

2022 SBU ICPC Selection Contest

December 2, 2022

About

Welcome to the 2022 SBU ICPC Selection Contest! The goal of the contest is to identify the top candidates for our International Collegiate Programming Contest(ICPC) teams, which will participate in the Greater New York Regional Contest for a slot in the ICPC World Finals 2023.

Rules

This contest is a local ICPC-style contest. Each participant must compete alone (this is not a team contest). The contest lasts for 3 hours (7:20PM - 10:20PM). All submissions read from stdin and output to stdout. For more details regarding the execution environment, please refer to this page.

Scoring

- Participants are ranked according to the most problems solved.
- Participants who solve the same number of problems are ranked by least total time.
- The total time is the sum of the time consumed for each problem **solved**. The time consumed for a solved problem is the time elapsed from the beginning of the contest to the submittal of the accepted run plus 20 penalty minutes for every rejected run for that problem regardless of submittal time. There is no time consumed for a problem that is not solved.

Note

There are 8 problems to be solved in 3 hours. The **first problem is the easiest** one. The rest are randomly ordered.

A - War



Tom and Katy are playing their own version of *war*, the card game. Tom and Katy each have N cards with values that range from 1 to 9. They will flip the top card of their respective deck and the person with the higher value wins the round. When they tie, the outcome of the round is a draw. Whoever wins the most rounds when they run out of cards is the winner. Given each person's deck, who will win the game of *war*?

Input Format

The first line will contain the integer N , representing the number of cards in each deck. It will be followed by N lines each containing two integers representing the players' cards, separated by a space. The first number is Tom's card while the second is Katy's.

Constraints

$$1 \leq N \leq 100$$

$$1 \leq \text{card values} \leq 9$$

Output Format

If Tom wins print “Tom wins!”. If Katy wins print “Katy wins!”. If both of them have the same number of wins, print “Draw!”

Sample Input 0

```
3
1 9
2 3
5 4
```

Sample Output 0

Katy wins!

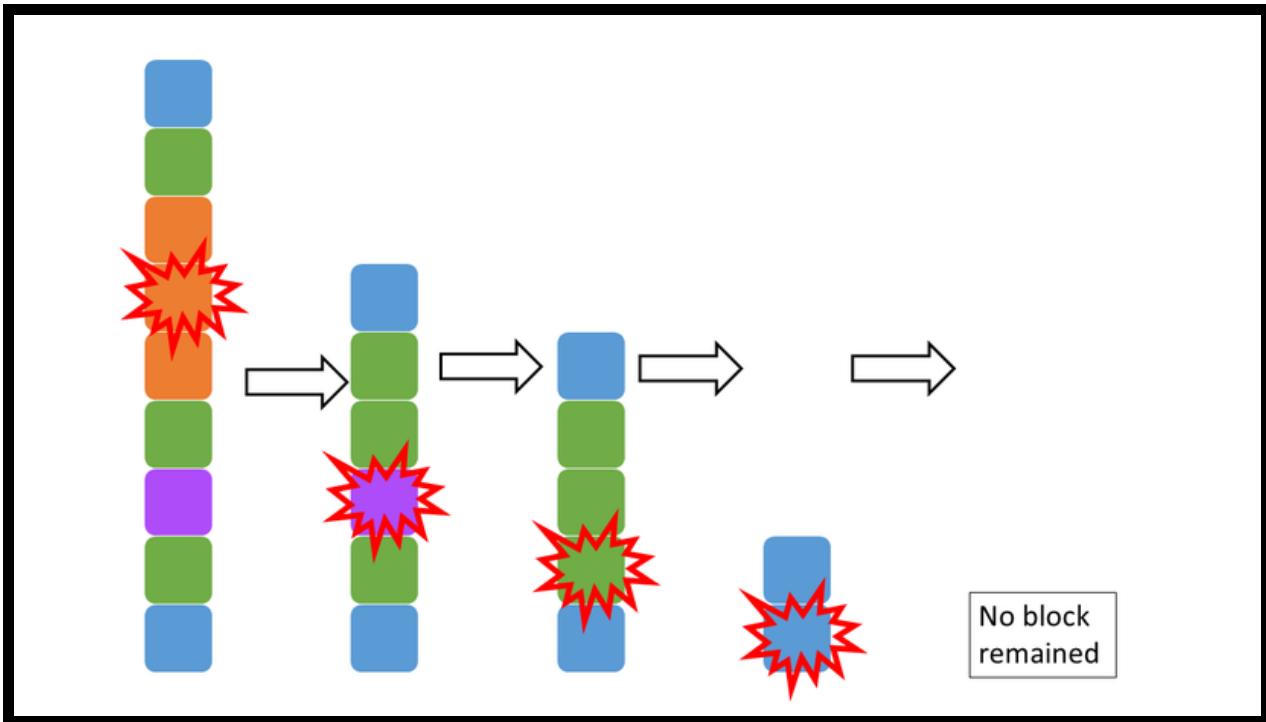
Sample Input 1

3
9 9
7 3
5 6

Sample Output 1

Draw!

B - Happy Elimination



Amy is playing a mobile game called *Happy Eliminating*. This game gives certain blocks in a vertical line at the beginning. Amy can tap on a block and eliminate all neighbors of the same color. An elimination of m blocks will give Amy m^2 points. After an elimination move, other blocks will drop so there is no space between the blocks. Please help Amy determine the maximum points she can earn for each game.

Input Format

The first line is an integer n indicating the number of blocks. The second line contains n integers c_1, c_2, \dots, c_n separated by space. c_i indicates the color of the i -th block.

Constraints

$$\begin{aligned}1 &\leq n \leq 100 \\1 &\leq c_i \leq 100\end{aligned}$$

Output Format

Print the maximum points Amy can get.

Sample Input 0

9
1 3 2 2 2 3 4 3 1

Sample Output 0

23

C - Purrfection



Toby is a fluffy cat with n toys! While he loves playing with his toys, he also loves doing math! To measure how much he likes each of his toys, he assigned each a purrfection value p_i . While playing with all of his toys, Toby discovered a way to measure the total purrfection T of his toys:

$$T = \left(\prod_{i=1}^n \prod_{j=i+1}^n |p_i - p_j| \right) \bmod m$$

Given the purrfection value of each of Toby's toys, compute the total purrfection, T . As T may be very large, output it modulo m .

Input Format

The first line contains two integers n, m - the number of toys and the modulo. The second line contains n integers p_1, p_2, \dots, p_n .

Constraints

$$\begin{aligned} 2 \leq n &\leq 2 \times 10^5 \\ 1 \leq m &\leq 1000 \\ 0 \leq p_i &\leq 10^9 \end{aligned}$$

Output Format

Output a single integer, the total purrfection modulo m .

Sample Input 0

2 10
5 8

Sample Output 0

3

Explanation 0

$$|5 - 8| \bmod 10 = 3 \bmod 10 = 3$$

Sample Input 1

3 12
1 4 5

Sample Output 1

0

Explanation 1

$$(|1 - 4| \cdot |1 - 5| \cdot |4 - 5|) \bmod 12 = 0$$

D - Meow You Doin?



Molly (gray), Toby (orange), and Nikki (white) are all napping at home. Their home has n rooms and some pairs of rooms are connected by two-way hallways. Two rooms can be directly connected by at most one hallway. It is guaranteed that you can get from any room to any other by moving through hallways. The rooms are numbered from 1 to n . Molly is napping in room a and Toby is napping in room b . Molly and Toby are not napping in the same room. Nikki was napping but she was interrupted by her hunger. She decided to get up and go get some of that sweet Meow Mix. On her way, she wonders if she will pass through the rooms in which Molly and Toby are napping. As Nikki is a meowthematician, she would like to know the number of pairs of rooms (x, y) ($x, y \notin \{a, b\}$) such that if she goes from room x to room y she will have to pass through both rooms a and b .

The order of two rooms in a pair does not matter, that is, the pairs x, y and y, x must be taken into account only once.

Input Format

The first line contains four integers n, m, a, b - the number of rooms, the number of hallways, the room where Molly is napping, and the room where Toby is napping. The next m lines contain descriptions of hallways between rooms. Each hallway description contains a pair of integers u_i, v_i - the room numbers connected by the hallway.

Constraints

$$4 \leq n \leq 10^5$$

$$n - 1 \leq m \leq \min \left\{ \frac{n(n-1)}{2}, 10^5 \right\}$$

$$1 \leq a, b \leq n, a \neq b$$

$$1 \leq u_i, v_i \leq n, u_i \neq v_i$$

Output Format

Output a single integer, the number of pairs of rooms x, y such that if Nikki goes from room x to room y she will have to pass through both rooms a and b .

Sample Input 0

5 4 2 3

1 2

2 3

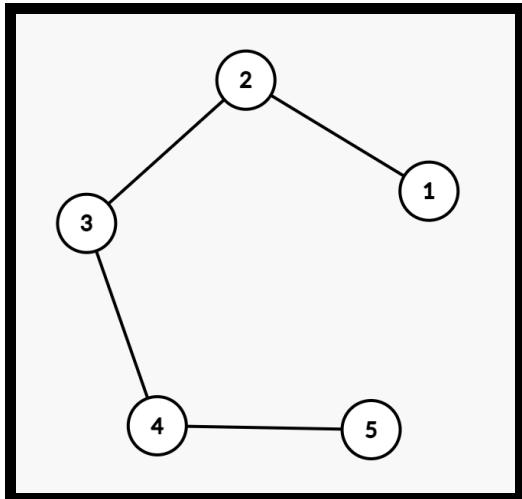
3 4

4 5

Sample Output 0

2

Explanation 0



Here, (1, 4) and (1, 5) are the only pairs of rooms that are connected by paths all of which pass through both rooms 2 and 3.

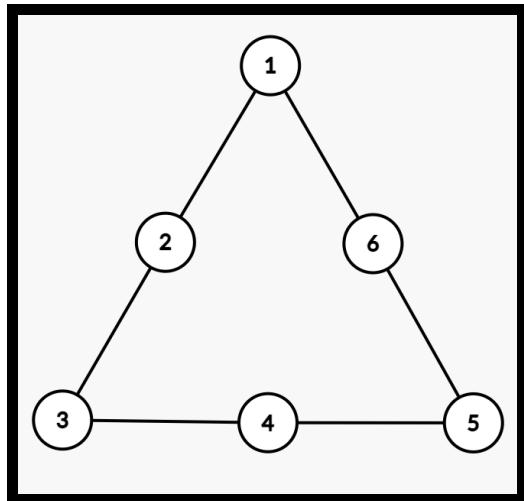
Sample Input 1

```
6 6 1 4  
1 2  
2 3  
3 4  
4 5  
5 6  
6 1
```

Sample Output 1

```
0
```

Explanation 1



There exists no pair of rooms (x, y) such that all paths between x and y pass through both rooms 1 and 4.

E - Dice Auction

This is a two-player game.

Both you and your opponent roll a die. Each of you only know your own roll. A third party looks at both dice and puts an amount of money equal to the sum of the two dice rolls in the middle, hidden from sight. You will now participate in an auction with your opponent on the money pot. The player to bid first is selected randomly. Bidding starts at 1, and, when it's your turn, you have the option of either raising by 1 or passing, as does your opponent on their turn. Whenever somebody passes, the other person must pay money equal to their current bid for the pot. Depending on the true value of the pot, this could result in a net gain or loss of money.

To give you an advantage, you will roll a $d6$ (i.e. a standard six sided die), while your opponent will roll a $d4$ (i.e. a standard four sided die).

This game will be played 1000 times with different randomly generated rolls. Both your and your opponent's goal is to make more net gain of money than the other person by the end of all of the rounds.

Input Format

The input consists of 1000 lines, each containing a natural number between 1 and 6, representing your roll in a given round of the game.

Constraints

None!

Output Format

The output must consist of 1000 lines, each with a natural number between 1 and 10 (it does not make sense to bid more than 10), representing the maximum value to which you are willing to bid in that given round of the auction. Note that, as an example, even if you output 7, it is possible that your opponent decides to stop bidding after you had bid 4. In this example, you get the money pot for only 4\$.

The problem will be judged against a custom judge that will play your strategy against a particular one hard-coded for your opponent. You will pass the problem if and only if your net earnings are greater than your opponent's. It is guaranteed that the opponent has no access to your outputs and it is always possible to beat the opponent for the set of test cases (rolls for both you and the opponent) we have randomly generated.

F - Stop the overflow



Given the description of a water jar and a water tap find the time required to fillup the jar with water before it overflows. You can ignore the surface tension and assume 1 unit of water can flow through each unit square of the tap opening area in a single second.

Input Format

A single line will contain three non-negative single-space seperated floating point numbers r, h, R representing the radius and the height of the water jar followed by the radius of the water tap opening.

Constraints

$$1 \leq r, h \leq 1000$$

$$1 \leq R \leq 1000$$

Output Format

Output a single number (rounded to the third decimal) specifying the number of seconds it will take to fill up the water jar.

Sample Input 0

3.0 4.0 2.0

Sample Output 0

9.000

Sample Input 0

5.0 4.0 3.0

Sample Output 0

11.111

G - String Problem

Given a string S , you are asked to find the number of good ways to split the string. A good way to split a string S must satisfy the following constraints:

- $S = (AB)^i C$ for some $i \geq 1$
- $F(A) \leq F(C)$
- A, B, C are all non-empty strings.

If $A = "xy"$ and $B = "we"$, then $AB = "xywe"$.

We recursively define A^n as follows: $A^n = A (A^{n-1})$. For example, $A = A^1 = "xy"$, $A^2 = "xyxy"$, and so on.

$F(A)$ is defined to be the number of characters in the alphabet who have an odd number of occurrences in string A . For example $F(abcwwwb) = 3$ since a, c , and w have odd number of occurrences in A .

Two *good* ways are considered different iff any of A, B, C is different in the two ways.

Input Format

The first line of the input consists of the number of test cases.

Each of the following lines includes one test case, which indicates the string S .

Constraints

The number of test cases is at most 5. The length of string S does not exceed 1.1×10^6 , and includes only lowercase letters.

Output Format

The output includes one line for each test case, which indicates the number of good ways to split the input string S .

Sample Input 0

```
3
qqpqqp
yyyrrz
yyfyyfq
```

Sample Output 0

```
8
9
16
```

Explanation 0

Explanation for "qpqqp":

1. $A = q, \quad B = qp, \quad C = qqp$
2. $A = q, \quad B = qpq, \quad C = qp$
3. $A = q, \quad B = qpqq, \quad C = p$
4. $A = qq, \quad B = p, \quad C = qqp$
5. $A = qq, \quad B = pq, \quad C = qp$
6. $A = qq, \quad B = pqq, \quad C = p$
7. $A = qqp, \quad B = q, \quad C = qp$
8. $A = qqp, \quad B = qq, \quad C = p$

Explanation for "yyrrz":

1. $A = y, \quad B = y, \quad C = yrrz$
2. $A = y, \quad B = yy, \quad C = rrz$
3. $A = y, \quad B = yyr, \quad C = rz$
4. $A = y, \quad B = yyrr, \quad C = z$
5. $A = yy, \quad B = y, \quad C = rrz$
6. $A = yy, \quad B = yr, \quad C = rz$
7. $A = yy, \quad B = yrb, \quad C = z$
8. $A = yyy, \quad B = r, \quad C = rz$
9. $A = yyy, \quad B = rr, \quad C = z$

H - RothLand



One fine morning, the queen of the ducks of Rothland decides that it is time to conquer NCS, the New Computer Science building. The queen's advisor informs her that roughly N human beings are estimated to be at NCS on any day. The queen declares that they will invade NCS the day the population of Rothland is at least N . Currently there are D ducks in Rothland. So they start asking ducks from nearby ponds to come join them. The news starts spreading like wildfire.

The queen estimates that every day the population of Rothland is increasing by P percent. For example, if there are currently 20 ducks in Rothland and if their number increases by 11% every day, there will be 22.2 ducks after one day, 24.642 ducks after two days, and so on. If there were 23 humans in NCS, then they would have to wait two days before invading NCS. [Yes, it is very difficult to visualize 22.2 ducks, but what can we do? We are estimating after all.]

Given the number of ducks D , the number of humans in NCS N , and the percentage increase of ducks every day P , find the number of days before NCS is taken over by ducks.

Input Format

The first line contains a positive integer T , number of test cases. In each of the following T lines, there will be 3 integers, D , P , and N .

Constraints

$$1 \leq T \leq 250000$$

$$1 \leq D \leq 10^9$$

$$1 \leq P \leq 100$$

$$1 \leq N \leq 10^{18}$$

Output Format

Print the case number and then an integer signifying the number of days after which NCS will be invaded. Please see the sample output for exact output format.

Sample Input 0

```
3
20 11 22
20 11 24
230948879 7 123456789012345678
```

Sample Output 0

```
Case 1: 1
Case 2: 2
Case 3: 298
```