



# НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Пловдив, 12 – 14 юни 2026 г.

Група С – 7, 8 клас

## Задача С3. СЪСТАВЯНЕ НА УСЛОВИЕ

0.9 сек. 1024 MB

Изготвяте задача за национално състезание по информатика и пред Вас е немалко трудната задача – съставянето на условието. Старателно се опитвате да го напишете, но се виждате в чудо да удовлетворите изискванията на Дени, Атина, Фортуна, Сашка и други комисионери, отговорни за състезанието. Има различни критерии за условието, от много незначителни – дали името на задачата да е с главни букви, до доста по-значителни като това дали главният герой да се казва **Пътанаско** или **Насклон**. Поради малкото оставащо време, решавате да изпълните поне по едно изискване на всеки комисионер, за да се приеме задачата на време за състезанието.

Формално, има  $N$  критерия, означени с числата от 0 до  $N - 1$ , като всеки критерий е булев – може или да бъде изцяло изпълнен, или да не бъде въобще изпълнен. Ще наричаме разпределение всяка възможност, която определя за всеки критерий дали да бъде изпълнен или не.

Също така, има  $M$  члена на комисията, като  $i$ -тият ( $0 \leq i \leq M - 1$ ) има списък от  $l_i$  изисквания ( $1 \leq l_i \leq 2N$ ), които са двойки стойности –  $(c_0, p_0), (c_1, p_1), \dots, (c_{l_i-1}, p_{l_i-1})$ , където  $c_j$  е номер на критерий, а  $p_j$  е булева стойност за това дали комисионера иска да бъде изпълнен критерия или иска да не бъде изпълнен ( $0 \leq j \leq l_i - 1$ ). Гарантирано е, че няма повтарящи се двойки в списъка. Обърнете внимание, че все пак е възможно да има две изисквания за един и същи критерий  $x$  ( $0 \leq x \leq N - 1$ ):  $(x, \text{true})$  и  $(x, \text{false})$ ; все пак някои комисионери са доста заблеляни.

Освен това, се оказва, че всички тези критерии и изисквания са възникнали в двустранни спорове и диалози и съответно всеки критерий се среща **най-много общо два пъти** сред списъците от изисквания на комисионерите.

Разпределение, удовлетворяващо поне едно изискване на всеки комисионер, ще наричаме **валидно**. Напишете програма **statement**, която по списъците от изисквания на членовете на комисията, намира **валидно** разпределение или открива, че няма такова.

### Детайли по имплементацията

Трябва да имплементирате функцията `satisfy`:

```
std::vector<bool> satisfy(int N,  
    std::vector<std::vector<std::pair<int, bool>>> S)
```

- $N$  – брой критерии;
- $S$  – вектор със списъците на всички  $M$  комисионери, като всеки списък се състои от двойки стойности, които задават изискване, описано с номер на критерий и булева стойност за това дали трябва да бъде изпълнен или не.

Тази функция ще бъде извикана веднъж в рамките на едно изпълнение на програмата и трябва да върне празен вектор, ако не съществува **валидно** разпределение, или вектор с  $N$  булеви стойности  $b_0, \dots, b_{N-1}$ . В този случай тези стойности трябва да са **валидно** разпределение, като  $b_i$  задава дали е изпълнен  $i$ -тият критерий. Ако има повече от едно **валидно** разпределение, върнете което и да е.

### Ограничения

- $1 \leq N, M \leq 5 \times 10^5$
- Гарантирано е, че за всеки списък няма повтарящи се двойки в него. Също така всеки критерий се среща най-много общо два пъти сред списъците на комисионерите.



# НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Пловдив, 12 – 14 юни 2026 г.

Група С – 7, 8 клас

## Подзадачи

| № | Точки | Необходими подзадачи | $N, M$               | Други ограничения   |
|---|-------|----------------------|----------------------|---|
| 0 | 0     | —                    | —                    | Примерите.  |
| 1 | 13    | —                    | $\leq 5 \times 10^5$ | Всеки списък има само едно изискване или има изискване, за което е изпълнено <b>едно от двете</b> : <ul style="list-style-type: none"><li>изискването се среща и в друг списък;</li><li>критерият му не се среща в друг списък.</li></ul> |
| 2 | 13    | —                    |                      | За всеки два списъка е изпълнено <b>едно от двете</b> : <ul style="list-style-type: none"><li>няма общи критерии между двата списъка;</li><li>всички критерии от единия списък се срещат и в другия списък.</li></ul>                     |
| 3 | 22    | —                    |                      | Всеки списък има най-много две изисквания.  |
| 4 | 8     | 0                    | $N \leq 20$          | $M \leq 40$   |
| 5 | 21    | 0, 4                 | $\leq 2 \times 10^3$ | —   |
| 6 | 23    | 0 – 5                | $\leq 5 \times 10^5$ | —   |

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

## Примери

| Вход    | Изход | Вход  | Изход |
|---------|-------|-------|-------|
| 5 5     | 0     | 4 5   |       |
| 1       | 1     | 1     |       |
| 3       | 1     | 3     |       |
| 2       | 1     | 2     |       |
| 2 !3    | 0     | 2 !3  |       |
| 3       |       | 2     |       |
| 4 !0 !2 |       | !0 !2 |       |
| 2       |       | 1     |       |
| !4 0    |       | 0     |       |
| 1       |       | 1     |       |
| 1       |       | 1     |       |

## Локален грейдър

Формат на входа:

- ред 1: две цели числа  $N$  и  $M$  – брой критерии и брой комисионери;
- ред  $2 + 2i$ : едно цяло число  $l_i$  – брой изисквания на  $i$ -тия комисионер;
- ред  $2 + 2i + 1$ :  $l_i$  на брой низа, разделени с по един интервал – изискванията на  $i$ -тия комисионер, като низ, който е номер на критерий  $c$ , задава изискването  $(c, \text{true})$ , а низ, започващ с ! и долепен номер на критерий, задава изискването  $(c, \text{false})$ .

Формат на изхода:

- ред  $i$ :  $i$ -тата върната стойност при извикването на `satisfy`.