

Triangulation

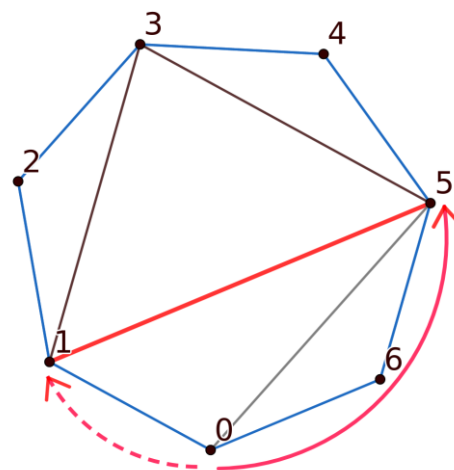
Zadatak

Ana je nacrtala mnogokut s n vrhova označenih od 0 do $n - 1$ u smjeru kazaljke sata. Potom ga je podijelila na trokute crtajući $n - 3$ dijagonala koje se ne sijeku, osim eventualno dodiruju u vrhovima. Dijagonala je dužina između dvaju različitih vrhova mnogokuta koji ne dijele stranicu.

Definirajmo udaljenost vrha A do dijagonale D.

Pretpostavimo da krećemo od A i mičemo se u smjeru kazaljke sata dok ne dođemo do jedne od krajnjih točaka dijagonale D. Broj prijeđenih stranica zovemo **lijeva_udaljenost**. Slično, **desna_udaljenost** je broj prijeđenih stranica počevši od A i mičući se u smjeru suprotnom od kazaljke sata dok ne dođemo do D. Tada je **udaljenost** od A do D **maksimum** brojeva **lijeva_udaljenost** i **desna_udaljenost**.

U primjeru na slici, udaljenost vrha 0 do dijagonale (1, 5) iznosi 2 gdje je lijeva_udaljenost=1 i desna_udaljenost=2. Za dijagonalu (0, 5) udaljenost od vrha 0 iznosi 5, uz lijeva_udaljenost=5 i desna_udaljenost=2.



Anna želi izazvati Jacoba. Jakov uopće ne zna koje su dijagonale nacrtane. Zna samo vrijednost n , ali može Annu više puta pitati za neke parove vrhova i ona će mu reći postoji li dijagonala između tih vrhova. Jacobov je cilj pronaći najbližu (s udaljenošću definiranom kao gore) nacrtanu dijagonalu od vrha 0. Pomoći ćete mu da postigne svoj cilj postavljajući Ani ograničenu količinu pitanja.

Ograničenja

- $5 \leq n \leq 100$

Implementacijski detalji

Trebate implementirati sljedeću funkciju:

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

- Ovu funkciju grader poziva jednom
- n : broj vrhova mnogokuta
- Funkcija treba vratiti dijagonalu između vrhova a i b kao cijeli broj vrijednosti $a \cdot n + b$
- Ako je više dijagonala s minimalnom udaljenošću, možete vratiti bilo koju od njih

Gornja funkcija može upućivati pozive sljedećoj funkciji:

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

- x : oznaka prvog vrha
- y : oznaka drugog vrha
- $0 \leq x, y \leq n - 1$
- vraća 1 ako postoji dijagonala između x i y , a 0 inače

Primjer interakcije

Evo primjera unosa za grader i odgovarajućih poziva funkcija. Ovaj je primjer prikazan na gornjoj slici.

Jedini redak na ulazu ima jedan cijeli broj: n

Grader će ispisati svaki upit na stdout i na njega morate ručno odgovoriti s 1 ili 0.

Primjer ulaza u grader	Primjeri poziva			
	Poziva	Vraća	Poziva	Vraća
7	solve(7)			
			query(0, 3)	
				query vraća 0
			query(0, 5)	
				query vraća 1
			query(1, 5)	
				query vraća 1
		solve vraća $1 \cdot 7 + 5 = 12$		
		Točno!		

Bodovanje

Neka je q broj pitanja koje ste koristili na jednom testnom primjeru. Dodatno, $w = \frac{n \cdot (n-3)}{2}$.

- Ako postavite neispravno pitanje ili pogrešno pogodite, dobit ćete 0% bodova za test
- Ako je $w < q$ dobit ćete 0% bodova za test
- Ako je $n < q \leq w$ dobit ćete $10 + 60 \cdot \frac{w-q}{w-n}$ % bodova za test
- Ako je $q \leq n$ dobit ćete 100% bodova za test

Podzadatci

Postoji samo jedan podzadatak i vaš je rezultat zbroj pojedinačnih rezultata na testnim primjerima. No tijekom natjecanja moći ćete vidjeti rezultate na samo pola testnih primjera (vrijednih 50 bodova). Druga polovina rezultata bit će objavljena nakon natjecanja. Vaš konačni rezultat bit će **najbolji ukupni rezultat od poslanih rješenja**.