



НАЦИОНАЛНА ШКОЛА ПО ИНФОРМАТИКА КОНТРОЛНО СЪСТЕЗАНИЕ

Габрово, 28 август 2026 г.

8 клас



Задача Т82. ПЕЩЕРА

2 сек. 256 MB

Будите се облени от студена пот. Сънували сте сън, не, кошмар. В пещера сте. Изходът навън е защитен от N последователни стоманени врати, всяка разположена зад предходната, и N ключа, всеки свързан с една различна врата.

Вратите са номерирани от 0 до $N - 1$, като врата 0 е най-близо до Вас. Има и N ключа, също номерирани от 0 до $N - 1$, но не знаете кой ключ коя врата управлява.

Всеки ключ може да бъде в едно от две положения – нагоре (стойност 0) или надолу (стойност 1). Само едно от тези положения е правилно за даден ключ. Ако ключът е в правилното положение, съответната врата ще бъде отворена. Ако е в грешното – вратата ще бъде затворена. Правилните положения могат да са различни за различните ключове, и в началото Вие нямате никаква информация за тях.

Можете да избирате произволна комбинация от положения на ключовете, след което да влезете в пещерата, за да видите коя е първата затворена врата. Вратите не са прозрачни – щом срещнете първата затворена врата, не може да продължите нататък, както и да видите какво има зад тази врата. Имате право да изпробвате най-много 70 000 комбинации. Целта Ви е да определите както правилното положение за всеки ключ, така и коя врата управлява.

Детайли по имплементацията

Напишете програма `cave`, имплементираща функцията:

```
void explore(int N)
```

Функцията ще бъде извикана веднъж и трябва да използва предоставените функции на журито, за да определи:

- правилното положение (нагоре/надолу) за всеки ключ;
- номера на вратата, която управлява всеки ключ.

За тази задача са предоставени следните функции от журито:

```
int attempt(const std::vector<int>& Q)  
void answer(const std::vector<int>& S, const std::vector<int>& D)
```

Функцията `attempt` получава като аргумент вектор с дължина N , съдържащ положението (0 или 1) на всеки ключ и връща номера на първата затворена врата след задаване на тази комбинация и влизане в пещерата. Ако всички врати са отворени, функцията връща N . Функцията работи за $O(N)$ време.

Функцията `answer` получава като аргумент два вектора S и D с дължина N . S съдържа положението (0 или 1) на всеки ключ. D съдържа информация за това коя врата кой ключ отваря – $D[i]$ е индексът на вратата, която се отваря при правилно зададен i -ти ключ.

Вашата програма `cave` трябва да имплементира функцията `explore`. Тя може да съдържа и друг код, и функции, необходими за работата Ви, но не трябва да съдържа главната функция `main`. Също така, не трябва да четете от стандартния вход или да отпечатвате на стандартния изход. Програмата Ви трябва да включва хедър файла `cave.h` чрез указание към предпроцесора:

```
#include "cave.h"
```



НАЦИОНАЛНА ШКОЛА ПО ИНФОРМАТИКА
КОНТРОЛНО СЪСТЕЗАНИЕ
Габрово, 28 август 2026 г.
8 клас



Ограничения

- $1 \leq N \leq 5\,000$

Подзадачи

№	Точки	Необходимы подзадачи	N	Други ограничения
0	0	—	—	Примерът.
1	12	—	$\leq 5\,000$	Ключ i управлява врата i .
2	13	—	$\leq 5\,000$	Винаги вярна комбинация: $[0, 0, \dots, 0]$.
3	21	0	≤ 100	—
4	30	0, 3	$\leq 2\,000$	
5	24	0 — 4	$\leq 5\,000$	

Точките за дадена подзадача се получават само ако се преминат успешно всички тестове, предвидени за нея.

Примерна комуникация

Извикване на функция	Обяснение
<code>attempt([1, 0, 1, 1])</code>	Функцията връща 1. Врата 0 е отворена, врата 1 е първата затворена.
<code>attempt([0, 1, 1, 0])</code>	Функцията връща 3. Врати 0, 1, 2 са отворени, врата 3 е затворена.
<code>attempt([1, 1, 1, 0])</code>	Функцията връща 4. Всички врати са отворени.
<code>answer([1, 1, 1, 0], [3, 1, 0, 2])</code>	Подаване на решението; програмата приключва.

Локално тестване

Може да използвате предоставения локален грейдър. Той очаква вход със следния формат:

- ред 1: N
- ред 2: Правилните положения на ключовете $S[0]$ до $S[N-1]$
- ред 3: Съответствията $D[0]$ до $D[N-1]$

Примерната комуникация би била въведена по следния начин:

4

1 1 1 0

3 1 0 2