今日头条杯
2018 年首届湖北省大学生程序设计竞赛
(网络赛)

The 2018 Hubei Collegiate Programming Contest (Online Round)

2018 年 04 月 15 日

April 15th, 2018

赞助商

Event Sponsor

[出题/验题人]

DoveCCL, Srdce, zmf1997, Team Preludes, Team SmoothLatte, Team WakeUp

预祝我校“序曲”队在 ACM/ICPC World Finals 2018 中取得好成绩

Problem A. GSS and CQ

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	512 mebibytes

`CQCQCQ this is BG6TOE Bravo Golf Six Tango Oscar Echo BG6TOE`

In amateur radio communication, every ham has his or her call sign, which is a sequence (the length of which is no more than 6) of letters and digits, for example, GSS's call sign is `BG6TOE`, where `B` means HAM from Mainland China, and `G` means the type of radio station, and `6` means Hubei Province, Anhui Province or Jiangxi Province, then `TOE` is the suffix, in some cases (e.g. just call someone who is on the channel), call `TOE` instead `BG6TOE` is enough.

To start a call, you cal use `CQ`, the correct way to call is to: Say `CQ` for three times, e.g. `CQCQCQ` and then Your call sign, e.g. `BG6TOE`, for three times, the second time should use NATO Alphabet.

Other parts are optional, for a typical `CQ` call, one may say:

`CQCQCQ this is <Call> <Call in NATO Alphabet> <Call>`

Given a list of call signs, your task is to determine what thay would say in the format above.

Table 1: NATO Alphabet

A	Alfa	J	Juliett	S	Sierra	1	One
B	Bravo	K	Kilo	T	Tango	2	Two
C	Charlie	L	Lima	U	Uniform	3	Three
D	Delta	M	Mike	V	Victor	4	Four
E	Echo	N	November	W	Whiskey	5	Five
F	Foxtrot	O	Oscar	X	Xray	6	Six
G	Golf	P	Papa	Y	Yankee	7	Seven
H	Hotel	Q	Quebec	Z	Zulu	8	Eight
I	India	R	Romeo	0	Zero	9	Nine

The pure text version of the table above can be found at the link below:

http://acm.whu.edu.cn/upload/2018/whupc/nato.txt

Input

Input contains multiple lines, please process to the end of input.

Each line contains a valid call sign.

Output

For each line of input, output one line with the CQ call in the description above.

Examples

standard input
BG6TOE bg6rr bA1Aa m0xpd
standard output
CQCQCQ tHis Is BG6ToE BraVo gOLf SiX tAngo oScaR eChO bg6tOe CQCQCQ this is BG6RR Bravo Golf Six Romeo Romeo BG6RR CQCQCQ this is BA1AA Bravo Alfa One Alfa Alfa BA1AA CQCQCQ this is M0XPD Mike Zero Xray Papa Delta M0XPD

Note

You should note that both input and output are Case Insensitive

Problem B. GSS and Interesting Sculpture

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	512 mebibytes

GSS is painting an strange sculpture, the sculpture contests of two balls and they may intersect. Your task is to calculate the area of the surface of the sculpture.

Input

Input contains multiple cases, please process to the end of input.

For each line, there are three integers, R, r, L, R and r the radius of the two balls, L is the distance of the center of two balls.

$0 < R, r, L \leq 100$, $|R - r| < L \leq |R + r|$

Output

For each input, output one line with the answer, the area of the surface of the sculpture. Let the standard answer be a, and your answer be b, your answer will be considered as correct if and only if $\frac{|a-b|}{\max(a,1)} < 10^{-6}$.

Examples

standard input	standard output
3 4 5	271.433605270158
3 3 3	169.646003293849
1 2 3	62.831853071796

Problem C. GSS and Bubble Sort

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	512 mebibytes

Do you remember the problem "Time Limit Exceeded" last year? Here is the code GSS wrote in that problem.

```
#include <stdio.h>
int main() {
  int n;
  int *vec;
  scanf("%d", &n);
  vec = malloc(sizeof(int) * n);
  for (int i = 0; i < n; i++) {
    int t;
    scanf("%d", &t);
    vec[i] = t;
  }
  for (int i = 0; i < n; i++)
    for (int j = 0; j < n - 1; j++)
      if (vec[j] > vec[j+1])
        swap(vec[j], vec[j+1]);
  /* some other code takes O(1) time */
  return 0;
}
```

As you are GSS's team mate, your task is to calculate the expected time the code will run with an input of size n, $(0 < n < 10^9)$. The time is measured by how many times the function `swap` is called. You should note that the input is a permutation of $\{1, 2, \ldots, n\}$ in the original problem.

Input

Input contains multiple (about 1000) test cases, please process to the end of input.

Each test cases contains an integer n as described above.

Output

For each test case, output one line with the answer.

If your answer is p/q, then you should output $p \times q^{10^9+5}$ module 1000000007 $(10^9 + 7)$.

Examples

standard input	standard output
1 2	0 500000004

Note

In the second sample, there are two possible input $\{1, 2\}$ and $\{2, 1\}$, so the expected time the function `swap` is called is $(0 + 1)/2 = 1/2$, and $1 \times 2^{10^9+5}$ module $10^9 + 7$ is 500000004.

Problem D. GSS and Version Control

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	512 mebibytes

GSS has invented a brand new version control system, called Gss's Interesting Theory, or Git in short. It support two basic operations:

1. Append a new commit to current version (thus made a new version)
2. Revert the state of the project of the i-th day.

As GSS just started a brand-new project, it is day 0 now. Every other day, GSS would upload his work (one of the two operations above) to the repository of the project.

The version control system would calculate the hash of the whole project. And the checksum of the project is the bitwise exclusive or (xor) of all the checksum of commits which is valid (not reverted) at one version.

Your task is to calculate the hash of the whole project after GSS's commit.

Input

The first line of input is a single number n, the number of days that GSS works on the project.

Then n line follows, the i-th line consists of a string and an integer in the format below:

1. `commit x`, append a new commit to the current version, and x is the checksum of the commit, $0 \leq x \leq 10^9$.
2. `checkout k`, revert the state of the project to the k-th day, $0 \leq k < i$.

Output

For each operation, output the hash of the project after the operation.

Examples

standard input	standard output
7 commit 1 commit 2 checkout 1 checkout 0 commit 4 checkout 2 checkout 5	1 3 1 0 4 3 4

Problem E. Jump A Jump

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	512 mebibytes

There's a very popular game called jump a jump recently.In order to get good marks, many people spend a lot of time in this game.Wei is the master of jump a jump,he can easily get a very high score every time.

Recently he discovered that because of his muscle memory,he could not fully control his game score.

Now he wants you to help him figure out the nearest score that he can get from his desired score.

Every time he jumps, he chooses one from m distances,he can jump until he wants to end,and he can also use every distance any times.

Input

The first line has a number $n(0 < n \leq 10^{15})$ to indicate the score he desired.

The next line has a number $m(0 < m \leq 2000)$, which indicates how many choices each step has.

The last line has m integers $b_i(0 < b_i \leq 5000)$ representing the ith distances.

Output

Output is the absolute value of the difference between the nearest score he can get and his desired score.

Examples

standard input	standard output
11 3 5 7 8	1

Note

In the example,

He can jump $5 + 7 = 12, 12 - 11 = 1$.

He can also jump distance 5 for two times, and $2 \times 5 = 10, 11 - 10 = 1$.

Problem F. A-maze-ing

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	512 mebibytes

Long long age, a wise but barbaric king made a plan to convert the arena to a maze so that he could get pleasure from watching his servant running the maze confusedly.

The structure of the arena could be abstracted into a directed connected graph comprised of n ($1 \leq n \leq 10^5$) nodes and m ($1 \leq m \leq 2 \times 10^5$) edges.

The king had not decided where to set up the starting nodes and the end node.

So the king proposed a requirement.

Whichever two points u, v he chose, there existed a path from one point to the other(a path from u to v or from v to u).

Moreover, the king hoped to improve the difficulty of the game, so the number of nodes in the maze should be as large as possible.

You are only required to output the maximum number of nodes in the maze.

Input

There are two positive integers n and m in the first line.

And then m lines follow, in each line there are two positive integers u and v, describing that an edge from node u to node v.

Output

Just output one integer, the maximum size of the maze.

Examples

standard input	standard output
3 2 1 2 1 3	2

Problem G. Pretty Derby

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	512 mebibytes

This is an interactive problem.

There are exactly 25 little ponies, and they want to know who is the fastest and who is the first runner-up (or the second place), so they asked you for help.

You have a field which can hold a race with at most 5 ponies, and you do not have a stopwatch, so you only know who is the the first, second, third, forth and last in a run. The ponies' speed is so stable that no matter how many runs you make, their rank won't change.

Please help them with the minimum number of runs!

Interaction Protocol

Let's number the ponies from 1 to 25. For each run, you should output five numbers x_1, x_2, x_3, x_4, x_5 in a line separated by spaces, means that you held a race with these five ponies.

Then, you can read a permutation of 1 to 5, the rank of the five ponies in the race.

After you can determine which two ponies runs fastest, output two number, the first and second fastest pony among the 25 ponies, and three zeroes in a line, like `1 2 0 0 0` if you thought that the pony number 1 is the fastest and pony number 2 is the second fastest.

Remember to flush `stdout` after each line of your output!

Examples

standard input	standard output
1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5	1 2 3 4 5 1 2 6 7 8 1 2 9 10 11 1 2 12 13 14 1 2 15 16 17 1 2 18 19 20 1 2 21 22 23 1 2 23 24 25 1 2 0 0 0

Note

The example above is just a example interaction, you should not take it seriously.

When you write the solution for the interactive problem it is important to keep in mind that if you output some data it is possible that this data is first placed to some internal buffer and may be not directly transferred to the interactor. In order to avoid such situation you have to use special flush operation each time you output some data. There are these flush operations in standard libraries of almost all languages. For example, in `C++` you may use `fflush(stdout)` or `cout < < flush` (it depends on what do you use for output data —`scanf/printf` or `cout`). In Java you can use method flush for output stream, for example, `System.out.flush()`. In Python you can use `stdout.flush()`.

Input/output in interactive problems works much slower than in usual problems —try to use `scanf/printf` instead of `cin/cout` in `C++`, `BufferedReader/PrintWriter` in Java and etc.

Problem H. GSS and OJ Submissions

Input file:	standard input
Output file:	standard output
Time limit:	4 seconds
Memory limit:	512 mebibytes

GSS is holding a large programming contest on the website called EOJ or compileError Online Judge which is the largest programming contest platform in the world. Every day, millians of codes are submitted to the platform.

One day, someone submitted the n-th code ($1 \leq n \leq 4 \times 10^8$), where n is GSS's lucky number, to celebrate this day, GSS is going to send a huge prize. In this case, he allocated every submission a number in $[0, 2^{64})$ (see the paragraph below), then he generate another number L in $[1, n]$. The user who submitted the submission with the L-th small number will receive the prize.

GSS allocate numbers to submissions in the following way:

```
typedef unsigned long long ull;
void allocate(ull A, ull B, ull s0, ull s[])
{
    s[0] = s0;
    for (int i = 1; i < n; i++) {
        s[i] = s[i - 1] * A + B;
    }
}
```

`A`, `B` and `s0` are generated elsewhere, and $0 \leq A, B, s_0 < 2^{64}$. And you can assume that A, B and s0 are generated in random.

Special notice: the code above will comsume about 2 seconds or more on judger.

Now, Mingming have collected all numbers allocated to his submissions, and he wants to know weather he will win the prize. But he is too lazy to solve, so he asked you for help, please, tell him the number allocated to the submission which win the prize.

Input

Input contains 5 integers, $A, B, L, n, s0$.

Output

Output one line with the number —— the number allocated to the submission which win the prize.

Examples

standard input	standard output
5 7 9 11 13	5761717

Note

The numbers allocated to submissions are:

13, 72, 367, 1842, 9217, 46092, 230467, 1152342, 5761717, 28808592, 144042967

it is clear that the ninth one is 5761717.