



## ТРЕНИРОВЪЧНО СЪСТЕЗАНИЕ МОМИЧЕТА

Пловдив, 12-14 юни 2026 г.

Група G

### Задача GT2. Brackets (ПревземиПловдивинатор)

⌚ 0.5 сек. 💾 256 MB

Помогнете на агент Пери да спре злия Дуфеншмърц от превземане на Пловдив, като за целта трябва да разбиете паролата на лаптопа му.

Единственото, което знаете за нея обаче, е дължината ѝ и факта, че се състои само от нормални скоби – ( и ) . Освен това знаете също, че **низът има четна дължина и съдържа равен брой отварящи и затварящи скоби**. За ваше щастие, агент Хари е успял да внедри приложение в лаптопа, което при зададени индекси  $L$  и  $R$  ще ви отговори дали низът, започващ от позиция  $L$  и завършващ в позиция  $R$ , е правилен. Номерацията на низа започва от 1.



Една редица от скоби дефинираме като математически правилна по следния начин:

- ( ) е правилна
- Ако  $A$  е правилна редица, то  $(A)$  също е правилна
- Ако  $A$  и  $B$  са правилни редици, то конкатенацията им  $AB$  също е правилна редица

Все пак Дуфеншмърц не е глупав и има антивирусна програма, която ще се задейства, ако се направят повече от  $Q$  питания.

Помогнете на Пери, като напишете програма, която да познае паролата вместо него.

Това е **интерактивна задача**.

#### Детайли по имплементацията

Вашата функция `findBrackets` трябва да има следния прототип:

```
std::string findBrackets(int N, int Q);
```

Тя ще бъде извикана точно веднъж и ще получи като аргумент дължината на редицата и максималния брой заявки, които можете да направите. Функцията трябва да върне низ с дължина  $N$ , който представлява търсената парола.

Функция `isValid` на журито има следния прототип:

```
bool isValid(int L, int R);
```

Вашата програма може да я вика максимум  $Q$  пъти. Като аргументи се дават двата индекса. Тя връща дали редицата, започваща от позиция  $L$  и завършваща в позиция  $R$ , е математически правилна (индексирането започва от 1). Сложността на функцията е  $O(1)$ . Следните изисквания винаги трябва да са изпълнени:  $1 \leq L, R \leq N$ .

Вашата програма трябва да имплементира функцията `findBrackets`, но **не трябва** да съдържа функция `main`. Освен това, тя не трябва да чете от стандартния вход или да печата на стандартния изход. Програмата Ви също така трябва да включва хедър файла `brackets.h` чрез указание към препроцесора:

```
#include "brackets.h"
```

Стига да спазва тези условия, програмата Ви може да съдържа каквито и да е помощни функции, променливи, константи и прочие.



**ТРЕНИРОВЪЧНО СЪСТЕЗАНИЕ**  
**МОМИЧЕТА**  
**Пловдив, 12-14 юни 2026 г.**  
**Група G**

Ограничения

- $1 \leq N \leq 10^5$

Подзадачи

Подзадача	Точки	Допълнителни ограничения
1	14	$1 \leq N \leq 1000$ , $Q = \frac{N^2}{4}$ , цялата парола е математически валидна редица
2	7	$1 \leq N \leq 1000$ , $Q = \frac{N^2}{4}$
3	57	$1 \leq N \leq 10^5$ , $Q = N - 1$ , цялата парола е математически валидна редица
4	22	$1 \leq N \leq 10^5$ , $Q = N - 1$

Локално тестване

Предоставени са Ви файловете `brackets.h` и `Lgrader.cpp`, които можете да компилирате заедно с Вашата програма, за да я тествате.

При стартиране на програмата трябва първо да се въведат числата  $N$  и  $Q$ , а след това паролата (от  $N$  символа). След това решението Ви ще се изпълни и ще се изведе броят заявки, които то е направило.

Забележете, че локалният грейдър работи със сложност на заявката  $O(N)$ , а истинският с  $O(1)$ . Чувствайте се свободни да променят грейдъра, така че да Ви бъде максимално полезен.

Примерна комуникация

No	Действия на <code>brackets</code>	Действия и отговори на журито
1	<code>findBrackets(6, 9)</code>	
2	<code>isValid(1, 6)</code>	<code>return 1</code>
3	<code>isValid(1, 2)</code>	<code>return 0</code>
4	<code>isValid(2, 4)</code>	<code>return 0</code>
5	<code>isValid(2, 5)</code>	<code>return 1</code>
6	<code>isValid(3, 4)</code>	<code>return 1</code>
7	<code>return ((( )))</code>	