



# ТРЕНИРОВЪЧНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

София, 27 юни 2026 г.

Група А

## Задача AT23. ПЪТУВАНЕ ВЪВ ВРЕМЕТО

7 сек. 1024 MB

Сашка и Тестенчо са решили да се промъкнат в едно правителствено съоръжение, за да стигнат до скритата машина на времето и да поправят случилото се на IATI 2026. Достъпът до машината е заключен зад охранителна система. При опит за достъп до машината, на екрана се появява цяло число  $q$  с  $1 \leq q \leq N$ , а операторът трябва да отговори с *да* или *не*, като при верен отговор достъпа се дава, а грешен отговор би включил алармата. Известно е, че показаното число  $q$  винаги е или  $x$  (чийто верен отговор е “да”), или  $y$  (чийто верен отговор е “не”), при  $x \neq y$ .

За жалост, докато подготвят удара, двамата са забравили кои точно са  $x$  и  $y$ . Затова Тестенчо отива сам пред машината, а Сашка остава по-далеч, за да пази. Точно когато въпросът светва, Сашка изведнъж си спомня и  $x$ , и  $y$  – но от разстояние може единствено да подвикне на Тестенчо едно цяло число  $h$  ( $h \geq 1$ ), в което трябва да предаде всичко необходимо, за да отговори Тестенчо вярно.

Напишете програма **timetravel**, която изпълнява и двете роли: по  $N$ ,  $x$  и  $y$  да определи числото  $h$ , което Сашка подвиква, и по  $N$ ,  $q$  и  $h$  – отговора, който Тестенчо дава. При тестване решението се изпълнява като **два отделни процеса** без обща памет: в единия грейдърът извиква `encode` за всеки от  $T$  случая, а в другия – `answer` с върнатото за този случай  $h$ .

### Детайли по реализацията

Трябва да предадете файла `timetravel.cpp`, който включва `timetravel.h` и реализира функциите `encode` и `answer`:

```
int encode(int N, int x, int y)
```

където:

- $N$  е броят възможни въпроси;
- $x$  и  $y$  са две различни числа между 1 и  $N$ ;
- функцията връща числото  $h$  ( $h \geq 1$ ), което Сашка да извика.

```
bool answer(int N, int q, int h)
```

където:

- $N$  е броят възможни въпроси;
- $q$  е зададеният въпрос;
- $h$  е числото, подадено от функцията `encode`;
- функцията връща отговора, който Е да избере: `true` за  $q = x$  и `false` за  $q = y$ .

Двата процеса са напълно отделни: стойности, записани в глобални променливи от `encode`, не са достъпни за `answer`, и обратно.

### Ограничения

- $N = 920$  ( $N$  се подава във входа на функциите, но винаги е равно на 920);
- $1 \leq T \leq 2 \cdot 10^6$ .



# ТРЕНИРОВЪЧНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

София, 27 юни 2026 г.

Група А

## Оценяване

Задачата има две подзадачи – нулевата се състои от примерния тест, първата се състои от единствен тест с всички  $N(N - 1)$  двойки  $(x, y)$ , изброени по веднъж. Точките на решението се определят от най-голямото число  $h$ , което Сашка извиква за теста в първата подзадача:

Най-голямо $h$	Точки
$\geq 21$	0 (отчита се като грешен изход)
20	27
19	30
18	33
17	37
16	42
15	50
14	60
13	75
$\leq 12$	100

Освен това всички отговори на функцията `answer` за всички тестове трябва да са верни; в противен случай решението получава 0 точки. Ограниченията за време и памет важат за всеки от двата процеса поотделно.

## Локален грейдър

Предоставени са ви хедър файлът `timetravel.h` и локален грейдър `grader.cpp`, които да компилирате заедно с програмата си. Локалният грейдър **не** използва отделни процеси, т.е. внимавайте да не споделяте глобална информация между `encode` и `answer`, без да искате – това няма да работи на оценяващата система.

Формат на входа:

- ред 1: две цели числа  $N$   $T$  – стойността  $N$  и броят случаи;
- редове 2 до  $T + 1$ : по три цели числа  $x$   $y$   $q$  за съответния случай.

Формат на изхода:

- ред 1: ОК и най-голямата получена стойност  $h$  при верни отговори, или ERR и описание на грешката.

Форматът на примерите следва формата на локалния грейдър.



**ТРЕНИРОВЪЧНО СЪСТЕЗАНИЕ  
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР  
София, 27 юни 2026 г.  
Група А**

*Примери*

Тестов файл	Извиквания на <code>encode</code>	Извиквания на <code>answer</code>
920 6	<code>encode(920, 1, 2) = 12</code>	<code>answer(920, 1, 12) = true</code>
1 2 1	<code>encode(920, 4, 5) = 2</code>	<code>answer(920, 4, 2) = true</code>
4 5 4	<code>encode(920, 1, 2) = 12</code>	<code>answer(920, 2, 12) = false</code>
1 2 2	<code>encode(920, 3, 5) = 4</code>	<code>answer(920, 3, 4) = true</code>
3 5 3	<code>encode(920, 4, 5) = 2</code>	<code>answer(920, 5, 2) = false</code>
4 5 5	<code>encode(920, 5, 2) = 1</code>	<code>answer(920, 2, 1) = false</code>
5 2 2		

*Много други стойности `h` са валидни, но при показаните стойности на `h` изведените отговори на `answer` са единствените верни. Най-голямото `h` тук е 12, което носи пълни точки.*