



Prisoner Challenge

In a prison, there are 500 prisoners. One day, the warden offers them a chance to free themselves. He places two bags with money, bag A and bag B, in a room. Each bag contains between 1 and N coins, inclusive. The number of coins in bag A is **different** from the number of coins in bag B. The warden presents the prisoners with a challenge. The goal of the prisoners is to identify the bag with fewer coins.

The room, in addition to the money bags, also contains a whiteboard. A single number must be written on the whiteboard at any time. Initially, the number on the whiteboard is 0.

Then, the warden asks the prisoners to enter the room, one by one. The prisoner who enters the room does not know which or how many other prisoners have entered the room before them. Every time a prisoner enters the room, they read the number currently written on the whiteboard. After reading the number, they must choose either bag A or bag B. The prisoner then **inspects** the chosen bag, thus getting to know the number of coins inside it. Then, the prisoner must perform either of the following two **actions**:

- Overwrite the number on the whiteboard with a non-negative integer and leave the room. Note that they can either change or keep the current number. The challenge continues after that (unless all 500 prisoners have already entered the room).
- Identify one bag as the one that has fewer coins. This immediately ends the challenge.

The warden will not ask a prisoner who has left the room to enter the room again.

The prisoners win the challenge if one of them correctly identifies the bag with fewer coins. They lose if any of them identifies the bag incorrectly, or all 500 of them have entered the room and not attempted to identify the bag with fewer coins.

Before the challenge starts, the prisoners gather in the prison hall and decide on a common **strategy** for the challenge in three steps.

- They pick a non-negative integer x , which is the largest number they may ever want to write on the whiteboard.
- They decide, for any number i written on the whiteboard ($0 \leq i \leq x$), which bag should be inspected by a prisoner who reads number i from the whiteboard upon entering the room.
- They decide what action a prisoner in the room should perform after getting to know the number of coins in the chosen bag. Specifically, for any number i written on the whiteboard

($0 \leq i \leq x$) and any number of coins j seen in the inspected bag ($1 \leq j \leq N$), they either decide

- what number between 0 and x (inclusive) should be written on the whiteboard, or
- which bag should be identified as the one with fewer coins.

Upon winning the challenge, the warden will release the prisoners after serving x more days.

Your task is to devise a strategy for the prisoners that would ensure they win the challenge (regardless of the number of coins in bag A and bag B). The score of your solution depends on the value of x (see Subtasks section for details).

Implementation Details

You should implement the following procedure:

```
int[][] devise_strategy(int N)
```

- N : the maximum possible number of coins in each bag.
- This procedure should return an array s of arrays of $N + 1$ integers, representing your strategy. The value of x is the length of array s minus one. For each i such that $0 \leq i \leq x$, the array $s[i]$ represents what a prisoner should do if they read number i from the whiteboard upon entering the room:
 1. The value of $s[i][0]$ is 0 if the prisoner should inspect bag A, or 1 if the prisoner should inspect bag B.
 2. Let j be the number of coins seen in the chosen bag. The prisoner should then perform the following action:
 - If the value of $s[i][j]$ is -1 , the prisoner should identify bag A as the one with fewer coins.
 - If the value of $s[i][j]$ is -2 , the prisoner should identify bag B as the one with fewer coins.
 - If the value of $s[i][j]$ is a non-negative number, the prisoner should write that number on the whiteboard. Note that $s[i][j]$ must be at most x .
- This procedure is called exactly once.

Example

Consider the following call:

```
devise_strategy(3)
```

Let v denote the number the prisoner reads from the whiteboard upon entering the room. One of the correct strategies is as follows:

- If $v = 0$ (including the initial number), inspect bag A.

- If it contains 1 coin, identify bag A as the one with fewer coins.
- If it contains 3 coins, identify bag B as the one with fewer coins.
- If it contains 2 coins, write 1 on the whiteboard (overwriting 0).
- If $v = 1$, inspect bag B.
 - If it contains 1 coin, identify bag B as the one with fewer coins.
 - If it contains 3 coins, identify bag A as the one with fewer coins.
 - If it contains 2 coins, write 0 on the whiteboard (overwriting 1). Note that this case can never happen, as we can conclude that both bags contain 2 coins, which is not allowed.

To report this strategy the procedure should return $[[0, -1, 1, -2], [1, -2, 0, -1]]$. The length of the returned array is 2, so for this return value the value of x is $2 - 1 = 1$.

Constraints

- $2 \leq N \leq 5000$

Subtasks

1. (5 points) $N \leq 500$, the value of x must not be more than 500.
2. (5 points) $N \leq 500$, the value of x must not be more than 70.
3. (90 points) The value of x must not be more than 60.

If in any of the test cases, the array returned by `devise_strategy` does not represent a correct strategy, the score of your solution for that subtask will be 0.

In subtask 3 you can obtain a partial score. Let m be the maximum value of x for the returned arrays over all test cases in this subtask. Your score for this subtask is calculated according to the following table:

Condition	Points
$40 \leq m \leq 60$	20
$26 \leq m \leq 39$	$25 + 1.5 \times (40 - m)$
$m = 25$	50
$m = 24$	55
$m = 23$	62
$m = 22$	70
$m = 21$	80
$m \leq 20$	90

Sample Grader

The sample grader reads the input in the following format:

- line 1: N
- line $2 + k$ ($0 \leq k$): $A[k] B[k]$
- last line: -1

Each line except the first and the last one represents a scenario. We refer to the scenario described in line $2 + k$ as scenario k . In scenario k bag A contains $A[k]$ coins and bag B contains $B[k]$ coins.

The sample grader first calls `devise_strategy(N)`. The value of x is the length of the array returned by the call minus one. Then, if the sample grader detects that the array returned by `devise_strategy` does not conform to the constraints described in Implementation Details, it prints one of the following error messages and exits:

- `s` is an empty array: `s` is an empty array (which does not represent a valid strategy).
- `s[i]` contains incorrect length: There exists an index i ($0 \leq i \leq x$) such that the length of `s[i]` is not $N + 1$.
- First element of `s[i]` is non-binary: There exists an index i ($0 \leq i \leq x$) such that `s[i][0]` is neither 0 nor 1.
- `s[i][j]` contains incorrect value: There exist indices i, j ($0 \leq i \leq x, 1 \leq j \leq N$) such that `s[i][j]` is not between -2 and x .

Otherwise, the sample grader produces two outputs.

First, the sample grader prints the output of your strategy in the following format:

- line $1 + k$ ($0 \leq k$): output of your strategy for scenario k . If applying the strategy leads to a prisoner identifying bag A as the one with fewer coins, then the output is the character A. If applying the strategy leads to a prisoner identifying bag B as the one with fewer coins, then the output is the character B. If applying the strategy does not lead to any prisoner identifying a bag with fewer coins, then the output is the character X.

Second, the sample grader writes a file `log.txt` in the current directory in the following format:

- line $1 + k$ ($0 \leq k$): $w[k][0] w[k][1] \dots$

The sequence on line $1 + k$ corresponds to scenario k and describes the numbers written on the whiteboard. Specifically, $w[k][l]$ is the number written by the $(l + 1)^{th}$ prisoner to enter the room.