

BÀI TẬP C++

1. **LCM & GCD.** Cho hai số nguyên a, b . Nhiệm vụ của bạn là tìm $\text{LCM}(a, b)$ và $\text{GCD}(a, b)$.

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 vào khoảng trống.
- 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	

2. **LCM of Array.** Cho hai số tự nhiên n . Nhiệm vụ của bạn là tìm số nguyên nhỏ nhất chia hết cho $1, 2, \dots, n$.

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	

3. **GCD of Array.** 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 $A[]$, $g(x)$ là ước số chung lớn nhất của các số trong mảng $A[]$. Nhiệm vụ của bạn là tìm giá trị $h(x)^{g(x)}$. Chú ý, khi lời giải cho kết quả lớn hãy đưa ra giá trị modulo với 10^9+7 .

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 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 trống.
- 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	

4. **GCD of Lager Number.** 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	

5. **GCD(a, x, y).** 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 trống.
- 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	

6. **GCD(n, m).** 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 trị tuyệt đối của tổng là 3 tuy nhiên cặp 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 đưa các bộ test. Mỗi bộ test được viết trên một dòng là bộ hai số n, m phân biệt nhau bởi một vài khoảng 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	

7. **Smallest Number.** 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	

8. **Re-arrang Array.** Cho mảng A[] gồm n số nguyên dương. Hãy sắp đặt lại các phần tử trong mảng sao cho khi ghép các phần tử ta nhận được một số lớn nhất. Ví dụ với A[] = {54, 546, 548, 60} ta nhận được số lớn nhất là 6054854654. Đối với mảng A[] = {1, 34, 3, 98, 9, 76, 45, 4} ta nhận được số lớn nhất là 998764543431.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên ghi lại số các phần tử của mảng A[]. Dòng tiếp theo đưa vào n số phân biệt các phần tử của A[].
- Các số T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 10^2$; $1 \leq A[i] \leq 10^3$.

Output:

- Đưa ra theo từng dòng kết quả mỗi test .

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

9. **Largest Smaller N.** Cho số tự nhiên N. Bạn chỉ được phép sử dụng nhiều nhất một phép đổi chỗ để nhận được số lớn nhất nhỏ hơn N. Ví dụ với số N=12435, sử dụng một phép đổi chỗ ta nhận được số lớn nhất nhỏ hơn N là 12354. 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	12435
12435	-1
12345	

10. Số đối xứng lớn 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.

B: sắp đặt lại vị trí các chữ số.

Hãy tìm số

nguyên đối 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 đối 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:
2 12345000 11233300000998	5 910000019

11. **Prime Facstors.** 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:
2 315 31	3 3 5 7 31

12. **Max Prime Facstors.** 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:
2 315	7 31

13. **Prime Eratosthenes.** Cho số nguyên dương N . Hãy đưa ra tất cả các 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	

14. **Least Prime Factor.** 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 . 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	

15. **Prime in Range.** 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	

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

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.

Input:	Output:
2	2 71
74	3 1021
1024	

17. **Sphenic Number.** 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 Sphenic 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	

18. 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 trống.
- 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	

19. 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	

20. **Smith Number.** 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	No
666	

21. **Perfect Number.** 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	

22. **Kth Prime Divisors.** 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 $N = 255$, $k = 2$ ta có kết quả là 3 vì $255 = 3 \times 3 \times 5 \times 5$. 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 $1 \leq T \leq 100$; $1 \leq N, k \leq 10^4$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3
25 5	-1
81 5	

23. **Số các số có ước số lẻ.** Cho số tự nhiên N. Nhiệm vụ của bạn là hãy đếm số các số có ước số lẻ nhỏ hơn hoặc bằng N.

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
1	2
5	

24. **Number 3-Divisors.** 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	

25. **Smallest Divisors.** 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 $1 \leq T \leq 100$; $1 \leq N \leq 10^6$.

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	

26. **Number 3-Divisors.** Cho số tự nhiên N. Nhiệm vụ của bạn là hãy đếm tất cả các số có đúng ba ước số. Ví dụ n=100, ta có các số 4.

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	

27. **Number 3-Divisors.** 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	4
1 10	78498
1 1000000000000	

28. **Number 3-Divisors.** 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	

29. 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 trống.
- 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	

30. 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:

- Dòng 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 trống.
- 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 phương trình đồng dư không có nghiệm, hãy đưa ra -1

Input:	Output:
2	4
3 11	12
10 17	

31. 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:

- 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 trống.
- 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	10
1 11	

32. 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 cặp 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 trống.
- 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	

33. 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 12345 10	5

34. Cho số nguyên dương a, b, M , trong đó a là số rất lớn được biểu diễn như một xâu ký tự số. Hãy tìm $K = (a^b) \% M$. Ví dụ $a = 3, b = 2, M = 4$ thì $K = (3^2) \% 4 = 1$

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 a, b, M được viết trên một dòng.
- T, a, b, M thỏa mãn ràng buộc : $1 \leq T \leq 100; 0 \leq \text{length}(a) \leq 1000; 2 \leq b, M \leq 10^{12}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1 3 2 4	1

35. 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 trống.
- $T, A[i], n$ 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 Kết quả mỗi test theo từng dòng.

Input:	Output:
2 3 6 38 34 2 3 2	3 1

36. Tìm số nghiệm của phương trình đồng dư $x^2 = 1 \pmod{p}$ trong khoảng $[1, b]$. Ví dụ với $b = 5, p = 7$ ta tìm được $x = 1 \in [1, 5]$ để $x^2 = 1 \% 7 = 1$. Với $b = 8, p = 6$ ta tìm được $x = \{1, 5, 7\}$ để $x^2 = 1 \pmod{6}$.

$$6 \% 1 = 38 \% 1 = 34 \% 1 = 0; 6 \% 2 = 38 \% 2 = 34 \% 2 = 0; 6 \% 4 = 38 \% 4 = 34 \% 4 = 2;$$

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 trống.
- T, b, p thỏa mãn ràng buộc : $1 \leq T \leq 100; 0 \leq b \leq 10^9; 1 \leq p \leq 10^5$.

Output:

- Đưa ra số các số Kết quả mỗi test theo từng dòng.

Input:	Output:
2 5 7 8 6	1 3

--	--

37. Cho hai số n và r , hãy tìm $C(n, r) \% P$. Trong đó, $P = 10^9 + 7$.

Input:

- Dòng 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 trống.
- 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	

38. Tính toán giá trị đa thức $P(n, x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_0$. Cho hai số n và r , hãy tìm $C(n, r) \% P$. Trong đó, $P = 10^9 + 7$.

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 : $1 \leq T \leq 100$; $0 \leq n \leq 2000$; $0 \leq x, P[i] \leq 1000$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

39. Cho hai số rất lớn X và Y được biểu diễn như hai chuỗi 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 chuỗi X ; dòng tiếp theo đưa vào chuỗi 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:
2	11999
978	09999900
12977	
100	
1000000	

40. Cho hai số rất lớn X và Y được biểu diễn như hai chuỗi 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 chuỗi X ; dòng tiếp theo đưa vào chuỗi 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	198123
12	
198111	

41. Cho hai số rất lớn X và Y được biểu diễn như hai chuỗi 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 test gồm hai dòng: dòng thứ nhất đưa chuỗi X; dòng tiếp theo đưa vào chuỗi 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	6
2 3	

42. Cho hai số rất lớn X và Y được biểu diễn như hai chuỗi 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 chuỗi X; dòng tiếp theo đưa vào chuỗi 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	

43. Cho ba số a, b, c có thể lớn cỡ 10^{16} . Nhiệm vụ của bạn là tính $(a*b)\%c$.

Input:

- Dòng 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	

44. Dãy số Fibonacci được định nghĩa $F_n = F_{n-1} + F_{n-2}$, $n > 1$ 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	

45. 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: $1 \leq T \leq 100; 1 \leq n \leq 10^{18}$.

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	

46. Cho mảng $A[]$ gồm n số nguyên không âm. Hãy tìm dãy con lớn nhất chỉ toàn các số Fibonacci.

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 trống.
- $T, n, A[i]$ 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	

47. **Array Rotation** 1. Cho mảng $A[]$ gồm n phần tử và số d . Hãy thực hiện phép quay vòng d phần 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 trống.
- $T, n, d, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq d < n \leq 10^6; 1 \leq A[i] \leq 10^7$;

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	

48. **Array Rotation 2.** Cho mảng $A[]$ là mảng đã được quay trái k lần từ mảng $A[]$ ban đầu gồm các phần tử khác nhau được sắp xếp theo thứ tự tăng dần. Hãy tìm số $k=?$. Ví dụ $A[] = \{15, 18, 2, 3, 6, 12\}$ ta tìm được $k = 2$ vì mảng ban đầu là $A[] = \{2, 3, 6, 12, 15, 18\}$ sau khi quay trái $k=2$ ta nhận được mảng $A[] = \{15, 18, 2, 3, 6, 12\}$.

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: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $1 \leq A[i] \leq 10^{18}$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
6	1
15 18 2 3 6 12	
5	
5 1 2 3 4	

49. **Array Rotation 3 .** Cho mảng $A[]$ gồm n phần tử. Hãy tìm tổng lớn nhất của $i \cdot A[i]$ trong số các mảng quay trái của $A[]$. Ví dụ mảng $A[] = \{8, 3, 1, 2\}$, ta có tổng $i \cdot A[i]$ của mảng quay trái như sau:

$A[] = \{8, 3, 1, 2\}$: tổng $S = 0 \cdot 8 + 1 \cdot 3 + 2 \cdot 1 + 3 \cdot 2 = 11$.

$A[] = \{3, 1, 2, 8\}$: tổng $S = 0 \cdot 3 + 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 8 = 29$.

$A[] = \{1, 2, 8, 3\}$: tổng $S = 0 \cdot 1 + 1 \cdot 2 + 2 \cdot 8 + 3 \cdot 3 = 27$.

$A[] = \{2, 8, 3, 1\}$: tổng $S = 0 \cdot 2 + 1 \cdot 8 + 2 \cdot 3 + 3 \cdot 1 = 17$.

Từ đó ta có $\max = 27$.

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: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $1 \leq A[i] \leq 10^{18}$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	27
4	7
8 3 1 2	
3	
3 2 1	

50. **Array Rotation 4.** Cho mảng $A[]$ là mảng đã được sắp xếp và quay trái vòng. Hãy tìm phần tử nhỏ nhất của mảng với thời gian $\log(n)$.

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: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $1 \leq A[i] \leq 10^{18}$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

51. **Re-arrang Array 1.** 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, -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 trống.
- 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 2 3 4 -1 6 -1 -1 9
10	0 1 -1 3 -1 -1
-1 -1 6 1 9 3 2 -1 4 -1	
6	
0 -3 1 -2 3 -4	

52. **Re-arrang Array 2.** 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 trống.
- 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 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	

53. **Re-arrang Array 3.** 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} 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 trống.
- 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	

54. **Re-arrang Array 4.** 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 trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A[i] \leq 10^3$;

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	

55. **Re-arrang Array 5.** Cho mảng A[] gồm n số nguyên dương. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng sao cho 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 trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A[i] \leq 10^3$;

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	

56. **Re-arrang Array 6.** Cho mảng A[] gồm n số nguyên không âm. Ta gọi phần tử A[i] là hợp lệ nếu A[i] ≠ 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 trống.
- $T, n, A[i]$ 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
2 2 0 4 0 8	
10	
0 2 2 2 0 6 6 0 0 8	

57. Re-arrang Array 7. 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 trống.
- $T, n, A[i]$ 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	

58. Re-arrang Array 8. 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 trống.
- $T, n, A[i]$ 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	

59. **Re-arrang Array 9.** 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 trống.
- $T, n, A[i]$ 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	

60. **Re-arrang Array 10.** 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 $A[]$. Chú ý, các số không được phép lặp lại. Dưới đây là một số ví dụ mẫu:

- $A[] = "I"$: số tăng nhỏ nhất là 12.
- $A[] = "D"$: số giảm nhỏ nhất là 21
- $A[] = "DD"$: số giảm nhỏ nhất là 321
- $A[] = "DDIDDIID"$: 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
- $T, \text{Length}(A)$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{Length}(A) \leq 9$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
4	12
I	21
D	321
DD	321654798
DDIDDIID	

61. **Re-arrang Array 11.** 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 trống.
- 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	

62. **Re-arrang Array 12.** 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 dãy con liên tục trong mảng A[]. Ví dụ với mảng A[] = {6, -3, -10, 0, 2} ta có kết quả là 180 tương ứng với tích các phần tử của dãy con {6, -3, -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 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 trống.
- 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	

63. **Re-arrang Array 13.** 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 A2[]. 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:

- 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 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 trống.
- T, n, m, A1[i], A2[j] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n, m \leq 10^6$; $1 \leq A1[i], A2[i] \leq 10^6$.

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	

64. **Re-arrang Array 14.** 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; dòng thứ hai đưa vào n×k số của mảng A[n][k] các số được viết cách nhau một vài khoảng trống.
- T, n, k, A[i][j] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq k \leq 10$; $1 \leq A[i][j] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

65. **Oder Statistics 1.** Cho mảng A[] gồm n số và số k. Hãy tìm phần 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 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 trống.
- 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	9
7 10 4 3 20 15	
6 4	
9 7 12 8 6 5	

66. **Oder Statistics 2.** Cho mảng A[] gồm n phần tử. Hãy tìm phần 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 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: $1 \leq T \leq 100$; $1 \leq n \leq 10^5$; $1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

67. **Oder Statistics 3.** Cho ma trận vuông $A[][]$ cấp n . Các phần tử của ma trận $A[][]$ đã được sắp xếp theo hàng, cột. Hãy tìm phần 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	20
4 7	
10 20 30 40	
15 25 35 45	
24 29 37 48	
32 33 39 50	

68. **Oder Statistics 4.** 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 kế 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 trống.
- $T, n, A[i]$ 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	

69. **Oder Statistics 5.** Cho mảng $A[]$ gồm n số nguyên. Hãy tìm hiệu lớn nhất của bất kể hai phần tử nào của mảng dãy 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: $1 \leq T \leq 100$; $1 \leq n \leq 10^5$; $1 \leq A[i] \leq 10^5$;

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	

70. **Oder Statistics 6.** Cho mảng A[] gồm n số nguyên dương. Hãy tìm hiệu lớn nhất của i-j thỏa mãn ràng buộc $A[i] \leq A[j]$. 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 trống.
- 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	
34 8 10 3 2 80 30 33 1	

71. **Oder Statistics 7.** 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 trống.
- 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	

72. **Oder Statistics 8.** 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 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 trống.
- 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 5 1 2 7 4 3	7

73. **Oder Statistics 9.** 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 là 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 trống.
- 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:
2 7 1 2 3 4 5 2 1 5 1 2 7 4 3	5 7

74. **Oder Statistics 9.** 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 trống.
- 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 16 0 8 4 12 2 10 6 14 1 9 5 13 3 11 7 15 6 5 8 3 7 9 1	6 3

75. **Oder Statistics 10.** 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 trống.
- 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	

76. **Oder Statistics 11.** 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 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 trống.
- 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	

77. **Oder Statistics 12.** Cho hai mảng đã được sắp xếp A1[] và A2[] gồm n, m phần tử theo thứ tự. Nhiệm vụ của bạn là tìm tổng lớn nhất của các số bắt đầu tại phần tử đầu tiên của một mảng và kết thúc ở phần tử cuối cùng của một trong hai mảng. Ví dụ với mảng A1[] = {2, 3, 7, 10, 12}, A2[] = {1, 5, 7, 8} ta có kết quả là 35 tương ứng với tổng S = 1+5+7+10+12.

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: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A1[i], A2[j] \leq 10^4$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	35
5 4	
2 3 7 10 12	
1 5 7 8	

78. **Oder Statistics 11.** 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:

- 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: $1 \leq T \leq 100$; $1 \leq n \leq 10^6$; $10^{-6} \leq A[i] \leq 10^6$;

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	

79. **Range Query 1.** 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 $A[] = \{1, 1, 2, 1, 3, 4, 5, 2, 8\}$, các câu hỏi Q : $[0, 4]$, $[1, 3]$, $[2, 4]$ 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 trống.
- $T, n, Q, L, R, A[i]$ 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	7 6 10 9 15 3 2 _ 8
9 3	_ 7 12 13
1 1 2 1 3 4 5 2 8	
0 4 1 3 2 4	

80. **Range Query 2.** 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 trống.
- $T, n, L, R, A[i]$ 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	

81. **Range Query 3.** 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 trống.
- 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	

82. **Range Query 4.** 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 trống.
- 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	

83. **Range Query 5.** 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 ta 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 -> 3->8->9 (A[0]=1 nên ta dịch chuyển nhiều nhất 1 bước đến A[1]; A[1]=3 nên ta được phép dịch chuyển nhiều nhất 3 bước tương ứng với A[2]=5, A[3]=8, A[4]=9; do A[3]=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 trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $1 \leq A[i] \leq 10^7$.

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	3
11	2
1 3 5 8 9 2 6 7 6 8 9	
6	
1 4 3 2 6 7	

84. **Range Query 6.** 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 trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $1 \leq A[i] \leq 10^7$.

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	

85. **Range Query 7.** 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 dãy A[] = {1, 10, 5, 2, 7} và X = 10 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	3
6 51	-1
1 4 45 6 0 19	
3 8	
1 2 4	

86. **Range Query 8.** 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 $(12-5-6+30)/4=51$.

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 trống.
- $T, n, k, A[i]$ 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:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	15 20
5 2	34 56
10 4 5 15 20	
2	
4 2	
-12 34 56 7	

87. **Range Query 9.** 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 (Increament 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 Increament 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 trống.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A[i] \leq 10^3$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7
3	4
16 16 16	
2	
2 3	

88. **Range Query 10.** Cho mảng $A[]$ gồm n số nguyên dương. Hãy biến mảng $A[]$ thành một mảng đối xứng sao cho phép thay thế $A[i] = \text{Merge}(A[i], A[i+1])$ 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 trở 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 trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A[i] \leq 10^3$.

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	

89. **Range Query 11.** 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 5 = 1-(-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 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 trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A[i] \leq 10^3$.

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	

90. **Range Query 12.** 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 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:
1	4
6	
0 1 0 0 0 0	
1 0 1 0 0 1	

91. **Sorting 1.** 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 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	7 1 6 2 5 4 3
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	

92. **Sorting 2.** 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 kể trên: $A[] = \{5, 9, 10, 3, 2\}$ vì $|7-10|=3, |7-5|=2, |7-3|=4, |7-9|=2, |7-2|=5$.

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 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, 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	

93. **Sorting 3.** 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: $\text{Swap}(A[0], A[3]), \text{Swap}(A[1], A[2])$.

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	2
4	2
4 3 2 1	

5 1 5 4 3 2	
----------------	--

94. **Sorting 4.** 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}.

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 trống.
- 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	

95. **Sorting 5.** 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 trống.
- 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 1 5 2 3 6	
3 8 6 20 7	

96. **Sorting 6.** 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 trống.
- 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	

97. **Sorting 7.** Cho mảng $A[]$ gồm n phần tử. Hãy tìm dãy con liên tục của mảng $A[R], \dots, A[L]$ 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 $A[3], \dots, A[8]: \{30, 25, 40, 32, 31, 35\}$ 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	3 8
11	2 5
10 12 20 30 25 40 32 31 35 50 60	
9	
0 1 15 25 6 7 30 40 50	

98. **Searching 1.** Cho mảng $A[]$ gồm n phần tử. Hãy tìm vị trí của phần tử 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, 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 trống.
- 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	4
5 16	-1
9 7 2 16 4	
7 98	
1 22 57 47 34 18 66	

99. **Searching 2.** 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 trống.
- 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	1
5 16	-1
9 7 2 16 4	
7 98	
1 22 57 47 34 18 66	

100. **Searching 3.** Cho mảng $A[]$ gồm $n-1$ phần tử bao gồm các khác nhau từ 1, 2, ..., n . Hãy tìm số 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 trống.
- 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	

101. **Searching 4.** 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 $A[]$ với thời gian $\log(n)$.

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 trống.
- $T, n, A[i], x$ 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	8
10 3	2
5 6 7 8 9 10 1 2 3 4	
10 3	
1 2 3 4 5 6 7 8 9 10	

102. **Searching 5.** 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
---	-----------

103. **Searching 6.** 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	

104. **Searching 7.** 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 trống.
- 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 tổng gần nhất với 0 của cặp phần tử.

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

105. **Searching 8.** 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 trống.
- 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 3	
7 4	
1 1 2 2 2 3	

106. **Searching 9.** 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 trống.
- $T, N, X, A[i]$ 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 3	
7 4	
1 1 2 2 2 3	

107. **Searching 10.** 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	

108. **Searching 11.** Cho mảng $A[]$ gồm N phần tử được chia thành hai phần: phần thứ nhất bao gồm các số được sắp theo thứ tự tăng dần; phần còn lại được sắp theo thứ tự giảm dần. Nhiệm vụ của bạn là tìm số X có thuộc mảng $A[]$ hay không với thời gian $O(\log(n))$. Nếu tìm thấy X trong mảng $A[]$ hãy đưa ra vị trí của X 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 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: $1 \leq T \leq 100$; $1 \leq N \leq 10^7$, $-10^7 \leq X, A[i] \leq 10^7$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

10 4	
1 2 3 5 10 9 8 7 6 -1	

109. **Searching 12.** 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:

- 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 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	

110. **Searching 13.** Mảng $A[]$ được gọi là mảng sắp xếp vòng nếu $A[]$ được chia thành hai đoạn đã được sắp xếp. Đọc từ phần tử nhỏ nhất đến phần tử cuối cùng và vòng lại các phần tử ở phần còn lại ta sẽ được một dãy đã được sắp xếp. Ví dụ dãy $A[] = \{5, 6, 1, 2, 3, 4\}$ là mảng sắp xếp vòng. Nhiệm vụ của bạn là tìm phần tử nhỏ nhất của mảng sắp xếp vòng với thời gian $O(\log(n))$.

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: $1 \leq T \leq 100$; $1 \leq N \leq 10^5$, $1 \leq A[i] \leq 10^5$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
5	5
4 5 1 2 3	
6	
10 20 30 40 50 5 7	

111. **Searching 14.** 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.

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	1
5	20
4 5 1 2 1	
6	
10 20 30 30 20 5 7	

112. **Searching 17.** Cho mảng $A[]$ gồm N phần tử đã được sắp xếp. Hãy tìm K phần tử gần nhất của 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. Chú ý: X có thể có mặt hoặc 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 số phần tử của mảng N , K , X ; dòng tiếp theo là N số $A[i]$ là các phần tử của mảng $A[]$.
- $T, N, K, X, A[i]$ 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.

Input:	Output:
1 5 11 23 24 75 89 2 24	23 75

113. **Searching 18.** Cho mảng $A[], B[], C[]$ gồm $N1, N2, N3$ phần tử đã được sắp xếp. Hãy đưa ra các phần tử có mặt trong cả ba mảng theo thứ tự tăng 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 bốn dòng: dòng thứ nhất đưa vào $N1, N2, N3$ là số phần tử của mảng $A[], B[], C[]$; các dòng tiếp theo đưa vào $N1$ số $A[i]$, $N2$ số $B[j]$, $N3$ số $C[k]$.
- $T, N1, N2, N3, A[i], B[j], C[k]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N1, N2, N3 \leq 10^6$, $0 \leq A[i], B[j], C[k] \leq 10^{18}$.

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	20 80

114. **Searching 19.** 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