

BÀI TẬP

1. **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	

2. **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	

3. **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 3 5 7 2	-1
--------------------	----

4. **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	

5. **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	

6. **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.

Thao tác 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

7. **Next Greater Element.** Cho mảng A[] gồm n số khác nhau. Hãy đưa ra số lớn hơn tiếp theo về bên phải của mảng. Nếu không có số nào lớn hơn hãy đưa ra -1.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi test gồm hai dòng: dòng thứ nhất là số phần tử của mảng n; dòng tiếp theo đưa vào n phần tử khác nhau của mảng A[].
- 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 4 1 3 2 4 4 4 3 2 1	3 4 4 -1 -1 -1 -1 -1

8. **Max-Min-Sub Array.** Cho mảng các số nguyên A[] gồm N phần tử. Ta gọi $\text{Min}(A[], n, k)$ là dãy số bao gồm các số nhỏ nhất của dãy con liên tục k phần tử trong A[]. Ta gọi $\text{Max}(\text{Min}(A[], n, k))$ là số lớn nhất trong số các số nhỏ nhất của tất cả các dãy con k phần tử liên tục của dãy số k. Hãy tìm dãy kết quả $R[] = \{\text{Max}(\text{Min}(A[], n, 1)), \text{Max}(\text{Min}(A[], n, 2)), \dots, \text{Max}(\text{Min}(A[], n, n))\}$. Ví dụ với dãy A[] = {9, 7, 12, 8, 6, 5}, n=6 ta tính toán được dãy kết quả R[] như sau:

$k=1$: $\text{Min}(A[], n, 1) = \{9, 7, 12, 8, 6, 5\}$; $\text{Max}(\text{Min}(A[], 6, 1)) = 12$.
 $k=2$: $\text{Min}(A[], n, 2) = \{7, 7, 8, 6, 5\}$; $\text{Max}(\text{Min}(A[], 6, 2)) = 8$.
 $k=3$: $\text{Min}(A[], n, 3) = \{7, 7, 6, 5\}$; $\text{Max}(\text{Min}(A[], 6, 3)) = 7$.
 $k=4$: $\text{Min}(A[], n, 4) = \{7, 6, 5\}$; $\text{Max}(\text{Min}(A[], 6, 4)) = 7$.
 $k=5$: $\text{Min}(A[], n, 5) = \{6, 5\}$; $\text{Max}(\text{Min}(A[], 6, 5)) = 6$.
 $k=6$: $\text{Min}(A[], n, 6) = \{5\}$; $\text{Max}(\text{Min}(A[], 6, 6)) = 5$.

Từ đó ta có dãy kết quả $R[] = \{12, 8, 7, 7, 6, 5\}$.

Input:

- Dòng đầu tiên đưa vào số lượng test T ($T \leq 100$).
- Những dòng kế tiếp đưa vào các test. Mỗi test được tổ chức thành 2 dòng. Dòng thứ nhất ghi lại số N là số lượng phần tử của mảng A[] ($1 \leq N \leq 1000$). Dòng thứ 2 đưa vào N số A[i] ($(1 \leq A[i] \leq 5000)$).

Output:

- Đưa ra kết quả mỗi test theo từng dòng là k-size ($k\text{-size} = 1, 2, \dots, n$).

Input	Output
2 6 9 7 12 8 6 5	12 8 7 7 6 5 70 30 20 10 10 10 10

7 10 20 30 50 10 70 30	
---------------------------	--

9. **Finited Turing Machine.** Máy Turing hữu hạn là một máy gồm N trạng thái. Ứng với trạng thái i ($0 \leq i < N$) ta ghi lại một số nguyên không âm được gọi là giá trị trạng thái T_i ($0 \leq i < N$). Máy Turing hữu hạn với giá trị các trạng thái $T = \{T_0, T_1, \dots, T_{N-1}\}$ chỉ có thể thực hiện được ba thao tác cơ bản dưới đây:

- **Thao tác R (Right):** Di chuyển đến trạng thái tiếp theo ($T_i \rightarrow T_{i+1}$).
- **Thao tác L (Left):** Lùi lại trạng thái trước đó ($T_i \rightarrow T_{i-1}$).
- **Thao tác G (Go To):** Nhảy đến trạng thái có giá trị cùng với trạng thái hiện tại với khoảng cách ít nhất hai trạng thái ($T_i \rightarrow T_j : T_i = T_j$).

Cho hai trạng thái P, Q bất kỳ. Hãy dịch chuyển trạng thái P về trạng thái Q sao cho các thao tác L, R, G cần được thực hiện là ít nhất. Ví dụ với máy Turing gồm 7 trạng thái $T[] = \{5, 4, 2, 3, 5, 5, 0\}$, $P=1$, $Q=6$ khi đó các thao tác L, R, G ít nhất cần thực hiện là LGR: Lùi về trạng thái số 0 ($T_0=5$) bằng phép L , di chuyển đến vị trí số 5 bằng phép G ($T_5=5$), dịch phải một bước bằng phép R ($T_6=0$).

Input: Dòng đầu tiên đưa vào số lượng bộ test T ($T \leq 100$). Những dòng tiếp theo đưa vào T bộ test. Mỗi bộ test gồm hai phần. Phần thứ nhất ghi lại bộ 3 N, P, Q . Phần thứ hai ghi lại N số nguyên không âm là giá trị N trạng thái $T = \{T_0, T_1, \dots, T_{N-1}\}$ của máy Turing. Các số T_i, N, P, Q thỏa mãn ràng buộc $0 \leq T_i \leq 1000; 1 < N \leq 1000; 0 \leq P, Q \leq N$.

Output: Đưa ra kết quả mỗi test theo từng dòng. Kết quả mỗi test là xâu nhỏ nhất các phép dịch chuyển L, R, G .

Input	Output
2	GR
7 0 6	LGR
5 4 2 3 5 5 0	
7 1 6	
5 4 2 3 5 5 0	

10. **Knight tour move.** Cho bàn cờ kích cỡ $N \times N$ cùng với tọa độ hai ô bắt đầu (beg) và kết thúc (end). Hãy tìm số bước di chuyển ít nhất của quân mã đi từ ô beg đến ô kết thúc.

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 hai số là tọa độ ô beg, dòng thứ hai đưa vào hai số là tọa độ ô end. Các số được viết cách nhau một vài khoảng trống.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq N \leq 20$.

Output: Đưa ra theo từng dòng kết quả của mỗi test.

Input:	Output:
2	3
6	9
4 5	
1 1	
20	
5 7	
15 20	