**Задача B4. ОТСЪСТВАЩ УЧЕНИК**

Електронният дневник отново блокира точно когато г-жа Дженкова правеше доклад за отсъстващите през вчерашния ден. Тя си спомни, че точно един от всички ученици в нейния клас е отсъствал. Учениците в класа на г-жа Дженкова са на брой и са номерирани с естествените числа от до . Сега учителката иска да разбере кой е номерът на отсъствалия ученик.

Електронният дневник съхранява списък с номерата на присъствалите ученици в произволен ред. Понеже е блокирал, от него може да се извлече само информация какъв е -тия бит в двоичното представяне на -тото число от списъка. Числата в списъка са индексирани от до , а номерацията на битовете във всяко число е от до , като с е означен най-старшият бит, а с – най-младшият. Числата винаги се разглеждат като битови и при необходимост част от водещите битове се запълват с .

**Задача:**

Напишете функция play(), която ще се компилира с програмата на журито и ще комуникира с нея, с цел да открие отсъстващия ученик.

**Детайли по имплементацията:**

Функцията void play(int n), която трябва да напишете, ще бъде извикана само веднъж от програмата на журито и като аргумент ще бъде получи цялото число .

За комуникация с програмата на журито Ви се предоставят следните две функции:

bool get\_bit(int i, int j);

void submit\_absent(int x);

При всяко извикване на функцията get\_bit, тя ще върне 0 (лъжа) или 1 (истина), в зависимост от стойността на -тия бит на -тото число. Обърнете внимание, че сложността на изпълнение на функцията е константна.

След като сте открили отсъстващия ученик, вашата функция ще трябва да извика функцията submit\_absent и да предаде като аргумент цяло число , отговарящо на неговия номер. След това, изпълнението на вашата функция ще бъде прекратено.

Към системата изпратете файл **absent.cpp**. В него, освен функцията play, може да има всякакви помощни функции, структури, променливи и т. н. Той не трябва да съдържа функция main() и задължително трябва да включва хедър файла absent.h чрез указание към предпроцесора #include "absent.h" в началото си.

**Пример:**

|  |  |
| --- | --- |
| Функция на участника | Програма на журито |
|  | 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) |  |

**Ограничения:**

**Оценяване:**

Тестовете са разпределени в четири подзадачи, всяка от които носи по максимум 25 точки. Ако във всички тестове от дадена подзадача сте успели да отгатнете цветовете с не повече от 3000 заявки – получавате максималния брой точки. Ако в някои от тестовете сте използвали повече от 3000 заявки, но в никой от тях не повече от 12000, ще получите 5 точки. В противен случай няма да получите точки.

**Локално тестване:**

Предоставен Ви е файлът Lgrader.cpp, който може да компилирате заедно с вашата програма, за да я тествате. При стартиране ще Ви пита за 𝑁, след което за номера на отсъстващия ученик и списъка с присъстващите ученици. Ще отпечатва комуникацията, която се извършва. Може да модифицирате този файл както искате.