 8, 3\}$ sau khi sắp xếp ta được $A1[] = \{2, 2, 1, 1, 8, 8, 3, 5, 6, 7, 9\}$.

Input:

- Đòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng thứ nhất đưa là hai số n, m; dòng thứ hai đưa vào n số của mảng $A1[i]$; dòng thứ ba đưa vào m số của mảng $A2[i]$; các số được viết cách nhau một vài khoảng trắng.
- T, n, m, $A1[i], A2[j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n, m \leq 10^6$; $1 \leq A1[i], A2[i] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	2 2 1 1 8 8 3 5 6 7 9
11 4	
2 1 2 5 7 1 9 3 6 8 8	
2 1 8 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

232 TẬP HỢP NGUYÊN TỐ CÙNG NHAU

Cho hai số tự nhiên n, m. Nhiệm vụ của bạn là xác định xem có thể chia các số từ 1 đến n thành hai tập sao cho giá trị tuyệt đối của tổng hai tập là m và tổng các phần tử của cả hai tập là các số đồng nguyên tố (co-prime : nguyên tố cùng nhau) hay không? Ví dụ n=5, m = 7 ta có kết quả là Yes vì ta chia thành 2 tập {1, 2, 3, 5} và 4 có giá trị tuyệt đối của tổng hai tập là 7 và là các số nguyên tố cùng nhau. Với n=6, m=3 ta có câu trả lời là No vì ta có thể tìm ra hai tập {1, 2, 4, 5} và {3, 6} có giá trị tuyệt đối của tổng là 3 tuy nhiên $12=1+2+4+5$ và $9=3+6$ không là đồng nguyên tố.

Input:

- Dòng đầu tiên đưa vào T là số lượng bộ test.
- T dòng tiếp theo các bộ test. Mỗi bộ test được viết trên một dòng là bộ hai số n, m phân biệt nhau bởi một dấu cách trống.
- Các số T, n, m, thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n, m \leq 10^{12}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
2	Yes
5 7	No
6 3	

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

233 TÍCH CHẬP

Phép tích chập (convolution) là kỹ thuật quan trọng trong xử lý ảnh. Kết quả phép tích chập giữa ma trận $x[]$ và ma trận kernel $h[]$ được xác định bằng công thức:

$$y[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k h[u, v] \cdot x[i - u, j - v]$$

Trong đó ma trận kernel có kích thước bằng $2k+1$. Với kernel 3×3 thì $-1 \leq u, v \leq 1$, do đó, giá trị các phần tử của ma trận kết quả có dạng:

$$\begin{aligned} y[i, j] &= h[-1, -1] \cdot x[i+1, j+1] + h[-1, 0] \cdot x[i+1, j] + h[-1, 1] \cdot x[i+1, j-1] \\ &\quad + h[0, -1] \cdot x[i, j+1] + h[0, 0] \cdot x[i, j] + h[0, 1] \cdot x[i, j-1] \\ &\quad + h[1, -1] \cdot x[i-1, j+1] + h[1, 0] \cdot x[i-1, j] + h[1, 1] \cdot x[i-1, j-1] \end{aligned}$$

Cho ma trận ảnh và ma trận kernel 3×3 . Nhiệm vụ của bạn là hãy thực hiện phép nhân tích chập của 2 ma trận, sau đó tính tổng tất cả các phần tử của ma trận thu được.

2D Convolution

$$\begin{array}{|c|c|c|c|} \hline 2 & 1 & 0 & 0 \\ \hline 3 & 2 & 1 & 1 \\ \hline 4 & 3 & 2 & 1 \\ \hline 2 & 2 & 1 & 0 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 0 & -2 \\ \hline 7 & 5 \\ \hline \end{array}$$

Giải thích test: Vị trí ô đầu tiên của ma trận kết quả:

$$\begin{aligned} &(-1) \times 2 + (-1) \times 1 + (-1) \times 0 + \\ &(-1) \times 3 + (8) \times 2 + (-1) \times 1 + \\ &(-1) \times 4 + (-1) \times 3 + (-1) \times 2 = 0 \end{aligned}$$

Input:

- Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).
- Mỗi test bắt đầu bởi hai số nguyên N và M. ($3 \leq N, M \leq 300$).
- Kế tiếp là N dòng, mỗi dòng gồm M số nguyên mô tả ma trận ảnh.
- 3 dòng tiếp theo, mỗi dòng gồm 3 số nguyên mô tả ma trận kernel.
- Giá trị các phần tử của hai ma trận có giá trị tuyệt đối không vượt quá 100.

Output:

Với mỗi test, hãy in ra **tổng các phần tử của ma trận mới tìm được**.

Ví dụ:

Input	Output
2	10
4 4	45
2 1 0 0	
3 2 1 1	
4 3 2 1	
2 2 1 0	
-1 -1 -1	
-1 8 -1	
-1 -1 -1	
3 3	
1 2 3	
4 5 6	
7 8 9	
1 1 1	
1 1 1	
1 1 1	

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

234 LÀM MỊN ẢNH

Phương pháp làm mịn ảnh được thực hiện bằng cách sử dụng phép tích chập (convolution) giữa ma trận ảnh và một ma trận kernel có dạng:

$$K = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Ma trận kernel trong ví dụ trên có kích thước bằng 5. Với ma trận kernel có kích thước $L = 2k + 1$, giá trị điểm ảnh (i,j) của ma trận mới sẽ bằng tổng của $(2k + 1) \times (2k + 1)$ phần tử $(i+u, j+v)$ với mọi $-k \leq u, v \leq k$, sau đó chia cho $(2k + 1) \times (2k + 1)$. Kết quả điểm ảnh mới thu được sau khi thực hiện phép chia sẽ được làm tròn xuống.

Cho ma trận ảnh đầu vào và kích thước L của ma trận kernel, nhiệm vụ của bạn là hãy in ra ma trận ảnh sau khi được làm mịn.

Input:

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

Mỗi test bắt đầu bởi hai số nguyên N, M và L ($3 \leq N, M \leq 500; L \leq \min(n, m)$). L được đảm bảo là một số nguyên lẻ.

Ké tiếp là N dòng, mỗi dòng gồm M số nguyên mô tả ma trận ảnh đầu vào, có giá trị trong phạm vi từ 0 tới 255.

Output:

Với mỗi test, hãy in ra ma trận ảnh sau khi đã được làm mịn.

Ví dụ:

Input	Output
2	2 1
4 4 3	3 1
2 1 0 0	5
3 2 1 1	
4 5 2 1	
2 2 9 0	
3 3 3	
1 2 3	
4 5 6	
7 8 9	

Giải thích test 1: Giá trị phần tử $(1,1) = \text{floor}[(2+1+0+3+2+1+4+5+2) / 9] = \text{floor}[20/9] = 2$.

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

235 TÍCH TOÁN TRÊN DÃY SỐ

TÍCH TOÁN TRÊN DÃY SỐ

Bài làm tốt nhất

Cho hai hàm $h(x)$ và $g(x)$ xác định trên tập các số tự nhiên $A[]$ gồm n phần tử. Trong đó, $h(x)$ là tích của các số trong mảng $A[]$, $g(x)$ làước số chung lớn nhất của các số trong mảng $A[]$. Nhiệm vụ của bạn là tìm giá trị $h(x)g(x)^{10^9}$. Chủ ý, khi lời giải cho kết quả lớn hãy đưa ra giá trị modulo với $10^{18} + 7$.

Input:

- Dòng đầu tiên đưa vào T là số lượng bộ test.
- T dòng tiếp theo các bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào số n là số các phần tử của mảng $A[]$; dòng tiếp theo đưa vào n số tự nhiên phân biệt nhau bởi một vài khoảng trắng.
- Các số $T, N, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 26; 1 \leq n \leq 60; 1 \leq A[i] \leq 10^4$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
1	64
2	
2 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

236 TÍCH HAI SỐ NGUYÊN LỚN

Cho hai số rất lớn X và Y được biểu diễn như hai xâu ký tự. Nhiệm vụ của bạn là tìm $X \times Y$?

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa xâu X; dòng tiếp theo đưa vào xâu Y.
- T, X, Y thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq \text{length}(X), \text{length}(Y) \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	6
2 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

237 XÂU HẾT CHO 8

XÂU HẾT CHO 8

Bài làm tốt nhất

Cho xâu ký tự S bao gồm các số từ 0 đến 9. Nhiệm vụ của bạn là đếm số các xâu con của S là số chia hết cho 8 và không chia hết cho 3.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự số S.
- T, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq \text{Length}(S) \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	5
888	15
6564525600	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

238 XÂU CON NHỎ NHẤT – 1

XÂU CON NHỎ NHẤT - 1

Bài làm tốt nhất

Cho hai xâu ký tự S1 và S2. Nhiệm vụ của bạn là hãy tìm xâu con nhỏ nhất của S1 chứa đầy đủ các ký tự của S2. Nếu không tồn tại xâu con thỏa mãn yêu cầu bài toán, hãy đưa ra -1.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là bộ đôi S1 và xâu ký tự S2 được viết trên các dòng khác nhau.
- T, S1, S2 thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{Length}(S1), \text{Length}(S2) \leq 100$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	toprac
timetopractice	
toc	apzo
zoomlazapzo	
oza	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

239 XÂU CON NHỎ NHẤT – 2

XÂU CON NHỎ NHẤT - 2

Bài làm tốt nhất

Cho xâu ký tự S. Nhiệm vụ của bạn là hãy tìm độ dài xâu con nhỏ nhất của S chứa đầy đủ các ký tự của S mỗi ký tự ít nhất một lần.
Ví dụ với xâu S="aabcbcdcba" ta có xâu con nhỏ nhất là "dbca".

Input:

- Đông đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ký tự S được viết trên một dòng.
- T, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{Length}(S) \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
aabcbcdcba	2
aaab	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

240 BIẾN ĐỔI A – B

BIẾN ĐỔI A – B

Bài làm tốt nhất

Cho xâu ký tự s chỉ bao gồm hai chữ cái là 'A' và 'B'.

Mỗi bước được phép biến đổi một vị trí bất kỳ trong xâu (A thành B, B thành A) hoặc cũng có thể biến đổi một dãy liên tiếp các ký tự nào đó tính từ đầu xâu.

Hãy tính xem cần ít nhất bao nhiêu bước để biến đổi xâu về dạng toàn chữ cái A.

Input:

Chi có 1 dòng ghi xâu ký tự s, độ dài không quá 1 triệu ký tự.

Output:

Ghi ra kết quả bài toán

Ví dụ:

Input:	Output:
AAABBBAAABBB	4

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

241 SẮP XẾP THEO SỐ LẦN XUẤT HIỆN

SẮP XẾP THEO SỐ LẦN XUẤT HIỆN

Bài làm tốt nhất

Cho mảng A[] gồm n số nguyên. Nhiệm vụ của bạn là sắp xếp mảng theo số lần xuất hiện các phần tử của mảng. Số xuất hiện nhiều lần nhất đứng trước. Nếu hai phần tử có số lần xuất hiện như nhau, số nhỏ hơn đứng trước. Ví dụ A[] = {5, 5, 4, 6, 4}, ta nhận được kết quả là A[] = {4, 4, 5, 5, 6}.

Input:

- Đông đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n, tương ứng với số phần tử của mảng A[] và số k; dòng tiếp theo là n số A[i]; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^4$; $1 \leq k \leq 10^2$; $1 \leq A[i] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4 4 5 5 6
5	9 9 9 2 5
5 5 4 6 4	
5	
9 9 9 2 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

242 SỐ NHỎ NHẤT LỚN HƠN A[i]

SỐ NHỎ NHẤT LỚN HƠN A[i]

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử. Nhiệm vụ của bạn là tìm giá trị nhỏ nhất lớn hơn A[i] ($i=0, 1, 2, \dots, n-1$). Đưa ra ' ' nếu A[i] không có phần tử nhỏ hơn nó. Ví dụ với mảng A[] = {13, 6, 7, 12}, ta có kết quả là {_, 7, 12, 13}.

Input:

- Đông đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^4$; $10^{-6} \leq A[i] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7 6 10 9 15 3 2 _ 8
9	_ 7 12 13
6 3 9 8 10 2 1 15 7	
4	
13 6 7 12	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

243 PHẦN TỬ CHUNG CỦA BA DÃY SỐ

Cho mảng $A[i], B[j], C[k]$ gồm N_1, N_2, N_3 phần tử đã được sắp xếp. Hãy đưa ra các phần tử có mặt trong cả ba mảng theo thứ tự tăng dần. Nếu không có đáp án, in ra -1.

Input:

- Đông đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm bốn dòng: dòng thứ nhất đưa vào N_1 , N_2 , N_3 là số phần tử của mảng $A[i], B[j], C[k]$; các dòng tiếp theo đưa vào N_1 số $A[i]$, N_2 số $B[j]$, N_3 số $C[k]$.
- $T, N_1, N_2, N_3, A[i], B[j], C[k]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N_1, N_2, N_3 \leq 10^6$; $0 \leq A[i], B[j], C[k] \leq 10^{10}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	
6 5 8	
1 5 10 20 40 80	
6 7 20 80 100	
3 4 15 20 30 70 80 120	

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

244 BỐN ĐIỂM TRÊN MẶT PHẲNG

Cho 4 điểm trong không gian 3 chiều. Nhiệm vụ của bạn là kiểm tra xem chúng có cùng nằm trên một mặt phẳng hay không?

Nếu có in ra "YES", in ra "NO" trong trường hợp ngược lại.

Input:

Đông đầu tiên là số lượng bộ test T ($T \leq 10000$).

Mỗi test gồm 4 dòng, lần lượt là tọa độ nguyên $x[i], y[i], z[i]$ của các điểm.

($-1000 \leq x[i], y[i], z[i] \leq 1000$).

Output:

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

Input:	Output
3	YES
1 2 0	YES
2 3 0	NO
4 0 0	
0 0 0	
1 1 1	
2 2 2	
3 3 3	
4 4 4	
5 6 7	
-8 -9 -10	
12 19 0	
3 1 5	

Giới hạn thời gian: 2s
Giới hạn bộ nhớ: 65536 Kb

245 QUẢN LÝ BÁN HÀNG – 1

Khai báo lớp Khách hàng với các thuộc tính:

- Mã khách hàng: tự động tăng, tính từ KH001
- Tên khách hàng: xâu ký tự độ dài không quá 50
- Giới tính: Nam hoặc Nữ
- Ngày sinh: Theo đúng chuẩn dd/mm/yyyy
- Địa chỉ: xâu ký tự độ dài không quá 100

Khai báo lớp Mặt hàng với các thuộc tính:

- Mã mặt hàng: tự động tăng, tính từ MH001
- Tên mặt hàng: xâu ký tự độ dài không quá 100
- Đơn vị tính: xâu ký tự độ dài không quá 10
- Giá mua: số nguyên dương không quá 7 chữ số
- Giá bán: số nguyên dương không quá 7 chữ số

Khai báo lớp Hóa đơn là bạn của lớp Khách hàng và lớp Mặt hàng trong đó có các thông tin:

- Mã hóa đơn
- Mã khách hàng
- Mã mặt hàng
- Số lượng (không quá 1000)

Viết chương trình nhập danh sách hóa đơn và in danh sách ra màn hình.

Input

Dòng đầu ghi số N là số khách hàng (không quá 20).

Tiếp theo là thông tin của N khách hàng, mỗi khách hàng ghi trên 4 dòng theo đúng thứ tự đã mô tả (không có mã)

Dòng tiếp theo ghi số M là số mặt hàng (không quá 40).

Tiếp theo là thông tin của M mặt hàng, mỗi mặt hàng ghi trên 4 dòng theo đúng thứ tự đã mô tả (không có mã)

Dòng tiếp theo ghi số K là số hóa đơn (không quá 100).

Mỗi hóa đơn ghi trên **1 dòng** gồm 3 thông tin theo đúng thứ tự đã mô tả (không có mã).

Output

Ghi ra danh sách hóa đơn theo đúng thứ tự nhập, trong đó gồm các thông tin sau, mỗi thông tin cách nhau đúng một khoảng trắng.

- Mã hóa đơn
- Tên khách hàng
- Địa chỉ
- Tên mặt hàng
- Đơn vị tính
- Giá mua
- Giá bán
- Số lượng
- Thành tiền

Ví dụ

Input
2 Nguyen Van Nam Nam 12/12/1997 Mo Lao-Ha Dong-Ha Noi Tran Van Binh Nam 11/14/1995 Phung Khoang-Nam Tu Liem-Ha Noi 2 Ao phong tre em Cai 25000 41000 Ao khoac nam Cai 240000 515000 3 KH001 MH001 2 KH001 MH002 3 KH002 MH002 4

Output
HD001 Nguyen Van Nam Mo Lao-Ha Dong-Ha Noi Ao phong tre em Cai 25000 41000 2 82000 HD002 Nguyen Van Nam Mo Lao-Ha Dong-Ha Noi Ao khoac nam Cai 240000 515000 3 1545000 HD003 Tran Van Binh Phung Khoang-Nam Tu Liem-Ha Noi Ao khoac nam Cai 240000 515000 4 2060000

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

246 QUẢN LÝ BÁN HÀNG – 2

Khai báo lớp Khách hàng với các thuộc tính:

- Mã khách hàng: tự động tăng, tính từ KH001
- Tên khách hàng: xâu ký tự độ dài không quá 50
- Giới tính: Nam hoặc Nữ
- Ngày sinh: Theo đúng chuẩn dd/mm/yyyy
- Địa chỉ: xâu ký tự độ dài không quá 100

Khai báo lớp Mật hàng với các thuộc tính:

- Mã mặt hàng: tự động tăng, tính từ MH001
- Tên mặt hàng: xâu ký tự độ dài không quá 100
- Đơn vị tính: xâu ký tự độ dài không quá 10
- Giá mua: số nguyên dương không quá 7 chữ số
- Giá bán: số nguyên dương không quá 7 chữ số

Khai báo lớp Hóa đơn là bạn của lớp Khách hàng và lớp Mật hàng trong đó có các thông tin:

- Mã hóa đơn
- Mã khách hàng
- Mã mặt hàng
- Số lượng (không quá 1000)
- Lợi nhuận

Viết chương trình nhập danh sách hóa đơn, sắp xếp theo lợi nhuận giảm dần và in danh sách ra màn hình.

Input

Dòng đầu ghi số N là số khách hàng (không quá 20).

Tiếp theo là thông tin của N khách hàng, mỗi khách hàng ghi trên 4 dòng theo đúng thứ tự đã mô tả (không có mã)

Dòng tiếp theo ghi số M là số mặt hàng (không quá 40).

Tiếp theo là thông tin của M mặt hàng, mỗi mặt hàng ghi trên 4 dòng theo đúng thứ tự đã mô tả (không có mã)

Dòng tiếp theo ghi số K là số hóa đơn (không quá 100).

Mỗi hóa đơn ghi trên 1 dòng gồm 3 thông tin theo đúng thứ tự đã mô tả (không có mã và lợi nhuận).

Output

Ghi ra danh sách hóa đơn đã sắp xếp, trong đó gồm các thông tin sau, mỗi thông tin cách nhau đúng một khoảng trắng.

- Mã hóa đơn
- Tên khách hàng
- Địa chỉ
- Tên mặt hàng
- Số lượng
- Thành tiền
- Lợi nhuận

247 ĐẾM SỐ CÁCH DI CHUYỂN

ĐẾM SỐ CÁCH DI CHUYỂN

Bài làm tốt nhất

Cho ma trận vuông $A[N][N]$ gồm các số nguyên dương và số tự nhiên K. Hãy tìm số các cách di chuyển từ phần tử đầu tiên ($A[0][0]$) đến phần tử cuối cùng $A[N-1][N-1]$ sao cho tổng các phần tử của phép di chuyển đúng bằng K. Biết từ phần tử $A[i][j]$, ta chỉ được phép dịch chuyển đến phần tử $A[i+1][j]$ hoặc $A[i][j+1]$. Ví dụ với ma trận dưới đây sẽ có 2 phép di chuyển theo nguyên tắc kể trên để có tổng bằng 12.

1	2	3
4	6	5
3	2	1

Cách 1: 1 → 2 → 6 → 2 → 1
Cách 2: 1 → 2 → 3 → 5 → 1

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: Dòng đầu tiên đưa vào hai số N, K ; dòng tiếp là $N \times N$ các phần tử của ma trận $A[][],$ các phần tử được viết cách nhau một vài khoảng trắng.
- T, N, $A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq N \leq 20; 1 \leq A[i][j] \leq 200.$

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	0
3 16	2
1 2 3	
4 6 5	
9 8 7	
3 12	
1 2 3	
4 6 5	
3 2 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

248 DÃY SỐ BITONIC

DÃY SỐ BITONIC

Bài làm tốt nhất

Cho mảng $A[]$ gồm n phần tử gồm các số nguyên dương. Mảng $A[]$ được gọi là Bitonic nếu các phần tử của mảng được chia thành hai phần: phần thứ nhất tăng dần, phần thứ hai giảm dần. Mảng $A[]$ được sắp xếp tăng dần cũng là mảng Bitonic khi xem phần thứ hai là rỗng. Tương tự như vậy, mảng $A[]$ được sắp xếp giảm dần cũng là một bitonic. Hãy tìm độ dài dãy con dài nhất của mảng $A[]$ là một Bitonic. Ví dụ với mảng $A[] = \{1, 11, 2, 10, 4, 5, 2, 1\}$ ta có kết quả là 6 tương ứng với độ dài dãy con $\{1, 2, 10, 4, 2, 1\}.$

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào số lượng phần tử của mảng N; phần thứ hai đưa vào n số $A[i]$; các số được viết cách nhau một vài khoảng trắng.
- T, n, $A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq n \leq 100; 1 \leq A[i] \leq 200.$

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	5
5	6
1 2 5 3 2	
8	
1 11 2 10 4 5 2 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

249 TỔNG LỚN NHẤT CỦA DÃY CON TĂNG DẦN

TỔNG LỚN NHẤT CỦA DÃY CON TĂNG DẦN

Bài làm tốt nhất

Cho mảng A[] gồm n số nguyên dương. Hãy tìm tổng lớn nhất của dãy con tăng dần của dãy số A[]. Ví dụ với A[] = { 1, 101, 2, 3, 100, 4, 5} ta có câu trả lời là $106=1+2+3+100$. Với dãy A[] = {10, 5, 4, 3} ta có câu trả lời là 10.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n; dòng tiếp theo là n số A[i] của mảng A[]; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A[i] \leq 10^7$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	106
7	10
1 101 2 3 100 4 5	
4	
10 5 4 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

250 ƯỚC SỐ CHUNG LỚN NHẤT CỦA SỐ NGUYÊN LỚN

Bài làm tốt nhất

ƯỚC SỐ CHUNG LỚN NHẤT CỦA SỐ NGUYÊN LỚN

Cho hai số a và b trong đó $a \leq 10^{12}$, $b \leq 10^{250}$. Nhiệm vụ của bạn là tìm ước số chung lớn nhất của hai số a, b.

Input:

- Dòng đầu tiên đưa vào T là số lượng bộ test.
- T dòng tiếp đưa các bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào số a; dòng tiếp theo đưa vào số b.
- Các số T, a, b thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq a \leq 10^{12}$; $1 \leq b \leq 10^{250}$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input	Output
1	3
1221	
1234567891011121314151617181920212223242526272829	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

251 QUẢN LÝ BÁN HÀNG – 3

Khai báo lớp Khách hàng với các thuộc tính:

- Mã khách hàng: tự động tăng, tính từ KH001
- Tên khách hàng: xâu ký tự độ dài không quá 50
- Giới tính: Nam hoặc Nữ
- Ngày sinh: Theo đúng chuẩn dd/mm/yyyy
- Địa chỉ: xâu ký tự độ dài không quá 100

Khai báo lớp Mặt hàng với các thuộc tính:

- Mã mặt hàng: tự động tăng, tính từ MH001
- Tên mặt hàng: xâu ký tự độ dài không quá 100
- Đơn vị tính: xâu ký tự độ dài không quá 10
- Giá mua: số nguyên dương không quá 7 chữ số
- Giá bán: số nguyên dương không quá 7 chữ số

Khai báo lớp Hóa đơn trong đó có các thông tin:

- Mã hóa đơn
- Khách hàng
- Mặt hàng
- Số lượng (không quá 1000)

Viết chương trình nhập danh sách hóa đơn và in danh sách ra màn hình.

Input - có 3 file văn bản

File KH.in

Dòng đầu ghi số N là số khách hàng (không quá 20).

Tiếp theo là thông tin của N khách hàng, mỗi khách hàng ghi trên 4 dòng theo đúng thứ tự đã mô tả (không có mã)

File MH.in

Dòng đầu ghi số M là số mặt hàng (không quá 40).

Tiếp theo là thông tin của M mặt hàng, mỗi mặt hàng ghi trên 4 dòng theo đúng thứ tự đã mô tả (không có mã)

File HD.in

Dòng đầu theo ghi số K là số hóa đơn (không quá 100)

Mỗi hóa đơn ghi trên **1 dòng** gồm 3 thông tin theo đúng thứ tự đã mô tả (không có mã).

Output

Ghi ra danh sách hóa đơn theo đúng thứ tự nhập, trong đó gồm các thông tin sau, mỗi thông tin cách nhau đúng một khoảng trắng.

- Mã hóa đơn
- Tên khách hàng
- Địa chỉ
- Tên mặt hàng
- Đơn vị tính
- Giá mua
- Giá bán
- Số lượng
- Thành tiền

Ví dụ

Input
File KH.in
2
Nguyen Van Nam
Nam
12/12/1997
Mo Lao-Ha Dong-Ha Noi
Tran Van Binh
Nam
11/14/1995
Phung Khoang-Nam Tu Liem-Ha Noi
File MH.in
2
Ao phong tre em
Cai
25000
41000
Ao khoac nam
Cai
240000
515000
File HD.in
3
KH001 MH001 2
KH001 MH002 3
KH002 MH002 4
Output
HD001 Nguyen Van Nam Mo Lao-Ha Dong-Ha Noi Ao phong tre em Cai 25000 41000 2 82000
HD002 Nguyen Van Nam Mo Lao-Ha Dong-Ha Noi Ao khoac nam Cai 240000 515000 3 1545000
HD003 Tran Van Binh Phung Khoang-Nam Tu Liem-Ha Noi Ao khoac nam Cai 240000 515000 4 2060000

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 150000 Kb

252 PHÂN SỐ NHỎ NHẤT

Cho phân số P/Q . Phân số này có thể được tối giản bằng cách xóa đi một số chữ số ở cả tử số và mẫu số (nếu xóa 1 chữ số X ở P thì cũng xóa đúng 1 chữ số X ở Q).

Bạn cần tìm cách xóa chữ số để thu được phân số nhỏ nhất A/B sao cho A/B đúng bằng với P/Q (A tìm được cần nhỏ nhất có thể).

Lưu ý: sau khi xóa, có thể tồn tại chữ số 0 ở đầu của số A (hoặc B). Khi này giá trị của A (hoặc B) vẫn được chấp nhận (xem test 5).

Input:

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

Mỗi test gồm 2 số nguyên P, Q. Input đảm bảo đáp án có tồn tại.

Output:

Với mỗi test, in ra phân số nhỏ nhất tìm được.

Giới hạn:

Subtask 1 (50%): $0 < P, Q \leq 10^9$

Subtask 2 (50%): $0 < P, Q \leq 2^{63}$

Test ví dụ:

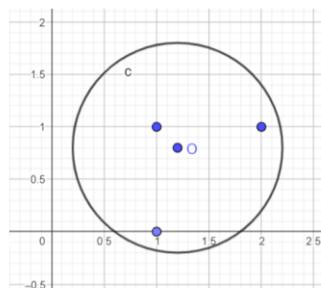
Input:	Output
5	1 2
163 326	2 1
326 163	1 1
1000 1000	232 16936
2232 162936	3522 5283
71635224 107452836	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 524288 Kb

253 PTIT 2024D - ĐƯỜNG TRÒN

Cho đường tròn có tâm tại vị trí (X, Y) và bán kính R . Hãy đếm số lượng các điểm có tọa độ nguyên nằm bên trong hoặc trên đường tròn?



Input:

Gồm 3 số thực X, Y, R có tối đa 4 chữ số sau dấu phẩy ($|X|, |Y| \leq 10^5$, $0 \leq R \leq 10^5$).

Output:

In ra số lượng điểm có tọa độ nguyên nằm bên trong hoặc bên trên đường tròn đã cho.

Test ví dụ:

Input	Output
1.2 0.8 1	3
0 0 1	5
-39066.9606 -83906 45886.5164	6614851027

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

254 XẾP HÀNG TỐI ƯU

Để trình diễn một tiết mục trong màn khai mạc Đại hội thể thao quốc tế, đạo diễn Mr. Tim đã mời N vận động viên tham gia. Theo kịch bản, N vận động viên sẽ được xếp thành một hàng dọc hoặc một hàng ngang liên tiếp (song song với trục tọa độ, các điểm cách nhau 1 đơn vị). Hiện tại, vận động viên thứ i đang ở vị trí ($X[i]$, $Y[i]$), nếu vận động viên này di chuyển đến vị trí ($U[i]$, $V[i]$) thì sẽ mất năng lượng là $|X[i] - U[i]| + |Y[i] - V[i]|$.

Yêu cầu: Hãy giúp đạo diễn xác định cách xếp hàng để tổng năng lượng di chuyển của cả N vận động viên là nhỏ nhất.

Input:

Dòng đầu tiên là số nguyên dương N.

N dòng tiếp theo, mỗi dòng gồm 2 số nguyên $X[i]$, $Y[i]$ có giá trị tuyệt đối không vượt quá 10^9 .

Output:

In ra một số nguyên là tổng năng lượng di chuyển của cả N vận động viên.

Giới hạn:

- Subtask 1 (40%): $0 \leq X[i], Y[i] \leq 100$, $N \leq 100$.
- Subtask 2 (40%): $0 \leq X[i], Y[i] \leq 10000$, $N \leq 10000$.
- Subtask 3 (20%): $N \leq 100000$.

Test ví dụ:

Input	Output
3	2
1 1	
1 2	
3 3	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

255 PHÂN TÍCH SỐ 123

Cho số N, bạn cần phân tích N thành K số A[1], A[2], ..., A[K] thỏa mãn:

- $A[1] + A[2] + \dots + A[K] = N$
- A[i] chỉ chứa các chữ số 1, 2, 3.

Có thể có nhiều cách phân tích. Nhiệm vụ của bạn là hãy tìm giá trị K nhỏ nhất thỏa mãn yêu cầu.

Input:

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

Mỗi test gồm một số nguyên N.

Output:

Với mỗi test, in ra giá trị K tìm được trên một dòng.

Giới hạn:

Subtask 1 (40%): $N \leq 10^5$;

Subtask 2 (60%): $N \leq 10^{18}$.

Test ví dụ:

Input	Output
5	1
123	2
456	4
90	1
1	4
1000	

Giải thích:

$$123 = 123$$

$$456 = 133 + 323$$

$$90 = 22 + 22 + 23 + 23$$

$$1 = 1$$

$$1000 = 132 + 232 + 313 + 323$$

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

256 PHÂN TÍCH SỐ 123 - BẢN ĐỀ

Cho số N, bạn cần phân tích N thành K số A[1], A[2], ..., A[K] thỏa mãn:

- $A[1] + A[2] + \dots + A[K] = N$
- $A[i]$ chỉ chứa các chữ số 1, 2, 3.

Có thể có nhiều cách phân tích. Nhiệm vụ của bạn là hãy tìm giá trị K nhỏ nhất thỏa mãn yêu cầu.

Input:

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

Mỗi test gồm một số nguyên N.

Output:

Với mỗi test, in ra giá trị K tìm được trên một dòng.

Giới hạn:

$N \leq 100000$;

Test ví dụ:

Input	Output
5	1
123	2
456	4
90	1
1	4
1000	

Giải thích:

$$123 = 123$$

$$456 = 133 + 323$$

$$90 = 22 + 22 + 23 + 23$$

$$1 = 1$$

$$1000 = 132 + 232 + 313 + 323$$

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

257 DI CHUYỂN KANGAROO

Cho một bảng kích thước $N \times M$. Trên bảng có đúng một cái hố, và những ô còn lại, mỗi ô chứa một chú kangaroo. Trò chơi được mô tả bởi một chuỗi thao tác được cho trước, bao gồm 4 loại:

D: di chuyển tất cả kangaroo đi xuống một ô.

U: di chuyển tất cả kangaroo di lên một ô.

L: di chuyển tất cả kangaroo di sang trái một ô.

R: di chuyển tất cả kangaroo di sang phải một ô.

Sau mỗi thao tác, nếu có chú kangaroo nào rơi xuống hố hoặc di ra ngoài bảng thì sẽ bị loại bỏ. Cho trước ba số nguyên N, M, K và chuỗi thao tác dưới dạng xâu S , nhiệm vụ của bạn là hãy tính số vị trí đặt hố sao cho sau khi chuỗi thao tác được thực hiện, số kangaroo còn lại trên bảng đúng bằng K .

Input:

Dòng đầu tiên là số lượng bộ test T ($1 \leq T \leq 1000$).

Mỗi test bắt đầu bởi 3 số nguyên N, M và K ($1 \leq N, M \leq 1000, 0 \leq K < N \times M$).

Dòng tiếp theo chứa một xâu S là các thao tác được thực hiện ($1 \leq |S| \leq 10^6$).

Input đảm bảo tổng các giá trị $N \times M$ và $|S|$ của tất cả các test không vượt quá 10^6 .

Lưu ý: 25% số test có $1 \leq N, M \leq 100$.

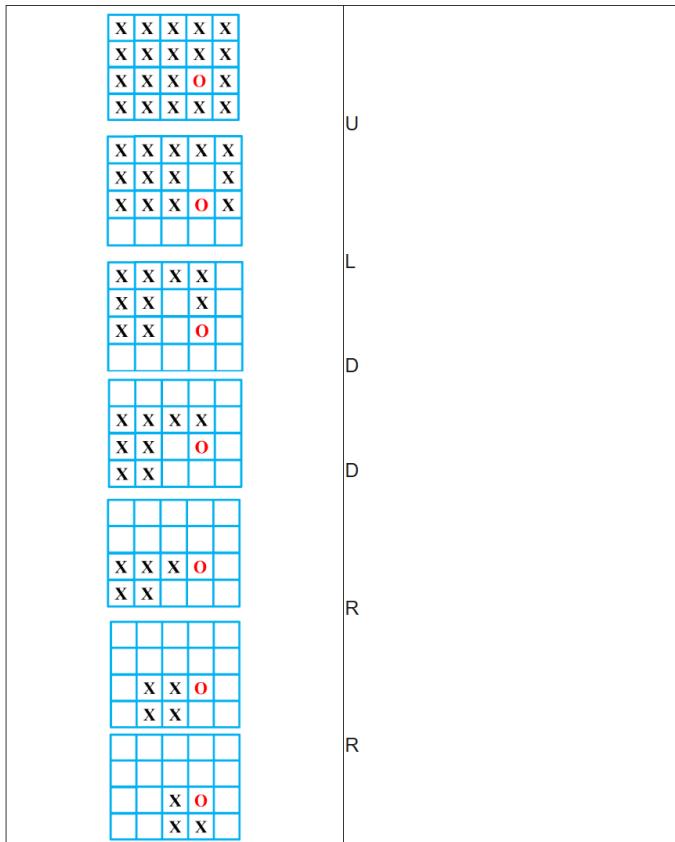
Output:

Với mỗi test, hãy in ra đáp án tìm được trên một dòng

Test ví dụ:

Input	Output
4	2
4 5 3	0
ULDDRR	6
4 5 10	20
UUUUUUU	
4 5 2	
UUDRULU	
4 5 0	
UUUUDLUL	

Giải thích test 1: Hai vị trí thỏa mãn là (3, 4) và (4, 3). Dưới đây là hình vẽ minh họa khi đặt cái hố ở vị trí (3, 4):



258 PTIT25R1A - DÃY SỐ TRUNG BÌNH CỘNG

PTIT25R1A - DÃY SỐ TRUNG BÌNH CỘNG

Bài làm tốt nhất

Cho dãy số nguyên $A[]$ có N phần tử. Số $A[p]$ ($1 \leq p \leq N$) được gọi là một số trung bình cộng trong dãy nếu tồn tại 3 chỉ số i, j, k đôi một khác nhau sao cho $A[p] = (A[i] + A[j] + A[k])/3$.

Nhiệm vụ của bạn là hãy đếm số lượng các số trung bình cộng có mặt trong dãy số.

Input:

Dòng đầu tiên là số nguyên dương N ($3 \leq N \leq 1000$).

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

Output:

In ra số lượng số trung bình cộng tìm được.

Test ví dụ:

Input:	Output
5 3 4 6 3 5	2
3 1 2 5	0

Giải thích test 1: $A[2] = (A[1] + A[3] + A[4])/3, A[5] = (A[2] + A[3] + A[5])/3$.

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

259 PTIT25R1H - PHÂN HOẠCH THÀNH CÁC XÂU ĐỐI XỨNG

Cho một xâu S, nhiệm vụ của bạn là hãy phân hoạch xâu S thành N xâu con liên tiếp:

$S = S_1 + S_2 + \dots + S_N$ thỏa mãn:

- Với mỗi xâu con S_i , ta có thể hoán vị các kí tự để thu được một xâu đối xứng.
- Giá trị N là nhỏ nhất có thể.

Input:

Chứa một xâu S có không quá 100000 kí tự thường.

Output:

In ra một số nguyên là số N nhỏ nhất tìm được.

Test ví dụ:

Input:	Output:
aabcddee	2
abcdefghijkl	10
abcabc	2

Giải thích test 1: Tách aabcddee thành aab và cddee. Ta có aba và deced là xâu đối xứng khi hoán vị lại các kí tự.

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

260 PTIT25R1 - LŨY THỪA

Cho số nguyên dương n . Gọi $f(n)$ là số lượng ước của n . Ví dụ $f(8) = 4$ vì 8 có 4 ước là 1, 2, 4 và 8.

Cho số nguyên dương X , nhiệm vụ của bạn là hãy tìm giá trị n nhỏ nhất sao cho $n^{f(n)} = X$?

Input:

Dữ liệu đầu vào chứa một số nguyên X duy nhất ($1 \leq X \leq 10^{18}$).

Output:

In ra một số nguyên là giá trị n nhỏ nhất tìm được. Nếu không tồn tại đáp án, in ra -1.

Test ví dụ:

Input:	Output:
1234	-1
64	4
10000000000000000000	100

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

261 SAMSUNG25F - DI CHUYỂN TRONG MÊ CUNG, BẢN KHÓ

Cho một mê cung có kích thước $N \times M$. Tại ô (i, j) có thể một ô trống (kí tự $.$) hoặc một ô cấm (kí tự $#$). Một nhà ào thuật đang đứng ở vị trí (Cx, Cy) , và anh ấy cần di chuyển đến vị trí (Dx, Dy) . Anh ấy có thể thực hiện một trong hai phép di chuyển sau:

- Di chuyển sang một ô kề chung cạnh với vị trí hiện tại đang đứng, tính là 1 bước.
- Sử dụng phép thuật để dịch chuyển đến một ô vuông trong khu vực 5×5 có tâm là ô vuông anh ta đang đứng, được tính là 2 bước.

Nhiệm vụ của bạn là hãy xác định xem nhà ào thuật cần sử dụng ít nhất bao nhiêu bước di chuyển để đi tới vị trí đích?

Input:

Dòng đầu tiên chứa hai số nguyên N và M ($1 \leq N, M \leq 1000$).

Dòng tiếp theo là 4 số nguyên Cx, Cy, Dx, Dy . Input đảm bảo 2 vị trí (Cx, Cy) và (Dx, Dy) là 2 ô trống.

N dòng tiếp theo, mỗi dòng là một xâu gồm M kí tự, mô tả một hàng của mê cung.

Output:

In ra một số nguyên là số bước di chuyển ít nhất tìm được. Nếu như nhà ào thuật không thể di chuyển được tới đích, hãy in ra -1 .

Test ví dụ:

Input:	Output:
4 4 1 1 4 4 . # . . # . . # .. . # ..	4
4 4 1 4 4 1 . # . . # # . . # # . . # # # . # # .	-1

Giải thích test 1: Sử dụng phép thuật di chuyển ô $(1, 1) \rightarrow (3, 3)$, rồi di 2 bước thường tới ô $(4, 4)$.

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

262 BẮT CÁ

Có N con cá, khối lượng từng con là W_i , ban đầu ở vị trí X_i , di chuyển với vận tốc V_i theo chiều dương của trục tọa độ Ox.

Tí có một chiếc lưới đánh cá có thể trải dài trong phạm vi chiều dài bằng A , nếu thả lưới ở vị trí L thì sẽ bắt được tất cả các con cá trong phạm vi $[L, L+A]$. Các bạn hãy xác định giúp Tí xem nên chọn thời điểm nào để quăng lưới và quăng lưới ở vị trí nào để bắt được khối lượng cá là nhiều nhất có thể?

Input:

Dòng đầu tiên là số nguyên dương N và A ($1 \leq A \leq 10000$).

N dòng tiếp theo, mỗi dòng gồm 3 số nguyên W_i, X_i, V_i ($1 \leq W_i, V_i \leq 10000, 0 \leq X_i \leq 10000$).

Output:

In ra một số nguyên là khối lượng cá lớn nhất có thể thu được.

Giới hạn:

Subtask 1 (20%): $1 \leq N \leq 20$.

Subtask 2 (20%): $1 \leq N \leq 200$.

Subtask 3 (60%): $1 \leq N \leq 2000$.

Ví dụ:

Input	Output
3 10 1000 10 30 1000 20 10 1000 0 100	3000
3 10 1000 100 100 1000 10 40 1000 20 20	2000

Giải thích test 1: Tại thời điểm $t = 0.25$, vị trí của 3 con cá lần lượt là 17.5, 22.5 và 25. Do vậy, Tí có thể bắt trọn 3 con cá nếu như đặt lưới ở bất kì vị trí nào trong phạm vi $[15, 17.5]$, chẳng hạn như $L = 15$ hoặc $L = 17.5$

Giải thích test 1: Bắt cá tại thời điểm $t = 0$ và thả lưới ở vị trí $L = 10$.

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb

263 SỐ HOÀN HẢO

SỐ HOÀN HẢO

Bài làm tốt nhất

Cho số tự nhiên N. Nhiệm vụ của bạn là hãy kiểm tra N có phải là số hoàn hảo hay không. Một số N được gọi là số hoàn hảo nếu tổng các ước số của nó bằng chính nó. Ví dụ $N = 6=1 + 2 + 3$ là số hoàn hảo.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên N.
- T, N thỏa mãn ràng buộc $1 \leq T \leq 100$; $1 \leq N \leq 10^{18}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
6	0
21	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

264 SỐ LỚN NHẤT CỦA DÃY CON LIÊN TỤC

SỐ LỚN NHẤT CỦA DÃY CON LIÊN TỤC

Bài làm tốt nhất

Cho mảng A[] gồm n số nguyên không âm và số k. Hãy tìm số lớn nhất của mỗi dãy con liên tục gồm k phần tử của mảng. Ví dụ với mảng A[] = {1, 2, 3, 1, 4, 5, 2, 3, 6}, K = 3, ta có kết quả 3 3 4 5 5 5 6.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[] và số k; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trắng.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq k < n \leq 10^7$; $0 \leq A[i] \leq 10^7$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

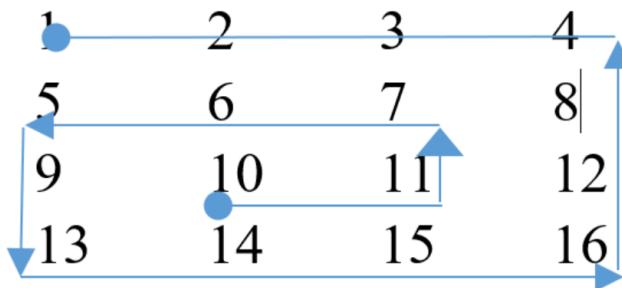
Input:	Output:
2	3 3 4 5 5 5 6
9 3	10 10 10 15 15 90 90
1 2 3 1 4 5 2 3 6	
10 4	
8 5 10 7 9 4 15 12 90 13	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

265 MA TRẬN XOĂN ỐC – 2

Cho ma trận $A[N][M]$. Nhiệm vụ của bạn là in các phần tử của ma trận theo hình xoắn ốc ngược. Ví dụ về in ma trận theo hình xoắn ốc ngược như dưới đây: 10 11 7 6 5 9 13 14 15 16 12 8 4 3 2 1.



Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N, M là cấp của ma trận $A[][],$ dòng tiếp theo đưa vào $N \times M$ số $A[i][j]$; các số được viết cách nhau một vài khoảng trắng.
- T, M, N, $A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100;$ $1 \leq M, N \leq 100;$ $1 \leq A[i][j] \leq 10^5.$

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

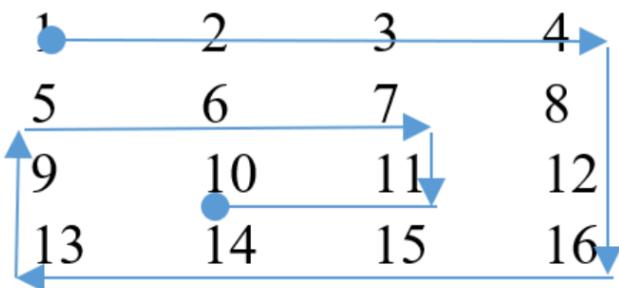
Input:	Output:
2	10 11 7 6 5 9 13 14 15 16 12 8 4 3 2 1
4 4	11 10 9 8 7 13 14 15 16 17 18 12 6 5 4 3 2 1
1 2 3 4	
5 6 7 8	
9 10 11 12	
13 14 15 16	
3 6	
1 2 3 4 5 6	
7 8 9 10 11 12	
13 14 15 16 17 18	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

266 MA TRẬN XOẮN ỐC – 3

Cho ma trận $A[N][M]$. Nhiệm vụ của bạn là đưa ra phần tử thứ k phép duyệt theo mô hình xoắn ốc trên ma trận của ma trận theo hình xoắn ốc. Ví dụ với $k=6$ của ma trận dưới đây sẽ cho ta kết quả là 12 (Phép duyệt xoắn ốc: 1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10).



Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N, M là cấp của ma trận $A[][],$ và số $k;$ dòng tiếp theo đưa vào $N \times M$ số $A[i][j];$ các số được viết cách nhau một vài khoảng trắng.
- $T, M, N, k, A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq M, N, k \leq 100; 1 \leq A[i][j] \leq 10^5.$

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	12
4 4 6	5
1 2 3 4	
5 6 7 8	
9 10 11 12	
13 14 15 16	
3 4 10	
1 2 3 4	
5 6 7 8	
9 10 11 12	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

267 THAY THẾ X – O

THAY THẾ X - O

Bài làm tốt nhất

Cho ma trận $A[N][M]$ có các phần tử hoặc là ký tự ‘‘O’’ hoặc là ký tự ‘‘X’’. Hãy thay thế các miền bao quanh ‘‘O’’ bằng ‘‘X’’. Một miền các ký tự ‘‘O’’ bị bao quang bởi ký tự ‘‘X’’ nếu các ký tự ‘‘X’’ xuất hiện ở phía dưới, phía trên, bên trái, bên phải các ký tự ‘‘O’’. Ví dụ với ma trận dưới đây ta sẽ có kết quả như sau:

X	X	X	X	X	X	X	X
X	O	X	X	X	X	X	X
X	O	O	X	X	X	X	X
X	O	X	X	X	X	X	X
X	X	O	O	X	X	O	O

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: Dòng đầu tiên đưa vào hai số N, M ; dòng tiếp là $N \times M$ các phần tử của ma trận $A[][]$; các phần tử được viết cách nhau một vài khoảng trắng.
- T, N, M thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N, M \leq 20$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 1 5 X O X O X 3 3 X X X X O X X X X	X O X O X X X X X X X X X X

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

268 TÍNH HẠNG CỦA MA TRẬN

TÍNH HẠNG CỦA MA TRẬN

Bài làm tốt nhất

Cho ma trận $A[N][M]$. Hãy tìm hạng của ma trận $A[N][M]$. Hạng của ma trận (Rank Matrix) là số các cột hoặc các hàng độc lập tuyến tính. Ví dụ hạng của ma trận dưới đây là 2 vì có hàng 1 và hàng 2 là phụ thuộc tuyến tính.

```
10    20    10  
20    40    20  
30    50    0
```

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: Dòng đầu tiên đưa vào hai số N, M ; dòng tiếp là $N \times M$ các phần tử của ma trận $A[][],$ các phần tử được viết cách nhau một vài khoảng trắng.
- T, N, M, $A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq N, M \leq 15; -10^2 \leq A[i][j] \leq 10^2.$

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
3 3	2
10 20 10 20 40 20 30 50 0	
3 3	
10 20 10 -20 -30 10 30 50 0	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

269 GIÁ TRỊ LỚN NHẤT CỦA MA TRẬN

GIÁ TRỊ LỚN NHẤT CỦA MA TRẬN

Bài làm tốt nhất

Giả sử giá trị của một ma trận là hiệu giữa tổng các số trên đường chéo chính và tổng các số trên đường chéo phụ. Cho ma trận A kích thước $N * N$, hãy tìm ma trận con của A sao cho ma trận con đó có giá trị lớn nhất.

Input

Dòng đầu ghi số N ($2 \leq N \leq 400$)

N dòng tiếp theo ghi ma trận A. Các số trong đoạn $[-1000, 1000].$

Output

Ghi ra giá trị lớn nhất tìm được.

Ví dụ

Input	Output
4 9 -2 -8 0 -6 -2 0 -9 4 -5 6 1 1 3 4 9	26

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

270 THƯƠNG HAI SỐ NGUYÊN LỚN

THƯƠNG HAI SỐ NGUYÊN LỚN

Bài làm tốt nhất

Cho hai số rất lớn X và Y được biểu diễn như hai xâu ký tự. Nhiệm vụ của bạn là tìm X / Y?

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa xâu X; dòng tiếp theo đưa vào xâu Y.
- T, X, Y thỏa mãn ràng buộc: $1 \leq T \leq 100$; $0 \leq \text{length}(X), \text{length}(Y) \leq 10^3$.

Output:

- Đưa ra số kết quả mỗi test theo từng dòng.

Input:	Output:
1	2
8 4	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

271 SỐ ĐỔI XỨNG LỚN NHẤT

SỐ ĐỔI XỨNG LỚN NHẤT

Bài làm tốt nhất

Cho số nguyên dương gồm N chữ số. Bạn chỉ được phép thực hiện hai thao tác:

Thao tác A: loại bỏ tất cả các chữ số giống nhau.

Thao tác B: sắp đặt lại vị trí các chữ số.

Hãy tìm số nguyên đổi xứng lớn nhất có thể được tạo ra bằng cách thực hiện hai thao tác A hoặc B ở trên. Ví dụ với số $N = 1122233300000998$, ta có thể tạo ra số đổi xứng lớn nhất là 910000019 bằng cách thực hiện các thao tác A, B như sau:

Thao tác A: loại bỏ các chữ số 2 ta nhận được số 1133300000998.

Thao tác A: loại bỏ các chữ số 3 ta nhận được số 1100000998.

Thao tác A: loại bỏ các chữ số 8 ta nhận được số 110000099.

Thao tác B: sắp đặt lại các số còn lại 110000099 để được số 910000019.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên dương có N chữ số.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 1000$.

Output:

- Đưa ra số nguyên lớn nhất được tạo ra bởi thao tác A, B của mỗi test theo từng dòng.

Input:	Output:
5	5
12345000	910000019
11233300000998	333
3335	63336
333566	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

272 SUM STRING

Cho xâu ký tự S bao gồm các chữ số. Xâu S được gọi là Sum String nếu tồn tại một số tự nhiên $k > 2$ sao cho ta có thể chia xâu S thành k xâu con khác nhau $S = (S_1, S_2, \dots, S_k)$ sao cho các số được tạo bởi các xâu con thỏa mãn điều kiện $S_i = S_{i-1} + S_{i-2}$ ($i = 3, 4, \dots, k$). Ví dụ xâu $S = "123415538"$ là một Sum String vì tồn tại số $k = 3$ để phân tích xâu S thành 3 xâu con $S = ("123", "415", "538")$ thỏa mãn $123 + 415 + 538 = 538$. Tương tự như vậy xâu $S = "12345"$ không phải là một Sum String.

Hãy kiểm tra xem S có phải là xâu Sum String hay không?

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các test. Mỗi test là một xâu ký tự số S.
- T và S thỏa mãn ràng buộc $1 \leq T \leq 100$, $3 \leq \text{length}(S) \leq 105$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
3	Yes
123415538	No
12345	Yes
1122335588143	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

273 ĐẾM CÁC SỐ CHIA HẾT CHO N

Cho xâu ký tự S chỉ bao gồm các ký tự số. Nhiệm vụ của bạn là đếm các xâu con của S chia hết cho N. Ví dụ $S = "1234"$, $N = 4$ ta nhận được kết quả là 4 tương ứng với 4 xâu con chia hết cho 4: 4, 12, 24, 124. Xâu con có thể chứa các ký tự không liên tiếp nhau.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là hai số M, N, trong đó M là độ dài xâu S; phần tiếp theo là xâu S gồm M số.
- T, M, N, S thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq M, N \leq 100$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4
4 4	3
1234	
3 6	
676	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

274 MẬT KHẨU

(Giới hạn thời gian chạy: 10 giây)

Hệ thống quản lý đào tạo của PTIT đang gặp một vấn đề về bảo mật. Do sự cố này, các account bị đổi thành tên viết liền của các sinh viên. Và chỉ cần đánh một chuỗi kí tự có chứa mật khẩu là có thể đăng nhập vào hệ thống. Chẳng hạn sinh viên A có mật khẩu là “abcd”, nếu ai đó đăng nhập với tài khoản là tên của A, mật khẩu “abcdef” hay “aaaabcd” đều được chấp nhận.

Nhân cơ hội này, rất nhiều bạn sinh viên đã cố gắng hack vào tài khoản của những người khác. Cho biết danh sách mật khẩu của tất cả các user, bài toán đặt ra là hãy xác định xem có nhiêu nhất bao nhiêu trường hợp user này có thể login vào user khác?

Input:

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

N dòng tiếp theo, mỗi dòng chứa mật khẩu của một user, có độ dài không quá 10 kí tự và chỉ gồm các kí tự thường.

Output:

In ra một số nguyên là đáp án đáp án tìm được.

Ví dụ:

Test 1	Test 2
Input :	Input :
3	3
aaa	x
aa	x
abb	xy
Output :	Output :
1	4

Giải thích test 2: User 1 có thể login vào user 2 và ngược lại. User 3 có thể login vào tài khoản của user 1 và 2.

Giới hạn thời gian: 10s

Giới hạn bộ nhớ: 65536 Kb

275 DIỆN TÍCH ĐA GIÁC

Cho một đa giác lồi có N đỉnh trên mặt phẳng Oxy.

Nhiệm vụ của bạn là hãy tính diện tích đa giác này.

Input:

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

Mỗi test bắt đầu bởi số nguyên N ($N \leq 1000$).

N dòng tiếp theo, mỗi dòng gồm 2 số nguyên $x[i], y[i]$ ($-1000 \leq x[i], y[i] \leq 1000$) là tọa độ của điểm thứ i. Các điểm được liệt kê theo thứ tự ngược chiều quay kim đồng hồ.

Output:

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

Input:	Output
2	0.500
3	4.000
0 0	
1 0	
0 1	
4	
0 0	
2 0	
2 2	
0 2	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

276 HÌNH CHỮ NHẬT CON CÓ TỔNG LỚN NHẤT

HÌNH CHỮ NHẬT CON CÓ TỔNG LỚN NHẤT

Bài làm tốt nhất

Cho ma trận $A[N][M]$. Nhiệm vụ của bạn là tìm hình chữ nhật con của $A[::]$ có tổng các phần tử lớn nhất. Ví dụ với ma trận dưới đây ta sẽ tìm ra được ma trận con có tổng các từ lớn nhất là 29.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào N, M; dòng tiếp theo đưa vào $N \times M$ số $A[i][j]$; các số được viết cách nhau một vài khoảng trắng.
- $T, M, N, A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq M, N \leq 200$; $-10^5 \leq A[i][j] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 4 5 1 2 -1 -4 -20 -8 -3 4 2 1 3 8 10 1 3 -4 -1 1 7 -6	29

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

277 SỐ BƯỚC DI CHUYỂN ÍT NHẤT

Cho mảng $A[]$ gồm n số nguyên. Giá trị mỗi phần tử biểu diễn số lượng lớn nhất các bước ta có thể dịch chuyển từ phần tử này. Phần tử có giá trị bằng 0 được hiểu là không được phép dịch chuyển. Xuất phát từ phần tử đầu tiên, hãy đếm số lượng ít nhất các bước dịch chuyển đến phần tử cuối cùng trong mảng. Ví dụ với $A[] = \{1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9\}$ ta có câu trả lời là 3 tương ứng với các phép dịch chuyển: $1 \rightarrow 3 \rightarrow 8 \rightarrow 9$ ($A[1] = 1$ nên ta dịch chuyển nhiều nhất 1 bước đến $A[2]$; $A[2] = 3$ nên ta được phép dịch chuyển nhiều nhất 3 bước tương ứng với $A[3]=5, A[4]=8, A[5] = 9$; do $A[4] = 8$ nên ta chỉ cần dịch chuyển 1 bước nữa là đến đích).

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n ; dòng tiếp theo là n số $A[i]$ của mảng $A[]$; các số được viết cách nhau một vài khoảng trắng.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq n \leq 10^3; 1 \leq A[i] \leq 10^7$.

Output:

- Đưa ra số bước dịch chuyển ít nhất của mỗi test, đưa ra -1 nếu ta không có phép dịch chuyển đến đích.

Input:	Output:
2 11 1 3 5 8 9 2 6 7 6 8 9 6 1 4 3 2 6 7	3 2

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

278 TÌM ĐƯỜNG TRONG MA TRẬN NHỊ PHÂN

Cho ma trận $A[N][M]$ chỉ bao gồm các số 0 và 1. Hãy tìm đường đi ngắn nhất từ một phần tử bắt đầu đến phần tử đích. Biết mỗi bước đi ta chỉ được phép dịch chuyển từ phần tử có giá trị 1 đến phần tử có giá trị 1. Ví dụ với ma trận dưới đây sẽ cho ta kết quả là 11.

```

A[9][10] ={{1, 0, 1, 1, 1, 1, 0, 1, 1, 1},
{1, 0, 1, 0, 1, 1, 1, 0, 1, 1},
{1, 1, 1, 0, 1, 1, 0, 1, 0, 1},
{0, 0, 0, 0, 1, 0, 0, 0, 0, 1},
{1, 1, 1, 0, 1, 1, 1, 0, 1, 0},
{1, 0, 1, 1, 1, 1, 0, 1, 0, 0},
{1, 0, 0, 0, 0, 0, 0, 0, 0, 1},
{1, 0, 1, 1, 1, 1, 0, 1, 1, 1},
{1, 1, 0, 0, 0, 1, 0, 0, 1, 1}};
Begin = A[0][0];
End = A[3][4];

```

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: Dòng đầu tiên đưa vào 6 số N, M, phần tử bắt đầu (x, y), phần tử kết thúc (z, t); dòng tiếp là $N \times M$ các phần tử của ma trận $A[][],$ các phần tử được viết cách nhau một vài khoảng trắng.
- T, N, M, x, y, z, t thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq N, M \leq 10^3; 1 \leq x, y, z, t \leq 10^3.$

Output:

- Đưa ra kết quả mỗi test theo từng dòng. Nếu không tìm được đáp án, in ra -1.

Input:	Output:
1 9 10 0 0 3 4 1 0 1 1 1 1 0 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 0 1 0 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1	11

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

279 HÌNH CHỮ NHẬT LỚN NHẤT – 2

HÌNH CHỮ NHẬT LỚN NHẤT - 2

Bài làm tốt nhất

Cho ma trận $A[N][M]$ chỉ bao gồm các số 0 và 1. Hãy tìm hình chữ nhật lớn nhất có các phần tử đều bằng 1. Ví dụ với ma trận dưới đây ta sẽ có hình chữ nhật lớn nhất có các phần tử là 1 bằng 8.

0	1	1	0					
1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	
1	1	0	0					

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: Dòng đầu tiên đưa vào hai số N, M ; dòng tiếp là $N \times M$ các phần tử của ma trận $A[][]$; các phần tử được viết cách nhau một vài khoảng trắng.
- T, N, M thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N, M \leq 500$.

Output:

Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 4 4 0 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0	8

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

280 MA TRẬN CON LỚN NHẤT

MA TRẬN CON LỚN NHẤT

Bài làm tốt nhất

Cho ma trận $A[N][M]$ chỉ bao gồm các số 0 và 1. Hãy tìm cấp ma trận vuông con lớn nhất có các phần tử đều bằng 1. Ví dụ với ma trận dưới đây ta sẽ có cấp ma trận vuông con lớn nhất có các phần tử là 1 bằng 3.

0	1	1	0	1
1	1	0	1	0
0	1	1	1	0
1	1	1	1	0
1	1	1	1	1
0	0	0	0	0

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: Dòng đầu tiên đưa vào hai số N, M ; dòng tiếp là $N \times M$ các phần tử của ma trận $A[][]$; các phần tử được viết cách nhau một vài khoảng trắng.
- T, N, M thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq N, M \leq 500$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3
5 6	2
0 1 0 1 0 1	
1 0 1 0 1 0	
0 1 1 1 1 0	
0 0 1 1 1 0	
1 1 1 1 1 1	
2 2	
1 1	
1 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

281 DÃY CON LIÊN TỤC NHỎ NHẤT

DÃY CON LIÊN TỤC NHỎ NHẤT

Bài làm tốt nhất

Cho mảng A[] gồm n số nguyên và số X. Hãy tìm độ dài dãy con liên tục nhỏ nhất có tổng lớn hơn X. Ví dụ với $A[] = \{1, 4, 45, 6, 0, 19\}$ và $X = 51$ ta có câu trả lời là 3 tương ứng với dãy con $\{4, 45, 6\}$. Với $A[] = \{1, 10, 5, 2, 7\}$ và $X = 9$ ta có câu trả lời là 1 tương ứng với dãy con $\{10\}$. Với dãy $A[] = \{1, 2, 4\}$ và $X = 8$ ta có câu trả lời là -1.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n và số X; dòng tiếp theo là n số $A[i]$ của mảng A[]; các số được viết cách nhau một vài khoảng trống.
- $T, n, X, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $1 \leq A[i] \leq 10^7$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2 6 51 1 4 45 6 0 19 3 8 1 2 4	3 -1

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

282 ĐỔI CHỖ ÍT NHẤT

ĐỔI CHỖ ÍT NHẤT

Bài làm tốt nhất

Cho mảng A[] gồm n phần tử. Hãy tìm số phép đổi chỗ ít nhất giữa các phần tử của mảng để mảng A[] được sắp xếp. Ví dụ với $A[] = \{4, 3, 2, 1\}$ ta cần thực hiện ít nhất 2 phép đổi chỗ: Swap(A[0], A[3]), Swap(A[1], A[2]).

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên là số phần tử của mảng n và X; dòng tiếp theo là n số $A[i]$ của mảng A[]; các số được viết cách nhau một vài khoảng trống.
- T, n thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

283 BẢNG MÀU R - G - B

Ta cần tạo một xâu ký tự S có độ dài n. Trong đó, mỗi ký tự trong S chỉ là các ký tự R, B, hoặc G. Xâu ký tự nhân được có ít nhất r ký tự R, b ký tự B, g ký tự G ($r + b + g \leq n$). Hãy đếm số các xâu ký tự thỏa mãn yêu cầu kể trên. Ví dụ với $n=4$, $r=1$, $b=1$, $g = 1$ ta có thể có 36 xâu ký tự khác nhau.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là bộ bốn số phân biệt n, r, b, g được viết trên một dòng.
- T, S, n, r, b, g thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 20$; $1 \leq r, b, g \leq N$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	36
4 1 1 1	22
4 2 0 1	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

284 SO SÁNH LŨY THỪA

SO SÁNH LŨY THỪA

Bài làm tốt nhất

Cho mảng X[] gồm n phần tử và mảng Y[] gồm m phần tử. Hãy đếm số các cặp $x^y > y^x$, trong đó $x \in X[]$ và $y \in Y[]$. Ví dụ $X[] = \{2, 1, 6\}$, $Y[] = \{1, 5\}$ ta có kết quả là 3 cặp $(2, 1), (2, 5), (6, 1)$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào n, m tương ứng với số phần tử của mảng X[] và Y[]; dòng tiếp theo là n số $X[i]$ của mảng X[]; dòng cuối cùng là m số của mảng Y[]; các số được viết cách nhau một vài khoảng trắng.
- T, n, m, $X[i], Y[j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n, m \leq 10^5$; $1 \leq X[i], Y[j] \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	3
3 2	
2 1 6	
1 5	

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

285 BỘI SỐ CHUNG NHỎ NHẤT CỦA TẬP CON

Cho một tập hợp gồm n số nguyên dương. Bạn cần tính tổng của bội chung nhỏ nhất của tất cả các tập hợp con của tập hợp ban đầu (trừ tập hợp rỗng).

Vì đáp số có thể rất lớn nên hãy in ra kết quả theo modulo 10007.

Input

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

Mỗi bộ test bao gồm số nguyên dương n ($1 \leq n \leq 100$), là số lượng phần tử của dãy số.

Dòng tiếp theo gồm n số a_i ($1 \leq a_i \leq 500$).

Output

Với mỗi test, in ra số thứ tự của test (Case ...) và đáp số của bài toán.

Example**Input:**

2

3

3 4 5

2

2 7

Output:

Case 1: 119

Case 2: 23

Giới hạn thời gian: 10s

Giới hạn bộ nhớ: 65536 Kb

286 PTIT25R1E - TỐI ƯU HỆ THỐNG GIAO THÔNG

Hệ thống giao thông của đất nước Highland có N thành phố được nối với nhau bởi $N-1$ con đường hai chiều. Chiều dài của mỗi tuyến đường bằng 1 đơn vị độ dài. Mạng lưới giao thông luôn đảm bảo rằng khi bạn xuất phát từ một thành phố bất kỳ, bạn đều có thể đi tới các thành phố khác.

Kinh tế năm qua phát triển mạnh, vì vậy chính phủ đã quyết định đầu tư thêm cho hệ thống giao thông của đất nước. Để giảm chi phí di lại cho người dân cũng như thúc đẩy các hoạt động kinh tế, chính phủ quyết định mở thêm một tuyến đường giữa hai thành phố nào đó, sao cho tổng chi phí vận hành của hệ thống giao thông là nhỏ nhất. Tổng chi phí của hệ thống giao thông được tính bằng tất cả các tổng đường đi ngắn nhất giữa các cặp thành phố (có tất cả $N*(N-1)/2$ cặp thành phố).

Nhiệm vụ của các bạn là hãy giúp ban tư vấn chính phủ xác định tổng chi phí nhỏ nhất cho hệ thống giao thông mới của đất nước?

Input:

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

Với mỗi test, dòng thứ nhất chứa số nguyên N là số thành phố của đất nước ($N \leq 300$).

$N-1$ dòng tiếp theo, mỗi dòng chứa hai số u, v biểu diễn con đường nối giữa thành phố u và v .

Output:

Với mỗi test, hãy in ra tổng chi phí nhỏ nhất của hệ thống giao thông.

Test ví dụ:

Input:	Output:
2	3
3	8
1 2	
2 3	
4	
1 2	
2 3	
3 4	

Giải thích test 1: con đường mới nối thành phố 1 với thành phố 3.

Khi đó tổng chi phí vận hành của hệ thống giao thông sẽ giảm từ 4 xuống còn 3.

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 524288 Kb