Broken Device 2

Anna and Bruno are gamble masters. They will play a game with D-taro who is the dealer of the game.

In this game, Anna and Bruno stay in different rooms. They can communicate with each other using a broken device only. D-taro gives an integer to Anna. For Anna and Bruno, the purpose of this game is to send the given integer from Anna to Bruno using the device.

When the game starts, in the beginning, Anna declares an integer m between 1 and 2 000, inclusive. Then they play Q rounds. The round i ($1 \leq i \leq Q$) is performed as follows.

1. D-taro gives an integer A_i to Anna.
2. Anna inputs arrays s_i, t_i into the device. Every element of the arrays s_i, t_i should be either 0 or 1. The arrays s_i, t_i should have the same length, and the length is between 1 and m, inclusive.
3. Let u_i be an array obtained from the arrays s_i and t_i by **riffle shuffle** (see below). Then the device sends u_i to Bruno.
4. Bruno sends an integer to D-taro. If this integer is the same as the integer A_i given by D-taro to Anna, Anna and Bruno win for this round.

Write programs which implements the strategies of Anna and Bruno so that theory win for all the Q rounds.

Riffle Shuffle

We say an array Z is obtained from the arrays X and Y by riffle shuffle if we can divide the elements of the array Z into two groups so that the following two conditions are satisfied.

- The array consisting of the elements in the first group in the same order is equal to the array X.
- The array consisting of the elements in the second group in the same order is equal to the array Y.

For example, the array $Z = [1, 1, 1, 0, 0, 0]$ is obtained from $X = [1, 1, 0]$ and $Y = [1, 0, 0]$ by riffle shuffle because the array consisting of the first, second, fourth elements of Z in the same order is the array $[1, 1, 0]$, and the array consisting of the third, fifth, sixth elements of Z in the same order is the array $[1, 0, 0]$.

On the other hand, if $X = [1, 1, 0]$, $Y = [1, 0, 0]$, and $Z = [0, 0, 0, 1, 1, 1]$, the array Z is not obtained from X and Y by riffle shuffle.

Implementation Details

You need to submit two files.

The first file is `Anna.cpp`. It should implement Anna's strategy. It should implement the following functions. The program should include `Anna.h` using the preprocessing directive `#include`.

- `int Declare()`
  This function is called once in the beginning.
  - The return value is the integer m declared by Anna.
  - The integer m should be between 1 and 2 000, inclusive. If this condition is not satisfied, your program is judged as **Wrong Answer [1]**.
- `std::pair<std::vector<int>, std::vector<int> > Anna(long long A)`
  After the function `Declare` is called, this function is called Q times. The i-th call ($1 \leq i \leq Q$) to this function corresponds to Step 1 and Step 2 for the round i.
  - The parameter `A` is the integer A_i given by D-taro to Anna.
  - The return value is the pair of the two arrays s_i, t_i input by Anna into the device.
  - Every element of the arrays s_i, t_i should be either 0 or 1. If this condition is not satisfied, your program is judged as **Wrong Answer [2]**.
  - Both of the lengths of the arrays s_i, t_i should be between 1 and m, inclusive. If this condition is not satisfied, your program is judged as **Wrong Answer [3]**.
  - The arrays s_i, t_i should have the same length. If this condition is not satisfied, your program is judged as **Wrong Answer [4]**.

The second file is `Bruno.cpp`. It should implement Bruno's strategy. It should implement the following function. The program should include `Bruno.h` using the preprocessing directive `#include`.

- `long long Bruno(std::vector<int> u)`
  After Anna inputs the arrays into the device, this function is called once. In total, this function is called Q times. The i-th call ($1 \leq i \leq Q$) to this function corresponds to Step 3 and Step 4 for the round i.
  - The parameter `u` is the array u_i sent by the device to Bruno for the round i.
  - The return value is the integer sent by Bruno to D-taro.
  - The return value should be the same as the integer A_i given by D-taro to Anna. If this condition is not satisfied, your program is judged as **Wrong Answer [5]**.

Important Notices

- Your program can implement other functions for internal use, or use global variables. Submitted files will be compiled with the grader, and become a single executable file. All global variables and internal functions should be declared in an unnamed namespace to avoid confliction with other files. When it is graded, it will be executed as two processes of Anna and Bruno. The process of Anna and the process of Bruno cannot share global variables.
- Your program must not use the standard input and the standard output. Your program must not communicate with other files by any methods. However, your program may output debugging information to the standard error.

Compilation and Test Run

You can download an archive file from the contest webpage which contains the sample grader to test your program. The archive file also contains a sample source file of your program.

The sample grader is the file `grader.cpp`. In order to test your program, put `grader.cpp`, `Anna.cpp`, `Bruno.cpp`, `Anna.h`, `Bruno.h` in the same directory, and run the following command to compile your programs.

```
g++ -std=gnu++17 -O2 -o grader grader.cpp Anna.cpp Bruno.cpp
```

When the compilation succeeds, the executable file `grader` is generated.

Note that the actual grader is different from the sample grader. The sample grader will be executed as a single process, which will read input data from the standard input and write the results to the standard output.

Input for the Sample Grader

The sample grader reads the following data from the standard input.

Q
A_1
A_2
$\vdots$
A_Q

Output of the Sample Grader

The sample grader outputs the following information to the standard output (quotes for clarity).

- If your program is judged as correct, it writes the return value m of the function `Declare` as "`Accepted: 2000`".
- If your program is judged as any one of Wrong Answer, the sample grader writes its type as "`Wrong Answer [1]`".

If your program satisfies the conditions of several types of Wrong Answer, the sample grader reports only one of them.

When the sample grader is executed, the riffle shuffle for each round is calculated using pseudo random numbers. Their results do not change for executions. In order to change the seed of pseudo random numbers, execute the sample grader by the following way.

```
./grader 2022
```

The first argument should be an integer.

Constraints

- $1 \leq Q \leq 1\,000$.
- $1 \leq A_i \leq 1\,000\,000\,000\,000\,000\,000$ $(= 10^{18})$ $(1 \leq i \leq Q)$.

Subtasks

1. (5 points) $A_i \leq 2\,000$ $(1 \leq i \leq Q)$.
2. (5 points) $A_i \leq 4\,000\,000$ $(1 \leq i \leq Q)$.
3. (3 points) $A_i \leq 10\,000\,000$ $(= 10^7)$ $(1 \leq i \leq Q)$.
4. (12 points) $A_i \leq 100\,000\,000$ $(= 10^8)$ $(1 \leq i \leq Q)$.
5. (15 points) $A_i \leq 100\,000\,000\,000$ $(= 10^{11})$ $(1 \leq i \leq Q)$.
6. (60 points) No additional constraints. For this subtask, your score is calculated as follows.
   - Let m^* be the maximum of the integers m declared by Anna for all test cases of this subtask.
   - Your score is as follows.

The value of m^*	Score
$201 \leq m^* \leq 2\,000$	$40 - 25 \times \log_{10}\left(\frac{m^*}{200}\right)$ points (rounded down to the nearest integer)
$161 \leq m^* \leq 200$	40 points
$156 \leq m^* \leq 160$	44 points
$151 \leq m^* \leq 155$	48 points
$146 \leq m^* \leq 150$	52 points
$141 \leq m^* \leq 145$	56 points
$m^* \leq 140$	60 points

Sample Communication

Here is a sample input for the sample grader and corresponding function calls.

Sample Input 1	Sample Function Calls	
	Call	Return Value

`2`
`42`
`2000`
	`Declare()`	`4`

	`Anna(42)`	`([0, 0, 1, 0], [1, 1, 0, 1])`
	`Bruno([1, 0, 0, 1, 0, 1, 0, 1])`	`42`
	`Anna(2000)`	`([0, 1], [0, 0])`
	`Bruno([0, 0, 1, 0])`	`2000`

In this sample input, there are Q (= 2) rounds. For the round 1, D-taro gives the integer A_1 (= 42) to Anna. For the round 2, D-taro gives the integer A_2 (= 2000) to Anna.

This sample input satisfies the constraints of all the subtasks.