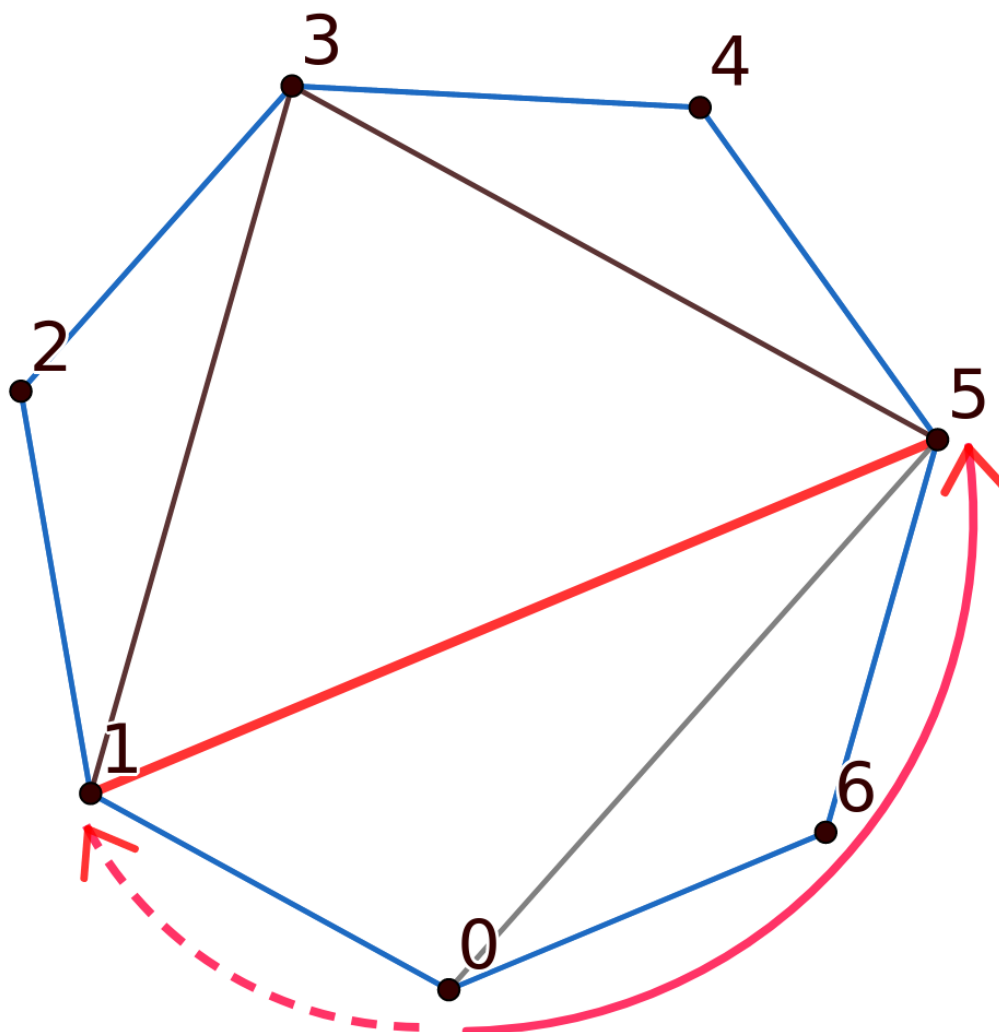


Triangulacija

Naloga

Ana je narisala večkotnik z n oglišči in jih označila s števili od 0 do $n - 1$ v smeri urinega kazalca. Kasneje ga je razdelila na trikotnike tako, da je narisala $n - 3$ diagonal, ki se med seboj ne sekajo, lahko pa se stikajo v oglišču. Diagonala je definirana kot daljica razpeta med dvema ogliščema, ki nimata skupne stranice.

Razdalja med ogliščem A in diagonalo D je definirana tako: Postavimo se v oglišče A in se premikamo po ogliščih v smeri urinega kazalca dokler ne dosežemo enega izmed krajišč diagonale D . Število prepotovanih oglišč imenujemo **leva_razdalja**. Če pa postopek iz oglišča A ponovimo v nasprotni smeri urinega kazalca, prepotovano razdaljo imenujemo **desna_razdalja**. **Razdalja** med A in D je enaka večji izmed vrednosti **leva_razdalja** in **desna_razdalja**.



V sliki primera je razdalja med ogliščem 0 ter diagonalo $(1, 5)$ enaka 2, saj je $leva_razdalja = 1$ in $desna_razdalja = 2$. Za diagonalo $(0, 5)$ je razdalja med ogliščema 0 in 5 enaka 5, saj je $leva_razdalja = 5$ in $desna_razdalja = 2$.

Ana pripravlja izziv za Jakoba. Jakob ne ve ničesar o narisanih diagonalah. Edino kar pozna, je število n . Jakob lahko sprašuje Ano o parih oglišč, ona pa mu pove, ali je med ogliščema narisala diagonalo. Jakobova naloga je, da poišče najbližjo diagonalo (po zgornji definiciji razdalje) oglišču 0. Pomagaj mu rešiti izziv tako, da Ani postaviš nadvse malo vprašanj.

Omejitve

- $5 \leq n \leq 100$

Implementacijski napotki

Implementirati moraš funkcijo:

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

- Ocenjevalnik funkcijo pokliče le enkrat
- n : število oglišč
- Funkcija naj vrne diagonalno med ogliščema a in b kot celo število z vrednostjo $a \cdot n + b$.
- Če je več diagonal z enako najmanjšo razdaljo vrni katerokoli izmed njih.

Zgoraj omenjena funkcija lahko kliče funkcijo:

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

- x : številka prvega oglišča
- y : številka drugega oglišča
- $0 \leq x, y \leq n - 1$
- Funkcija vrne 1, če se obstaja diagonalna med ogliščema x in y , drugače vrne 0.

Primer interakcije

Tukaj je primer vhoda za ocenjevalnik in ustrezni klici funkcij. Ta vhod je prikazan na zgornji sliki.

Na vhodu se nahaja le celo število n .

Vzorčni ocenjevalnik bo napisal vsak poizvedbeni klic na standardni izhod, nanj pa moraš ročno vpisati 1 ali 0.

Primer vhoda ocenjevalniku	Primeri klicev			
	Klic	Vračilo	Klic	Vračilo
7	solve(7)			
			query(0, 3)	
				query vrne 0
			query(0, 5)	
				query vrne 1
			query(1, 5)	
				query vrne 1
		solve vrne $1 \cdot 7 + 5 = 12$		
		Pravilno!		

Ocenjevanje

Naj bo q število vprašanj, ki jih je zastavil tvoj program za rešitev testnega primera. Naj bo $w = \frac{n \cdot (n-3)}{2}$

- Če program vpraša neveljavno vprašanje ali napačno ugiba odgovor, prejmeš 0% za testni primer.
- Če $w < q$, prejmeš 0% za testni primer.
- Če $n < q \leq w$, prejmeš $(10 + 80 \cdot \frac{w-q}{w-n})\%$ za testni primer.
- Če $q \leq n$, prejmeš 100% za testni primer.

Podnaloge

Pri tej nalogi je zgolj ena podnaloga in tvoja ocena je vsota ocen testnih primerov. Med tekmovanjem lahko vidiš rezultat le za polovico testnih primerov (vrednih 50 točk). Druga polovica točk pa bo znana po tekmovanju. **Končna ocena je enaka najboljši oceni med vsemi oddajami.**