**[Chia Và Trị]. Bài 1. Lũy thừa nhị phân**

Nhiệm vụ của bạn là tính N^K, kết quả được chia dư với 10^9 + 7

**Input Format**

Dòng duy nhất chứa 2 số N và K

**Constraints**

1<=N<=10^6; 1<=K<=10^9;

**Output Format**

Kết quả của bài toán

**Sample Input 0**

9 9

**Sample Output 0**

387420489

**Sample Input 1**

994402 999992619

**Sample Output 1**

925329307

**[Chia Và Trị]. Bài 2. Lũy thừa nhị phân đảo**

Cho số nguyên dương N, gọi M là số đảo của N. Hãy tính lũy thừa với cơ số N và số mũ M và chia dư kết quả cho 10^9 + 7

**Input Format**

Dòng duy nhất chứa số nguyên dương N

**Constraints**

1<=N<=10^9

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

8

**Sample Output 0**

16777216

**Sample Input 1**

999995841

**Sample Output 1**

987348397

**[Chia Và Trị]. Bài 3. Đếm dãy số**

Cho số nguyên dương n, bạn được phép sử dụng không giới hạn các số tự nhiên từ 1 tới n. Hỏi có bao nhiêu cách chọn ra 1 dãy có tổng các phần tử bằng n.

**Input Format**

Dòng duy nhất chứa số nguyên dương n

**Constraints**

1<=n<=10^12

**Output Format**

In ra đáp án của bài toán sau khi chia dư với 10^9 + 7

**Sample Input 0**

6

**Sample Output 0**

32

**[Chia Và Trị]. Bài 4. Gấp đôi dãy số**

Một dãy số tự nhiên bắt đầu bởi con số 1 và được thực hiện N-1 phép biến đổi “gấp đôi” dãy số như sau: Với dãy số A hiện tại, dãy số mới có dạng A, x, A trong đó x là số tự nhiên bé nhất chưa xuất hiện trong A.

Ví dụ với 2 bước biến đổi, ta có [1] - [1 2 1] - [1 2 1 3 1 2 1].

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

**Input Format**

Dòng duy nhất chứa 2 số nguyên dương N và K

**Constraints**

1<=N<=50; 1<=K<=2^N - 1

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

3 2

**Sample Output 0**

2

**[Chia Và Trị]. Bài 5. Xâu Fibo**

Xâu S chỉ bao gồm các kí tự A và B được gọi là xâu Fibonacci nếu S(1) = 'A', S(2) = 'B', S(n) = S(n - 2) + S(n - 1), trong đó phép + là phép nối 2 xâu. Bài toán đặt ra là tìm kí tự thứ K trong xâu fibonacci thứ N.

**Input Format**

Dòng duy nhất chứa 2 số N và K

**Constraints**

1<=N<=92; 1<=K<=7e18 và đảm bảo là vị trí hợp lệ trong xâu fibonacci thứ N

**Output Format**

In ra kí tự tìm được

**Sample Input 0**

5 3

**Sample Output 0**

B

**[Chia Và Trị]. Bài 6. Maximum sub array**

Cho dãy số nguyên A[] gồm N phần tử, nhiệm vụ của bạn là tính tổng của dãy con có tổng lớn nhất.

**Input Format**

Dòng đầu tiên chứa số nguyên dương N; Dòng thứ 2 chứa N số nguyên của dãy A[]

**Constraints**

1<=N<=10^5; 1<=abs(A[i])<=10^6

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

5

-3 6 1 -3 6

**Sample Output 0**

10

**[Chia Và Trị]. Bài 7. Floor**

Cho mảng đã được sắp xếp A[] gồm N phần tử không có hai phần tử giống nhau và số X. Nhiệm vụ của bạn là tìm floor(X). Trong đó, K=floor(X) là phần tử lớn nhất trong mảng A[] nhỏ hơn hoặc bằng X.

**Input Format**

Dòng 1 chứa 2 số N và X; Dòng 2 chứa N số của mảng A[]

**Constraints**

1<=N<=10^5; 1<=X,A[i]<=10^6;

**Output Format**

In ra đáp án của bài toán nếu tìm thấy số lớn hơn X, trường hợp không tìm thấy in -1

**Sample Input 0**

9 8

1 2 3 4 5 6 7 8 9

**Sample Output 0**

8

**[Chia Và Trị]. Bài 8. Trộn 2 mảng**

Cho hai mảng đã được sắp xếp A[], B[] gồm N, M phần tử theo thứ tự và số K. Nhiệm vụ của bạn là tìm phần tử ở vị trí số K sau khi trộn hai mảng để nhận được một mảng được sắp xếp.

**Input Format**

Dòng đầu tiên chứa 3 số N, M, K; Dòng thứ 2 chứa N số nguyên của mảng A[]; Dòng thứ 3 chứa M số nguyên của mảng B[];

**Constraints**

1<=N,M<=10^4; 1<=K<=N+M; 1<=A[i],B[i]<=10^6

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

7 9 14

4 6 7 9 10 10 10

1 1 2 5 7 8 8 9 10

**Sample Output 0**

10

**[Chia Và Trị]. Bài 9. Phần tử khác nhau**

Cho hai mảng đã được sắp xếp A[] và B[] gồm N và N-1 phần tử. Các phần tử của mảng A[] chỉ khác mảng B[] một phần tử duy nhất. Hãy tìm vị trí của phần tử khác nhau giữa A[] và B[].

**Input Format**

Dòng đầu chứa số nguyên dương N; Dòng thứ 2 chứa các phần tử của mảng A[]; Dòng thứ 3 chứa các phần tử của mảng B[];

**Constraints**

1<=N<=10^5; 1<=A[i], B[i]<=10^6;

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

8

1 2 3 4 5 6 7 8

1 2 3 4 5 6 8

**Sample Output 0**

7

**[Chia Và Trị]. Bài 10. Đếm số 0**

Cho mảng A[] gồm N phần tử chỉ bao gồm các số 0 và 1. Các số 0 được đặt trước các số 1. Hãy đếm các số 0 với thời gian log(N).

**Input Format**

Dòng 1 chứa số nguyên dương N; Dòng thứ 2 chứa N số nguyên của mảng A[]

**Constraints**

1<=N<=10^6; 0<=A[i]<=1;

**Output Format**

In ra đáp án của bài toán, nếu số 0 không xuất hiện in ra 0

**Sample Input 0**

6

0 0 1 1 1 1

**Sample Output 0**

2

**[Chia Và Trị]. Bài 11. Tìm kiếm nhị phân**

Cho dãy A[] gồm N phần tử đã được sắp xếp và số nguyên dương K. Hãy kiểm tra xem K có xuất hiện trong mảng hay không, nếu có in ra YES, ngược lại in ra NO

**Input Format**

Dòng 1 chứa 2 số N và K; Dòng 2 chứa N số nguyên của mảng A[]

**Constraints**

1<=N<=10^5; 1<=A[i], K<=10^6;

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

7 2

1 8 9 10 10 12 17

**Sample Output 0**

NO

**Sample Input 1**

6 4

4 8 9 16 19 20

**Sample Output 1**

YES

**[Chia Và Trị]. Bài 12. Số Fibonacci thứ N**

Tìm số Fibonacci thứ N sau khi chia dư kết quả cho 10^9 + 7

**Input Format**

Dòng duy nhất chứa số nguyên dương N;

**Constraints**

1<=N<=10^10

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

58

**Sample Output 0**

286725742

**Sample Input 1**

36

**Sample Output 1**

14930352

**[Chia Và Trị]. Bài 13. Dãy xâu nhị phân**

Xâu nhị phân S được định nghĩa như sau :

S[1] = “0”.

S[2] = “1”.

S[i] = S[i-2] + S[i-1] với i>2.

Với phép cộng (+) là phép nối hai xâu với nhau. Hãy xác định ký tự thứ K trong xâu S[N] là ký tự ‘0’ hay ký tự ‘1’.

**Input Format**

1 Dòng duy nhất chứa 2 số nguyên N và K

**Constraints**

2<=N<=92; 1<=K<=7e18 và đảm bảo là một vị trí hợp lệ trong xâu S

**Output Format**

In ra kí tự là đáp án của bài toán

**Sample Input 0**

7 7

**Sample Output 0**

0

**Sample Input 1**

8 6

**Sample Output 1**

1

**[Chia Và Trị]. Bài 14. Dãy số vô hạn SPOJ**

Dãy số nguyên dương vô hạn A[] được định nghĩa một cách đệ quy như sau: A[0] = 0; A[1] = 1; A[i] = A[i-1] + A[i-2] với mọi n ≥ 2. Cho số nguyên dương N không quá 10^9. Hãy xác định số thứ N trong dãy A[].

**Input Format**

Dòng duy nhất chứa số nguyên dương N

**Constraints**

1<=N<=10^9

**Output Format**

In ra đáp án của bài toán sau khi chia dư cho 1e9 + 7

**Sample Input 0**

7

**Sample Output 0**

13

**Sample Input 1**

999992343

**Sample Output 1**

865905443

**[Chia Và Trị]. Bài 15. Lũy thừa ma trận**

Cho ma trận A[][] là ma trận vuông cỡ N x N. Hãy tính kết quả A^K và chia dư các phần tử trong ma trận kết quả cho 10^9 + 7

**Input Format**

Dòng đầu tiên chứa 2 số nguyên dương N và K; N dòng tiếp theo là các phần tử trong ma trận A

**Constraints**

1<=N<=10; 1<=K<=10^9; 0<=A[i][j]<=1000

**Output Format**

In ra ma trận đáp án

**Sample Input 0**

2 3

712 141

55 619

**Sample Output 0**

376787593 188740908

73622340 252298909

**[Chia Và Trị]. Bài 16. Lũy thừa ma trận 2**

Cho ma trận vuông A[][] kích thước N x N. Nhiệm vụ của bạn là hãy tính ma trận A^K với K là số nguyên cho trước. Sau đó, tính tổng các phần tử của cột cuối cùng. Đáp số có thể rất lớn, hãy in ra kết quả theo modulo 10^9+7.

**Input Format**

Dòng đầu tiên chứa 2 số nguyên dương N và K; N dòng tiếp theo là các phần tử trong ma trận A

**Constraints**

1<=N<=10; 1<=K<=10^9; 0<=A[i][j]<=1000

**Output Format**

In ra đáp án của bài toán theo yêu cầu

**Sample Input 0**

2 2

216 71

923 906

**Sample Output 0**

966031

**[Chia Và Trị]. Bài 17. Count Inversion**

Cho mảng A[] gồm N phần tử, hãy đếm số lượng cặp A[i], A[j] sao cho i < j và A[i] > A[j];

**Input Format**

Dòng đầu tiên chứa số nguyên dương N; Dòng thứ 2 gồm N phần tử của mảng A

**Constraints**

1<=N<=10^6; 1<=A[i]<=10^6;

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

9

16 14 10 13 19 14 17 18 16

**Sample Output 0**

12

**[Chia Và Trị]. Bài 18. Đếm số bit 1**

Cho số nguyên dương N. Mỗi bước, bạn sẽ biến đổi N thành [N/2], N mod 2, [N/2]. Sau khi thực hiện một cách triệt để, ta thu được một dãy số chỉ toàn số 0 và 1. Nhiệm vụ của bạn là hãy đếm các số bằng 1 trong đoạn [L, R] của dãy số cuối cùng.

**Input Format**

Dòng duy nhất chứa 3 số nguyên N, L, R;

**Constraints**

1<=N,L<=R<=10^10; 0<=R-L<=100000;

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

7 2 5

**Sample Output 0**

4

**Explanation 0**

Test : [7] => [3, 1, 3] => [1, 1, 1, 1, 3] => [1, 1, 1, 1, 1, 1, 1]

## [Chia Và Trị]. Bài 19. Merge sort

* [**Problem**](https://www.hackerrank.com/contests/dsa-t2h71-211023-21chia-va-tr/challenges/chia-va-tr-bai-19-merge-sort)
* [**Submissions**](https://www.hackerrank.com/contests/dsa-t2h71-211023-21chia-va-tr/challenges/chia-va-tr-bai-19-merge-sort/submissions)
* [**Discussions**](https://www.hackerrank.com/contests/dsa-t2h71-211023-21chia-va-tr/challenges/chia-va-tr-bai-19-merge-sort/forum)

Cho mảng A[] có N phần tử, bạn hãy cài đặt thuật toán Merge Sort - Sắp xếp trộn để sắp dãy số tăng dần sau đó in ra màn hình.

**Input Format**

* Dòng 1 là N : Số phần tử trong mảng
* Dòng 2 gồm N phần tử trong mảng A[]

**Constraints**

* 1<=N<=10^6
* 0<=A[i]<=10^9

**Output Format**

In ra dãy số sau khi sắp xếp

**Sample Input 0**

10

231 449 632 324 260 898 609 654 950 700

**Sample Output 0**

231 260 324 449 609 632 654 700 898 950

## [Chia Và Trị]. Bài 20. Quicksort 1

* [**Problem**](https://www.hackerrank.com/contests/dsa-t2h71-211023-21chia-va-tr/challenges/chia-va-tr-bai-20-quicksort-1)
* [**Submissions**](https://www.hackerrank.com/contests/dsa-t2h71-211023-21chia-va-tr/challenges/chia-va-tr-bai-20-quicksort-1/submissions)
* [**Discussions**](https://www.hackerrank.com/contests/dsa-t2h71-211023-21chia-va-tr/challenges/chia-va-tr-bai-20-quicksort-1/forum)

Cho mảng A[] có N phần tử, bạn hãy cài đặt thuật toán Quick sort - Sắp xếp nhanh để sắp dãy số tăng dần sau đó in ra màn hình. Sử dụng phân hoạch Lomuto

**Input Format**

* Dòng 1 là N : Số phần tử trong mảng
* Dòng 2 gồm N phần tử trong mảng A[]

**Constraints**

* 1<=N<=10^6
* 0<=A[i]<=10^9

**Output Format**

In ra dãy số sau khi sắp xếp

**Sample Input 0**

10

139 871 801 611 301 713 413 970 786 659

**Sample Output 0**

139 301 413 611 659 713 786 801 871 970

## [Chia Và Trị]. Bài 21. Quicksort 2

* [**Problem**](https://www.hackerrank.com/contests/dsa-t2h71-211023-21chia-va-tr/challenges/chia-va-tr-bai-21-quicksort-2)
* [**Submissions**](https://www.hackerrank.com/contests/dsa-t2h71-211023-21chia-va-tr/challenges/chia-va-tr-bai-21-quicksort-2/submissions)
* [**Discussions**](https://www.hackerrank.com/contests/dsa-t2h71-211023-21chia-va-tr/challenges/chia-va-tr-bai-21-quicksort-2/forum)

Cho mảng A[] có N phần tử, bạn hãy cài đặt thuật toán Quick sort - Sắp xếp nhanh để sắp dãy số tăng dần sau đó in ra màn hình. Sử dụng phân hoạch Hoare

**Input Format**

* Dòng 1 là N : Số phần tử trong mảng
* Dòng 2 gồm N phần tử trong mảng A[]

**Constraints**

* 1<=N<=10^6
* 0<=A[i]<=10^9

**Output Format**

In ra dãy số sau khi sắp xếp

**Sample Input 0**

10

413 348 77 923 538 154 462 450 71 733

**Sample Output 0**

71 77 154 348 413 450 462 538 733 923

## [Chia Và Trị]. Bài 22. 28tech C++ DSA JAVA

* [**Problem**](https://www.hackerrank.com/contests/dsa-t2h71-211023-21chia-va-tr/challenges/chia-va-tr-bai-22-28tech-c-dsa-java)
* [**Submissions**](https://www.hackerrank.com/contests/dsa-t2h71-211023-21chia-va-tr/challenges/chia-va-tr-bai-22-28tech-c-dsa-java/submissions)
* [**Discussions**](https://www.hackerrank.com/contests/dsa-t2h71-211023-21chia-va-tr/challenges/chia-va-tr-bai-22-28tech-c-dsa-java/forum)

Cho xâu F được định nghĩa như sau : F(1) = "28tech", F(2) = "C++", F(3) = "DSA JAVA" F(N) = F(N - 3) + " " + F(N -2) + " " + F(N - 1). Vì thế F(4) = "28tech C++ DSA JAVA", F(5) = "C++ DSA JAVA 28tech C++ DSA JAVA". Nhiệm vụ của bạn là tìm từ thứ K trong xâu F(N), dữ liệu đảm bảo xâu F(N) có từ thứ K

**Input Format**

* Dòng 1 là T : số bộ test
* T dòng tiếp theo mỗi dòng gồm 2 số N, K

**Constraints**

* 1<=T<=100
* 1<=N<=60
* 1<=K<=10^16

**Output Format**

In ra kết quả của mỗi test trên từng dòng

**Sample Input 0**

10

3 2

6 2

10 24

4 2

5 4

6 12

7 22

8 1

7 18

8 14

**Sample Output 0**

JAVA

JAVA

JAVA

C++

28tech

DSA

C++

C++

C++

C++