Noctem Virtual II
Division I

1. You have three hours to complete this contest. There are 6 problems, and you may complete them in any order.

2. Cheating is strictly prohibited. This includes conferring with people that are not on your team or using code written by someone not on your team. Failure to abide by this rule will result in automatic disqualification.

3. As you complete the questions, you should submit your solutions on the grader website. There, you will be able to see how many test cases you got correct for the given question. You may resubmit on the grader as many times as you would like during the three hour period for no penalty. However, only the last submission you make on the grader for a question will be used to calculate your final score.

4. Your final score will be calculated based off of your last submission on each question. Each test case will be worth 1 point, for a total of 10 points per question and 60 points in total. If there is a tie between two teams, the tie will be broken by looking at which team had more of the three hour time window remaining when they submitted for the last time.

5. Last but not least, we wish your team the best of luck!

Problem 1 - Evacuation

Problem:
A rogue monkey has come to a hotel to wreak havoc. The hotel has N rooms numbered 1 through N, connected by N - 1 walkways such that each room can reach every other room through a series of walkways. M people are all currently meeting in room 1 when they find out monster monkey has arrived. Since everyone is all packed in one room, this makes it very easy for monkey to eat everyone, so you will have to distribute the people to different rooms. Starting at room 1, you can split the people into at most K groups, and send each group off to a room connected to room 1. You may also choose to leave any amount of people to stay in room 1. Then, you do the same at each room you send a group off to, and then do the same on the next group, until you are done distributing. After distributing the people, the monkey will choose the room with most people and eat them all. Since you want to minimize the amount of people the monkey eats, minimize the maximum number of people among all rooms by distributing the people optimally.

Input Format (evacuation.in):
The first line contains N ($1 \leq N \leq 10^5$), M ($1 \leq M \leq 10^9$), and K ($1 \leq K \leq 10^5$). The next N - 1 lines each contains two numbers, indicating that there is a walkway between those two rooms.

Output Format (evacuation.out):
The minimum number of people that will be eaten by monkey.

Sample Input:
```
7 20 2
1 2
1 3
1 4
2 5
2 6
4 7
```

Sample Output:
```
4
```

Credits: Steven Tan

Problem 2 - Cleaning Windows

Problem:
Jerome is a window cleaner. Today, he is cleaning a N story building with a single window on each story. The window on the i-th floor has dirtiness D_i. He starts at floor 1, and can move up or down in one unit of time, as well as lower the dirtiness of one window in a unit time by 1. Given that Jerome only has K units of time to do his job, find the maximum number of windows he can clean (a window is clean if its dirtiness is 0). Note that Jerome cannot move below floor 1 or above floor N.

Input Format (windows.in):
The first line contains N ($1 \leq N \leq 10^5$) and K ($1 \leq K \leq 10^9$). The i-th line of the next N lines each contain an integer D_i ($1 \leq D_i \leq 10^9$).

Output Format (windows.out):
The maximum number of windows Jerome can clean.

Sample Input:
5 15
2
5
8
3
1

Sample Output:
4

Credits: Marco Frigo

Problem 3 - Word Game

Problem:
Given a string S, and target string K, two players play a game. The two players will maintain an interval [L, R] on the string. Player one starts by picking any character from the string, and both L and R will be set to index of that character. Then starting with player 2, they will take turns extending the interval one to the left or one to the right ([L, R] becomes [L - 1, R] or [L, R + 1]), as long as the new L and R are still within string S. The game ends when the substring [L, R] becomes the target string, and the winner is the person who last moved. Note that if the target string is never reached and players run out of moves, it is a tie. If both players play optimally, determine who will win, or that it will be a draw.

Input Format (wordgame.in):
The first line contains T ($1 \leq T \leq 10^4$), the number of test cases. Each test case consists of two lines, the first line containing string S ($1 \leq |S| \leq 10^6$), and the second line containing string K ($1 \leq |K| \leq |S|$).

It is guaranteed that the sum of $|S|$ over all test cases does not exceed $2 * 10^6$.

Output Format (wordgame.out):
For each test case, output a single integer on a new line. Output 1 if player 1 wins, 2 if player 2 wins, and 0 if its a draw.

Sample Input:
```
3
cfzaabe
ab
bcacacd
cac
abababab
abab
```

Sample Output:
```
0
1
2
```

Credits: Steven Tan

Problem 4 - Colored Edges

Problem:
You are given a graph with N vertices and M bidirectional edges. Each edge is either colored black or white. You can only traverse an edge if it is white, not if it is black. In one move you can pick a node and turn all of the edges connecting to it the opposite color. At the start, their may not be a walkable path from node 1 to node N. Determine the minimum number of moves to make it such that there exists a path from node 1 to N that is all white, and therefore walkable. Note that you must first perform all the toggles the graph, and then you can start walking. You cannot traverse the graph while toggling.

Input Format (colored.in):
The first line N $(1 \leq N \leq 1000)$ and M $(1 \leq M \leq 5000)$. Each of the next M lines describes an edge and contains 3 integers, the first two specifying the nodes which it connects, and the third specifying the color (0 for black, 1 for white).

Output Format (colored.out):
The minimum number of operations that must be performed in order to create a clear path from node 1 to N.

Sample Input:
5 6
1 2 1
1 3 0
3 4 0
2 4 0
2 5 0
3 5 1

Sample Output:
1

Credits: Steven Tan

Problem 5 - Banana Factory

Problem:
Chieftain Monkey has set up a banana factory. He wishes to deliver only the highest quality bananas to his monkey republic. However, his two conveyor belts of length M that transport bananas do not live up to that standard. They will collapse if a certain weight K is exceeded. Once one of the belts collapse, neither will be able to transport more bananas. After each second, both conveyor belts will slide 1 unit. Anuj has a stack of N bananas each with weight W_i. Anuj must place the top banana on a conveyer belt every second, or else Cheiftain Monkey will be mad. Help Anuj find the maximum number of bananas to leave the conveyor belts before either breaks.

Input Format (factory.in):
The first line contains 3 integers, N ($1 \leq$ N ≤ 200), M ($1 \leq$ M ≤ 15), and K ($1 \leq$ K $\leq 10^9$). The next N lines contains $W_i (1 \leq W_i \leq 10^9)$ the weight of each banana. Consider the first weight given to be the top of the stack.

Output Format (factory.out):
A single integer signifying the maximum number of bananas to leave the conveyor belts.

Sample Input:
5 3 10
3
4
5
9
6

Sample Output:
2

Credits: Marco Frigo

Problem 6 - Phone

Problem:
Jerome has a trash phone, and he can barely type. He starts with a sequence of digits A, but he has not typed it correctly, and what he actually wants is a sequence of digits B of the same length. In an attempt to correct his sequence, he can perform some operations on A. In one operation, Jerome randomly selects an index i such that A_i doesn't equal B_i. Since the phone is bad, Jerome can't directly change A_i to B_i. Instead, the operation will transform A_i to one of the next K larger digits, including A_i itself, all with equal probability (if any digit exceeds 9, it will wrap back to 0). In other words, A_i will turn into one of $\{A_i, (A_i + 1) \mod 10, ..., (A_i + k - 1) \mod 10\}$, each with equal probability. If Jerome keeps on applying this operation, what is the expected number of operations he needs before sequence A becomes sequence B?

Input Format (phone.in):
The first line contains $K (2 \leq K \leq 10)$. The second line contains $A (1 \leq |A| \leq 10^5)$. The third line contains B, which is the same length as A.

Output Format (phone.out):
The expected number of operations before the sequence A becomes B, modulo 1000000007.

Explanation of modulo: In all cases, the expected value can be expressed in the form of an irreducible fraction C/D. Output the value $E (0 \leq E < 1000000007)$ such that $E \cdot D = C \pmod{1000000007}$.

Sample Input:
3
012
456

Sample Output:
395894457

Sample Output Explanation:
The expected number of operations in this case is $\frac{8892}{341}$, which is 395894457 under mod 1000000007.

Credits: Steven Tan