

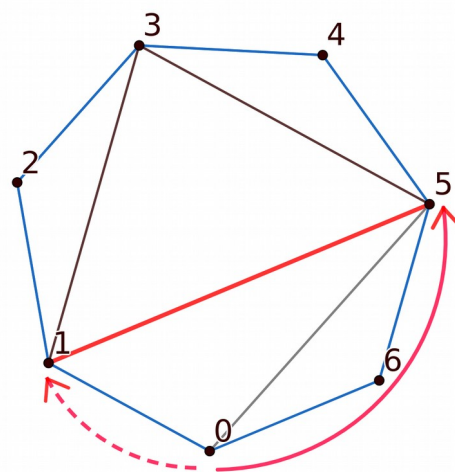
# Triangulation

## Текст

Ана је нацртала правилан многоугао са  $n$  темена нумерисаних од 0 до  $n-1$  у смеру казаљке на сату. Касније га је триангулисала цртањем  $n-3$  дијагонала (дужима између два темена која не деле страну) које се не секу осим да се можда додирују у крајњим тачкама (теменима). Ана жели да направи изазов за Јакова.

Прво, хајде да дефинишемо растојање од темена  $A$  до дијагонале  $D$ . Рецимо да кренемо из темена  $A$  и увек се крећемо до следећег темена у смеру казаљке на сату док не стигнемо до једне од крајњих тачака  $D$ . Број страница које смо тамо прешли називамо **left\_distance**. Слично, **right\_distance** је број страница ако кренемо из  $A$  и крећемо се супротно од казаљке на сату док не стигнемо до  $D$ . Растојање од  $A$  до  $D$  је максимум од **left\_distance** и **right\_distance**.

У примеру са слике растојање од темена 0 до дијагонале (1,5) је 2, где **left\_distance** једнак 1, а **right\_distance** једнак 2. Растојање од темена 0 до дијагонале (0,5) је 5, где **left\_distance**=5 и **right\_distance**=2.



Јаков уопште не зна које дијагонале су нацртане. Он једино зна вредност  $n$ , али може да пита Ану више пута за неки пар темена, на шта ће она одговорити са тим да ли постоји дијагонала између та два темена. Јаковов циљ је да нађе најближу (где је растојање дефинисано малопре) нацртану дијагоналу од темена 0. Помозите му да ово уради тако што ћете питати Ану неколико питања.

## Ограничења

- $5 \leq n \leq 100$

## Детаљи имплементације

Потребно је да имплементирате следећу функцију у вашем коду:

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

- Ову функцију ће оцењивач звати тачно једном
- $n$ : број темена многоугла
- Ова функција би требало да врати дијагоналу између нека два темена  $a$  и  $b$  као цео број са вредношћу  $a \cdot n + b$ .
- Ако има више дијагонала са најкраћим растојањем од 0 онда вратите било коју.

Ваш програм сме да позива следећу функцију:

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

- $x$ : број првог темена
- $y$ : број другог темена
- $0 \leq x, y \leq n$
- враћа 1 ако постоји дијагонала између  $x$  и  $y$ , а 0 у супротном

## Пример интеракције

Ово је пример улаза за грејдер и одговарајући позиви функција. Овај улаз је узет од слике са прве стране.

У јединој линији улаза налази се  $n$ .

Оцењивач ће исписати сваки позив `query` на `stdout` и потребно је да ви ручно одговорите са 1 или 0.

Пример улаза оцењивачу	Примери позива			
	Позиви	Враћене вредност	Позиви	Враћене вредности
7	<code>solve(7)</code>			
			<code>query(0, 3)</code>	
				<code>query</code> враћа 0
			<code>query(0, 5)</code>	
				<code>query</code> враћа 1
			<code>query(1, 5)</code>	
				<code>query</code> враћа 1

		solve враћа $1 \cdot 7 + 5 = 12$		
		Correct!		

## Бодовање

Означимо са  $q$  број позива функцији `query` који сте користили у једном примеру. Такође,

$$w = \frac{n \cdot (n-3)}{2}.$$

- Ако питате невалидно питање или одговорите нетачно, добићете 0% поена за тај пример
- Ако  $w < q$  добићете 0% поена за тај пример
- Ако  $n < q \leq w$  добићете  $10 + 60 \cdot \frac{w-q}{w-n} \%$  поена за тај пример
- Ако  $q \leq n$  добићете 100% поена за тај пример

## Подзадаци

Постоји само један подзадатак и ваш резултат је збир резултата на појединачним примерима. Али, за време такмичења моћи ћете да видите свој резултат само на пола примера (вредних 50 поена). Остатак резултата биће откривени после такмичења. Ваш резултат ће бити **најбољи резултат од свих слања вашег решења**.