



Casino

Azzurro and Bordeaux, a pair visiting a casino in Italy, decided to play a game proposed by the dealer Chiaro.

In this game, information is transmitted using an $N \times N$ grid ($N = 8$). The rows of the grid are numbered from 0 to $N - 1$ from top to bottom, and the columns of the grid are numbered from 0 to $N - 1$ from left to right. A cell with row number r and column number c is denoted by (r, c) .

In this game, Azzurro and Bordeaux are isolated in separate rooms. They will play Q turns. The i -th turn ($1 \leq i \leq Q$) proceeds as follows.

1. Azzurro receives from Chiaro an integer N , an integer L_i ($1 \leq L_i \leq 51$), a card on which a string S_i of length L_i consisting of 'A' and 'B' is written, and an $N \times N$ grid whose cells are all colored white.
2. Azzurro colors each of the N^2 cells either blue or red. He then hands the grid to Chiaro.
3. Chiaro performs the following operations out of sight of both Azzurro and Bordeaux.
 - (a) He selects one path from $(0, 0)$ to $(N - 1, N - 1)$ by repeatedly moving only to an adjacent cell either down or to the right.
 - (b) For every cell on the path, if the cell is colored blue, he repaints it red; if it is colored red, he repaints it blue.
4. Bordeaux receives from Chiaro a card with integers N and L_i written on it, along with the grid.
5. Bordeaux writes a string of length L_i consisting of A and B on a sheet of paper. If the written string matches S_i , Azzurro and Bordeaux win.

Write programs which implement the strategies of Azzurro and Bordeaux to win this game. For the grading of this task, see **Grading**.



Implementation Details

You need to submit two files.

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

- `std::vector<std::vector<int>> Azzurro(int N, int L, std::string S)`

This function is called Q times. The i -th call ($1 \leq i \leq Q$) corresponds to the procedures 1., 2. of the i -th turn of the game.

- The parameter N is the integer N written on the card given to Azzurro in the procedure 1. of the i -th turn.
- The parameter L is the integer L_i written on the card given to Azzurro in the procedure 1. of the i -th turn.
- The parameter S is the string S_i written on the card given to Azzurro in the procedure 1. of the i -th turn.

For each call to the function `Azzurro`, your program must return an $N \times N$ two-dimensional array x , each of whose elements is either 0 or 1. If this condition is not satisfied, your program is judged as **Wrong Answer [1]**.

- If $x[r][c] = 0$ ($0 \leq r \leq N - 1, 0 \leq c \leq N - 1$), it indicates that the cell (r, c) is colored blue.
- If $x[r][c] = 1$ ($0 \leq r \leq N - 1, 0 \leq c \leq N - 1$), it indicates that the cell (r, c) is colored red.

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

- `std::string Bordeaux(int N, int L, std::vector<std::vector<int>> T)`

This function is called every time when Azzurro finishes painting the grid. This function is called Q times in total. The i -th call ($1 \leq i \leq Q$) corresponds to the procedures 4., 5. of the i -th turn of the game.

- The parameter N is the integer N written on the card given to Bordeaux in the procedure 4. of the i -th turn.
- The parameter L is the integer L_i written on the card given to Bordeaux in the procedure 4. of the i -th turn.
- The parameter T is the $N \times N$ two-dimensional array corresponding to the grid of cells given to Bordeaux in the procedure 4. of the i -th turn. The color of the cell (r, c) ($0 \leq r \leq N - 1, 0 \leq c \leq N - 1$) is blue if $T[a][b] = 0$, and red if $T[a][b] = 1$.



For each call to the function `Bordeaux`, your program must return a string `s` of length L_i consisting of ‘A’ and ‘B’. If this condition is not satisfied, your program is judged as **Wrong Answer [2]**.

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 conflict with other files. When it is graded, it will be executed as two processes of `Azzurro` and `Bordeaux`. The process of `Azzurro` and the process of `Bordeaux` 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`, `Azzurro.cpp`, `Bordeaux.cpp`, `Azzurro.h`, `Bordeaux.h` in the same directory, and run the following command to compile your programs.

```
g++ -std=gnu++20 -O2 -o grader grader.cpp Azzurro.cpp Bordeaux.cpp
```

Instead, you may run `compile.sh` contained in the archive file. In this case, run the following command to compile your programs.

```
./compile.sh
```

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.

In the actual judging grader, **the path chosen by Chiaro is fixed in advance**. That is, the path selected by Chiaro is determined before the functions `Azzurro` and `Bordeaux` in your submitted program are called.



Input for the Sample Grader

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

Q N
 L_1
 S_1
 R_1
 L_2
 S_2
 R_2
 \vdots
 L_Q
 S_Q
 R_Q

Here, R_i ($1 \leq i \leq Q$) is a string of length $2(N - 1)$ consisting of exactly $N - 1$ occurrences of ‘D’ and $N - 1$ occurrences of ‘R’. This string represents the path chosen by Chiaro in the i -th turn, which starts from $(0, 0)$ and reaches $(N - 1, N - 1)$ by repeatedly moving only to an adjacent cell either down or to the right. Specifically, starting from $(0, 0)$, for each $j = 1, 2, \dots, 2(N - 1)$, if the j -th character of R_i is ‘D’, the move is to the adjacent cell below; if it is ‘R’, the move is to the adjacent cell to the right. Repeating this process results in reaching $(N - 1, N - 1)$.

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 value of L^* as “Accepted: 26”. For the value of L^* , see Grading.
- If your program is judged as any type 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. The sample grader may terminate the execution when one of the conditions for wrong answer is met.



Constraints

All the input data satisfy the following conditions.

- $1 \leq Q \leq 30\,000$.
- $N = 8$.
- $1 \leq L_i \leq 51$ ($1 \leq i \leq Q$).
- Q, L_i ($1 \leq i \leq Q$) are integers.
- S_i ($1 \leq i \leq Q$) is a string of length L_i consisting of ‘A’ and ‘B’.
- R_i ($1 \leq i \leq Q$) is a string of length $2(N - 1)$ consisting of exactly $N - 1$ occurrences of ‘D’ and $N - 1$ occurrences of ‘R’.

Grading

If your program is judged as any type of **Wrong Answer [1]** or **Wrong Answer [2]** (see Implementation Details), Time Limit Exceeded, Memory Limit Exceeded, or Runtime Error, in any testcase, your score is 0 points.

Otherwise, let L^* be the minimum of the following values for all test cases of this task. Your score is calculated as in the following table.

- The maximum value of L such that Azzurro and Bordeaux win all the turns satisfying $L_i \leq L$. However, if they win all the turns in the test case, we set $L = 51$.

The value of L^*	Your score
$1 \leq L^* \leq 28$	$2L^*$ points
$29 \leq L^* \leq 39$	$L^* + 28$ points
$40 \leq L^* \leq 50$	$67 + 3(L^* - 40)$ points
$L^* = 51$	100 points



Sample Communication

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

Sample Input 1	Sample Function Calls	
	Function call	Return value
2 2	Azzurro(2, 1, "B")	
		[[1, 0], [0, 1]]
B	Bordeaux(2, 1, [[0, 1], [0, 0]])	
RD		"B"
3	Azzurro(2, 3, "ABB")	
		[[0, 0], [0, 0]]
DR	Bordeaux(2, 3, [[1, 0], [1, 1]])	
		"ABB"

This sample input consists of $Q (= 2)$ turns, and in each turn an $N \times N$ grid ($N = 2$) is used. In this example, the first turn proceeds as follows.

1. Azzurro colors $(0, 1)$ and $(1, 0)$ blue, and $(0, 0)$ and $(1, 1)$ red. He then hands the grid to Chiaro.
2. Chiaro performs the following operations out of sight of Azzurro and Bordeaux.
 - (a) He selects the path $(0, 0) \rightarrow (0, 1) \rightarrow (1, 1)$ as a path from $(0, 0)$ to $(N - 1, N - 1)$ by repeatedly moving only to an adjacent cell either down or to the right.
 - (b) For the three cells $(0, 0)$, $(0, 1)$, $(1, 1)$ on this path, he changes their colors. As a result, the colors of $(0, 0)$, $(0, 1)$, $(1, 1)$ become blue, red, and blue, respectively.
3. Bordeaux can win this turn by writing "B" on the paper.

The second turn proceeds as follows.

1. Azzurro colors all cells blue. He then hands the grid to Chiaro.
2. Chiaro performs the following operations out of sight of Azzurro and Bordeaux.
 - (a) He selects the path $(0, 0) \rightarrow (1, 0) \rightarrow (1, 1)$ as a path from $(0, 0)$ to $(N - 1, N - 1)$ by repeatedly moving only to an adjacent cell either down or to the right.
 - (b) For the three cells $(0, 0)$, $(1, 0)$, $(1, 1)$ on this path, he changes their colors. As a result, the colors of $(0, 0)$, $(1, 0)$, $(1, 1)$ all become red.
3. Bordeaux can win this turn by writing "ABB" on the paper.



The 25th Japanese Olympiad in Informatics (JOI 2025/2026)

Final Stage/Qualifying Trial

March 20–24, 2026 (Komaba, Tokyo)

Contest 2 – **casino**

Note that this sample input **does not satisfy** the constraints of the problem. The file `sample-01-in.txt`, which can be downloaded from the contest site, corresponds to Sample Input 1. The file `sample-02-in.txt`, which can be downloaded from the contest site, is a sample input that satisfies the constraints.