Power(O(logn))

Take as input x and n, two numbers. Write a function to calculate x raise to power n. Target complexity is O(logn). NOTE: Try both recursive and bitmasking approach.

Input Format:

Enter the number N and its power P

Constraints:

None

output format

Display N^P

sample input

2

3

sample output

8

Birthday Paradox

Your task is simple, determine the minimum number of people required in the room so that two have same birthday with probability greater than or equal to 'p'. (Assume there are 365 days in every year)

Input Format:

Single line containing the given probability.

Constraints:

0<p<=1

output format

Print an integral value denoting minimum number of people.

sample input

0.5

sample output

23

Boston Numbers

A Boston number is a composite number, the sum of whose digits is the sum of the digits of its prime factors obtained as a result of prime factorization (excluding 1 ). The first few such numbers are 4,22 ,27 ,58 ,85 ,94 and 121 . For example, 378 = 2 × 3 × 3 × 3 × 7 is a Boston number since 3 + 7 + 8 = 2 + 3 + 3 + 3 + 7. Write a program to check whether a given integer is a Boston number.

Input Format:

There will be only one line of input:N , the number which needs to be checked.

Constraints:

1 < N < 2,147,483,647 (max value of an integer of the size of 4 bytes)

output format

1 if the number is a Boston number. 0 if the number is a not Boston number.

sample input

378

sample output

1

MARBLES

Hänschen dreams he is in a shop with an infinite amount of marbles. He is allowed to select n marbles. There are marbles of k different colors. From each color, there are also infinitely many marbles. Hänschen wants to have at least one marble of each color, but still there are a lot of possibilities for his selection. In his effort to make a decision he wakes up. Now he asks you how many possibilities for his selection he would have had. Assume that marbles of equal color can't be distinguished, and the order of the marbles is irrelevant.

Input Format:

The first line of input contains a number 'T' that indicates the number of test cases to follow. Each test case consists of one line containing n and k, where n is the number of marbles Hänschen selects and k is the number of different colors of the marbles.

Constraints:

1<=T<=100 1<=k<=n<=1000000

output format

For each test case print the number of possibilities that Hänschen would have had. You can assume that this number exceeds the limit of a signed 64-bit integer.

sample input

2

10 10

30 7

sample output

1

475020

A Factorial Problem

Mike found an interesting problem. In this problem you are given two integers, *n* and *k*. You need to find the the maximum value of *x*, such that, *n!* % *kx* = 0.

Input Format:

First line contains number of test cases, T. Each test case contains two integers each, *n* and *k*.

Constraints:

1<=T<=20 1<=n<=10^8 2<=k<=10^8

output format

Print the value of *x* for each test case.

sample input

2

5 2

1000 2

sample output

3

994

Catalan Numbers

Print nth [Catalan Number](https://en.wikipedia.org/wiki/Catalan_number). The first few Catalan numbers for n = 0, 1, 2, 3, … are 1, 1, 2, 5, 14,.. .

Input Format:

One integer n

Constraints:

1 <= N <= 100

output format

Print the catalan number at position N.

sample input

10

sample output

16796

RANDOM QUERY

You are given an array **a** consisting of **n** positive integers. You pick two integer numbers **l** and **r** from **1** to **n**, inclusive (numbers are picked randomly, equiprobably and independently). If **l** > **r**, then you swap values of **l** and **r**. You have to calculate the expected value of the number of unique elements in segment of the array from index **l** to index **r**, inclusive (1-indexed).

Input Format:

The first line contains one integer number The first line contains one integer number **n** . The second line contains **n** integer numbers **a1, a2, … an** — elements of the array. **(1 ≤ ai ≤ 106)**

Constraints:

1 ≤ n ≤ 10^6

output format

Print one number — the expected number of unique elements in chosen segment.   
**NOTE:** Print your answer upto the precision of 6 decimal places.

sample input

2

1 2

sample output

1.500000

Math Day

Math Day is being celebrated at Coding Blocks. So Prateek Bhaiya rolled out a contest on Maths Problems. Here goes one.  
Given three positive integers **A,N,P**. Compute **AN! %P**.

Input Format:

The first line of the input gives the number of test cases, **T**. **T** lines follow.   
Each line contains three integers **A, N** and **P**, as described above.

Constraints:

1 ≤ T ≤ 100 1 ≤ A ≤ 10^5. 1 ≤ N ≤ 10^5. 1 ≤ P ≤ 10^5.

output format

For every test case, print the answer in a new line.

sample input

2

2 1 2

3 3 2

sample output

0

1

Evaluating functions

Alice is learning mathematical functions. In his assignment, teacher has asked him to evaluate h(x)= f(x)+g(x) where   
f(x)=3x2-x+10 and   
g(x)=4x3+2x2-5x+4.  
As he is weak in mathematics, help him finish this task.

Hint: Since X is large, you need to work with Big Integers!

Input Format:

Single line containing an integral value denoting the value of x.

Constraints:

0<=x<=10^50

output format

Print integral value denoting value of h(x).

sample input

1

sample output

17

DIVISIBLE SUBARRAYS

You are given N elements, *a1,a2,a3….aN*. Find the number of *good* sub-arrays.  
A good sub-array is a sub-array [*ai,ai+1,ai+2….aj*] such that (*ai+ai+1+ai+2+….+aj*) is divisible by N.

Input Format:

The first line contains the number of test cases T. First line of each test case contains an integer N denoting the number of elements. Second line of each test case contains N integers, a1, a2, a3….aN, where ai denotes the ith element of the array.

Constraints:

1<=T<=10 1<=N<=10^5 |ai|<=10^9

output format

Output a single integer denoting the number of good sub-arrays.

sample input

2

5

1 1 1 1 1

5

5 5 5 5 5

sample output

1

15

FAST FIBONACCI

Fibonacci series is well-known series in which in which each number (Fibonacci number) is the sum of the two preceding numbers. The series looks like *1, 1, 2, 3, 5, 8….* and so on. Your task is to find *nth* number.  
Since the number can be large, output the answer modulo (109+7).

Input Format:

An integer T, denoting the number of test cases. Each test case contains and integer N.

Constraints:

1<=T<=10^5 1<=N<=10^9

output format

Print the *nth* Fibonacci number modulo (109+7).

sample input

4

3

4

5

6

sample output

2

3

5

8

SEQ - Recursive Sequence

Sequence (ai) of natural numbers is defined as follows:   
ai = bi (for i <= k)   
ai = c1ai-1 + c2ai-2 + … + ckai-k (for i > k)   
where bj and cj are given natural numbers for 1<=j<=k. Your task is to compute an for given n and output it modulo 109.   
For stricter evaluation of your code, try submitting your problem on [SPOJ](http://www.spoj.com/problems/SEQ/).

Input Format:

On the first row there is the number **C** of test cases   
Each test contains four lines:

**k** - number of elements of (c) and (b)   
**b1,…,bk** - k natural numbers where 0 <= **bj** <= 109 separated by spaces   
**c1,…,ck** - k natural numbers where 0 <= **cj** <= 109 separated by spaces   
**n** - natural number

Constraints:

1 <= C <= 1000 1 <= k <= 10 1 <= n <= 10^9

output format

Exactly **C** lines, one for each test case: **an modulo 109**.

sample input

sample output

SPP - Recursive Sequence (Version II)

Sequence (ai) of natural numbers is defined as follows:

ai = bi (for i <= k)

ai = c1ai-1 + c2ai-2 + … + ckai-k (for i > k)

where bj and cj are given natural numbers for 1<=j<=k. Your task is to compute am + am+1 + am+2 + … + an for given m <= n and output it modulo a given positive integer p.

Input Format:

On the first row there is the number C of test cases (equal to about 50). Each test contains four lines, k : number of elements of (c) and (b) (1 <= k <= 15) b1,…,bk : k natural numbers where 0 <= bj <= 109 separated by spaces c1,…,ck : k natural numbers where 0 <= cj <= 109 separated by spaces m, n, p : natural numbers separated by spaces (1 <= m <= n <= 10^18, 1<= p <= 10^8)

Constraints:

output format

Exactly C lines, one for each test case: (am + am+1 + am+2 + … + an) modulo p.

sample input

1

2

1 1

1 1

2 10 1000003

sample output

142

FIBOSUM

The Fibonacci sequence is defined by the following relation:

F(0) = 0  
F(1) = 1  
F(N) = F(N - 1) + F(N - 2), N >= 2   
Your task is very simple. Given two non-negative integers N and M, you have to calculate the sum (F(N) + F(N + 1) + … + F(M)) mod 1000000007.  
**Source:** [FIBOSUM (SPOJ)](http://www.spoj.com/problems/FIBOSUM/)

Input Format:

The first line contains an integer T (the number of test cases). Then, T lines follow. Each test case consists of a single line with two non-negative integers N and M.

Constraints:

1<=T<=1000 0<=N<=M<=10^9

output format

For each test case you have to output a single line containing the answer for the task.

sample input

3

0 3

3 5

10 19

sample output

4

10

10857

Dice expectations - 1

What is the expected number of dice throws required to get a "four"?

Choosing Numbers

N students are asked to choose a number from 1 to 100 inclusive. What is the expected number of students that would choose a single digit number?

Input Format:

An integer n denoting number of students

Constraints:

output format

A float denoting the answer upto 2 decimal point accuracy

sample input

8

sample output

0.72

Interview Candidates

Candidates are appearing for interview one after other. Probability of each candidate getting selected is 0.16. What is the expected number of candidates that you will need to interview to make sure that you select somebody?

Dice Roll Expectation

In a game of fair dice(6 sides) throw, what is the expected number of throws to make sure that all 6 outcomes appear atleast once? Round off the ans to next integer.

Head Expectation

What is the expected number of coin flips for getting N consecutive heads, given N?

Input Format:

An integer n

Constraints:

output format

An integer denoting the required answer

sample input

1

sample output

2

FAVDICE - Coupon Collector Problem

BuggyD loves to carry his favorite die around. Perhaps you wonder why it's his favorite? Well, his die is magical and can be transformed into an **N**-sided unbiased die with the push of a button. Now BuggyD wants to learn more about his die, so he raises a question:

What is the expected number of throws of his die while it has **N** sides so that each number is rolled at least once?

Input Format:

The first line of the input contains an integer **t**, the number of test cases. **t** test cases follow.

Each test case consists of a single line containing a single integer **N - the number of sides on BuggyD's die**

Constraints:

1 <= T <= 15 1 <= N <= 1000

output format

For each test case, print one line containing the expected number of times BuggyD needs to throw his **N**-sided die so that each number appears at least once. The expected number must be accurate to 2 decimal digits.

sample input

2

1

12

sample output

1.00

37.24

SUMSUMS - Summing Sums

The N (1 ≤ N ≤ 50,000) cows, conveniently numbered 1, 2, …, N, are trying to learn some encryption algorithms. After studying a few examples, they have decided to make one of their own! However, they are not very experienced at this, so their algorithm is very simple:

Each cow i is given a starting number Ci (0 ≤ Ci < 90,000,000), and then all the cows perform the following process in parallel:

First, each cow finds the sum of the numbers of the other N-1 cows. After all cows are finished, each cow replaces her number with the sum she computed. To avoid very large numbers, the cows will keep track of their numbers modulo 98,765,431. They told Canmuu the moose about it in November; he was quite impressed. Then one foggy Christmas Eve, Canmuu came to say:

"Your algorithm is too easy to break! You should repeat it T (1 ≤ T ≤ 1,414,213,562) times instead." Obviously, the cows were very frustrated with having to perform so many repetitions of the same boring algorithm, so after many hours of arguing, Canmuu and the cows reached a compromise: You are to calculate the numbers after the encryption is performed!

Input Format:

Line 1: Two space-separated integers: N and T. Lines 2..N+1: Line i+1 contains a single integer: Ci.

output format

Lines 1..N: Line i contains a single integer representing the number of cow i (modulo 98,765,431) at the end of the encryption.

sample input

3 4

1

0

4

sample output

26

25

29

: