XVI INTERNATIONAL ADVANCED TOURNAMENT IN INFORMATICS
SHUMEN 2025

Task A03. ABSENT STUDENT

⏳ 0.3 sec. 💾 256 MB

Author: Dobrin Bashev

The electronic gradebook crashed again just as Mrs. Jenkova was reporting the students who were absent yesterday. She remembered that exactly one student from her class was absent. The students in Mrs. Jenkova's class are N in total, and they are numbered with natural numbers from 1 to N. Now the teacher wants to find out the number of the absent student.

The electronic gradebook stores a list of the numbers of the present students in a random order. Since it crashed, the only information that can be retrieved is the j-th bit in the binary representation of the i-th number in the list. The numbers in the list are indexed from 1 to $N-1$, and the bits in each number are numbered from 1 to 10, where 1 represents the most significant bit and 10 represents the least significant bit. The numbers are always treated as 10-bit values, and if necessary, the leading bits are padded with 0s.

Task

Write a program **absent** that will find the absent student. It must have the function `play` which will be compiled with the jury's program.

Implementation details

The function `void play(int n)`, which you need to implement, will be called only once by the jury's program and will receive the integer N as its argument.

To communicate with the jury's program, the following two functions are provided:

```
bool get_bit(int i, int j);
void submit_absent(int x);
```

Each call to the `get_bit` function will return 0 (false) or 1 (true), depending on the value of the j-th bit of the i-th number. Note that the execution time of this function is constant.

Once you have identified the absent student, your function should call the `submit_absent` function and pass as an argument an integer x corresponding to their number. After this call, the execution of your function will be terminated.

Your program **absent** must implement the function `play`. It can also contain other code, functions, and global variables, but it must not contain the `main` function. Also, you should not read from the standard input or print to the standard output. Your program must include the header file `absent.h` by instruction to the preprocessor:

```
#include "absent.h"
```

Constraints

- $1 \leq N \leq 1\,000$.

Subtasks

Subtask	Points
1	25
2	25
3	25
4	25

*The points for a given subtask are obtained only if all the tests for it and the required subtasks are **successfully** passed, and the points are equal to the minimum test score in it, multiplied by the points of the subtask.*

Scoring

Each test receives a score that is a fractional number between 0 and 1 inclusive. If a test has a positive score, it is considered **successful** for your solution. A test has a positive score if you successfully find the absent student.

If cnt is the number of calls to the function `get_bit` for a particular test, then the score of the test is calculated in the following way:

- If $cnt \leq 3\,000$, then the score is equal to 1.
- If $3\,000 < cnt \leq 12\,000$, then the score is equal to 0.2.
- If $12\,000 < cnt$, then the score is equal to 0.

Sample communication

Actions of your program	Actions and answers of the jury
	`play(4)`
`get_bit(1, 10)`	`0`
`get_bit(1, 9)`	`0`
`get_bit(2, 10)`	`1`
`get_bit(3, 10)`	`1`
`submit_absent(2)`	

Local testing

For local testing the following files are provided: `absent.h` and `Lgrader.cpp`. When the provided files are in the same folder, you can compile together your program `absent.cpp` and `Lgrader.cpp`. This will make a program to check the correctness of your function.

The program will require from the standard input the following sequence of numbers:

– on the first line: two positive integers – the number of students N and the number of the absent student;
– on the second line: $N - 1$ positive integers separated by spaces - the list with the students that were present.

Your communication will be outputted. You can modify the `Lgrader.cpp` file as you like.