

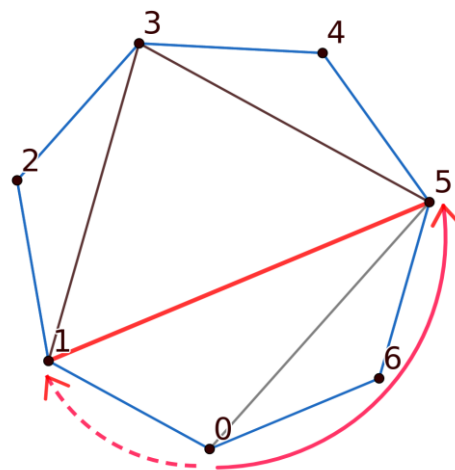
Триангулација

Атина нацрта правилен (рамностран) многуаголник со n темиња, нумерирани од 0 до $n-1$ по редослед на стрелките на часовникот. Подоцна, таа го триангулираше цртајќи $n-3$ дијагонали (отсечки помеѓу две различни темиња што не делат страна) кои не се пресекуваат едни со други, освен ако не се допираат во крајните точки. Атина сака од тоа да направи предизвик за Јан.

Прво, да го дефинираме растојанието од темето A до дијагоналата D . Да претпоставиме дека започнуваме од темето A и продолжуваме да се движиме кон следното теме во насока на движење на стрелките на часовникот, сè додека не стигнеме до една од крајните точки на D . Бројот на изминати страни ќе го викаме **left_distance**. Слично на тоа, **right_distance** е бројот на страни што се поминуваат ако започнеме од A и се движиме спротивно од стрелките на часовникот, сè додека не стигнеме до D . **Растојание** од A до D е максимумот од **left_distance** и **right_distance**.

На пример-сликата, растојанието од темето 0 до дијагоналата (1,5) е 2, затоа што **left_distance** е 1, а **right_distance** е 2. За дијагоналата (0,5), растојанието од темето 0 е 5, бидејќи **left_distance**=5 и **right_distance**=2.

Јан не знае кои од дијагоналите се исцртани. Тој знае само колку е n , но повеќе пати може да ја праша Атина за некои парови на темиња и таа ќе му каже дали постои дијагонала помеѓу тие две темиња. Целта на Јан е да ја најде најблиску нацртаната дијагонала (со растојание дефинирано како погоре) од темето 0. Вие ќе му помогнете да го постигне тоа со поставување на ограничен број прашања на Атина.



Ограничувања

- $5 \leq n \leq 100$

Имплементациски детали

Треба да ја имплементирате функцијата:

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

- Функцијата се повикува точно еднаш од оценувачот
- n : број на темиња на многуаголникот
- Функцијата треба да ја враќа дијагоналата меѓу некои од темињата a и b како цел број со вредност $a \cdot n + b$.
- Ако има повеќе дијагонали на минимално растојание, може да ја вратите која било од нив.

Горната функција може да ја повикува функцијата:

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

- x : бројот на првото теме
- y : бројот на второто теме
- $0 \leq x, y \leq n - 1$
- враќа 1 ако има дијагонала меѓу x и y , или пак 0 ако нема

Пример за интеракција

Еве пример влез за оценувачот и соодветните функциски повици. Овој влез е прикажан на горната слика.

Во единствениот ред од влезот е запишан еден цел број: n .

Оценувачот ќе го отпечати секој функциски повик на `stdout` и вие ќе треба рачно да го одговорите со 1 или 0.

Пример влез за оценувачот	Пример повици			
	Повици	Враќа	Повици	Враќа
7	solve(7)			
			query(0, 3)	
				query returns 0
			query(0, 5)	
				query returns 1
			query(1, 5)	
				query returns 1
		solve returns $1 \cdot 7 + 5 = 12$		
		Correct!		

Оценување

Нека го означиме со q бројот на прашања кои сте ги искористиле во еден тест случај.

Нека $w = \frac{n \cdot (n-3)}{2}$.

- Ако поставите невалидно прашање или дадете погрешен одговор ќе добиете 0% од поените за тој тест
- Ако $w < q$ ќе добиете 0% од поените за тој тест
- Ако $n < q \leq w$ ќе добиете $10 + 60 \cdot \frac{w-q}{w-n}\%$ од поените за тој тест
- Ако $q \leq n$ ќе добиете 100% од поените за тој тест.

Подзадачи

Има само една подзадача и резултатот е збир од освоените поени на секој тест. Но, дури трае натпреварот ќе можете да ги видите само освоените поени за половина од тестовите (кои носат 50 поени). Другата половина ќе биде откриена по завршување на натпреварот. Вашиот финален резултат ќе биде **најдобриот вкупен резултат од сите решенија (поднесоци, submissions) што сте ги испратиле.**