



Задача Грейдър67

 0.1 sec.  256 MB

(Поради технически проблеми задачата не може да се казва grader.)

Дадено е число N . Написана е програма, която връща редица a с N елемента, който изпълнява следните условия:

- има поне една двойка индекси $1 \leq i, j \leq N, i \neq j$, където $a_i \neq a_j$.
- $\sum_{i=1}^N a_i \equiv 0 \pmod{N}$.
- разликата $\max(a_1, a_2, \dots) - \min(a_1, a_2, \dots)$ е минимална.



Детайли по имплементацията

Вашата задача е да напишете грейдър, който проверява коректността на даденото решение. То е имплементирано чрез функцията:

```
std::vector<int> solve(int N);
```

Не трябва да я имплементирате! Трябва да имплементирате функцията `main`, която въвежда N , извиква `solve` и проверява дали функцията връща правилно решение. Не е гарантирано, че върнатата редица отговаря на всички условия. Грейдърът трябва да отпечата едно число между 0 и 1 - точките на даден тест. Точките се смятат по следния начин:

- изпълнено първо условие носи 0.15 на тест;
- изпълнено второ условие носи 0.25 на тест;
- изпълнено трето условие носи 0.60 на тест.



Ограничения

- $1 \leq N \leq 10^5 + 67$



Подзадачи

Подзадача	Точки	Ограничение
1	15	$1 \leq N \leq \text{около } 6 - 7$
2	20	$1 \leq N \leq 30$
3	25	$1 \leq N \leq 100$
4	20	$1 \leq N \leq 1067$
5	20	$1 \leq N \leq 10^5 + 67$



Пример

Жури	Състезател
	<code>solve(5)</code>
2, 1, 3, 2, 3	Грейдърът връща 0.75.

Този грейдър получава пълни точки на теста, защото правилно открива точките, които получава върнатата редица.



Примерен грейдър

Предоставен Ви е файл `Lgrader.cpp`, който действа като грейдър на Вашия грейдър. За да използвате функцията за комуникацията, във Вашия код трябва да включите библиотеката `grader67.h`:

```
#include "grader67.h"
```

Форматът на входа е следния:

- ред 1: числото N .
- ред 2: масива a , върху който ще се грейдва Вашия грейдър.

След това се изпълнява Вашия грейдър, който извежда точките, които получава това решение.