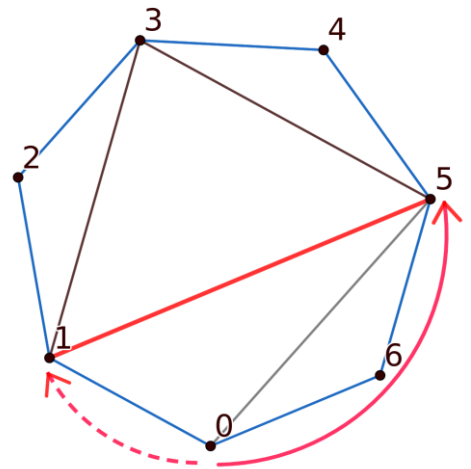


Triangulation

Постановка задачи

Анна нарисовала правильный многоугольник с n вершинами, пронумерованными от 0 до $n - 1$ по часовой стрелке. Позже она триангулировала его, начертив $n - 3$ диагонали (прямые линии между двумя разными вершинами, не имеющими общей стороны), не пересекающиеся друг с другом, за исключением, возможно, касания в конечных точках. Анна хочет сделать из этого испытание для Джейкоба.

Во-первых, давайте определим, что является расстоянием от точки A до диагонали D . Допустим, мы начинаем с точки A и движемся до следующей вершины по часовой стрелке, пока не достигнем одного из концов диагонали D . Количество пройденных сторон назовем **левым_расстоянием**. Аналогично, **правым_расстоянием** назовем количество пройденных сторон, если мы начинаем с точки A и движемся против часовой стрелки, пока не достигнем D . Расстояние от A до D это **максимум** из **левого_расстояния** и **правого_расстояния**.



В примере, изображенном на рисунке, расстояние от точки 0 до диагонали(1,5) равно 2, так как левое_расстояние равно 1, а правое_расстояние равно 2. Для диагонали(0,5) расстояние от вершины 0 равно 5, так как левое_расстояние=5 и правое_расстояние=2.

Джейкоб не знает, как нарисованы диагонали. Он знает только значение N , но может задавать вопросы Анне про некоторые пары вершин, на которые она ответит, проведена ли диагональ между этими вершинами. Цель Джейкоба найти ближайшую (по определению расстояния, приведенному выше) нарисованную диагональ от вершины 0. Вашей задачей является помочь ему пройти испытание, задав Анне ограниченное количество вопросов.

Ограничения

- $5 \leq n \leq 100$

Детали реализации

Вы должны реализовать следующую функцию в своей посылке:

```
int solve(int n)
```

- Эта функция вызывается грейдером ровно один раз
- n : количество вершин многоугольника
- Эта функция должна вернуть диагональ между некоторыми вершинами a и b как целое число равное $a \cdot n + b$
- Если существует несколько диагоналей с минимальным расстоянием, Вы можете вернуть любую из них

Вышеуказанная функция может вызывать следующую функцию:

```
int query(int x, int y)
```

- x : номер первой вершины
- y : номер второй вершины
- $0 \leq x, y \leq n$
- Возвращает 1, если вершины x и y соединены диагональю, и 0 в противном случае

Пример взаимодействия

Приведем пример ввода для грейдера и соответствующих вызовов функции. Пример соответствует изображению, приведенному ранее.

Единственная строка ввода содержит одно целое число: n

Грейдер будет выводить каждый запрос в `stdout` и Вы должны вручную ответить 1 или 0.

Пример ввода для грейдера	Пример вызовов			
	Вызовы	Возвраты	Вызовы	Возвраты
7	solve(7)			
			query(0, 3)	
				query вернула 0
			query(0, 5)	
				query вернула 1
			query(1, 5)	
				query вернула 1
		solve вернула $1 \cdot 7 + 5 = 12$		
		Верно!		

Оценка

Пусть q это количество запросов, использованных Вами в тесте. Также, $w = \frac{n \cdot (n-3)}{2}$.

- Если Вы задали некорректный вопрос или дали некорректный ответ, Вы получаете 0% баллов за тест
- Если $w < q$, Вы получаете 0% баллов за тест
- Если $n < q \leq w$, Вы получаете $10 + 60 \cdot \frac{w-q}{w-n}$ % баллов за тест
- Если $q \leq n$, Вы получаете 100% баллов за тест

Подзадачи

Существует единственная подзадача и Ваш результат является суммой результатов за каждый тест. Но во время соревнования Вы будете видеть только половину результатов за тесты (на 50 баллов). Другая половина результатов будет открыта после соревнования. Ваш финальный результат будет равен **лучшему результату среди всех посылок**.