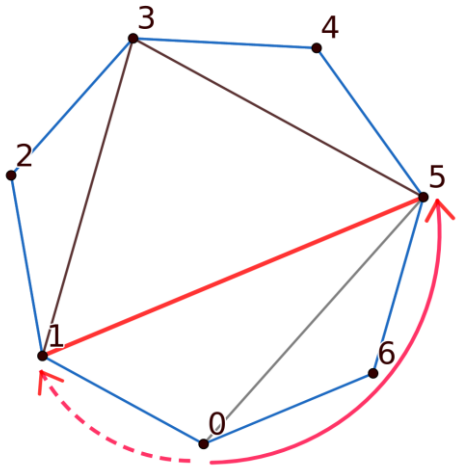
Triangulation(ტრიანგულაცია)

პირობა

ანამ დახატა წესიერი n კუთხედი, რომლის წვეროებიც გადანომრა 0-დან $n-1$ ჩათვლით საათის ისრის მიმართულებით. მოგვიანებით მან გაავლო $n-3$ დიაგონალი ისე, რომ არცერთი მათგანი ერთმანეთს არ კვეთდა(დიაგონალის ბოლოებში შეხების გარდა). რის შედეგადაც მოხდა მრავალკუთხედის ტრიანგულაცია. დიაგონალად არ ითვლება მეზობელ წვეროებს შორის გავლებული წიბო.



მოდით, განვმარტოთ მანძილი ნებისმიერ წვერო A-ს და დიაგონალ D-ს შორის: წარმოვიდგინოთ, რომ A წვეროში ვდგავართ და თითო ნაბიჯზე ვმოძრაობთ მრავალკუთხის წიბოებზე საათის ისრის მიმართულებით, სანამ დიაგონალი D-ს ერთ-ერთ ბოლოს არ მივადგებით. გადადგმული ნაბიჯების რაოდენობა იქნება **left_distance**. ანალოგიურად, **right_distance** არის ნაბიჯების რაოდენობა, როდესაც წიბოებს საათის საწინააღმდეგო მიმართულებით მივუყვებით ისევ D-ს ერთ-ერთ ბოლომდე.

საბოლოოდ, A-ს და D-ს შორის **მანძილი** განვმარტოთ, როგორც **მაქსიმუმი left_distance**-სა და **right_distance**-ს შორის.

ნაჩვენებ სურათში მაძილი წვერო 0-სა და დიაგონალ (1,5)-ს შორის არის 2, სადაც left_distance გამოდის 1, ხოლო right_distance კი 2. დიაგონალ (0,5)-მდე მაძილი კი არის 5 რადგან left_distance არის 5 ხოლო right_distance არის 2.

ანას უნდა რომ იაკობს გამოცანა შეუდგინოს. იაკობმა საერთოდ არ იცის რომელი დიაგონალები არის გავლებული მრავალკუთხედში. მან მხოლოდ იცის n, თუმცა მას შეუძლია ანასთვის კითხვების დასმა. მისი კითხვა შედგება ორი წვეროს ინდექსისგან, რაზეც ანა პასუხობს 1-ით, თუკი ამ ორ წვეროს შორის გავლებული არის დიაგონალი, ხოლო წინააღმდეგ შემთხვევაში კი 0-ით. იაკობის მიზანია, იპოვოს დიაგონალი, რომელიც არის ყველაზე ახლოს(თუ დისტანციას განვმარტავთ როგორც ზემოთ ვახსენეთ) წვეროსთან ნომრით 0. კითხვების რაოდენობა შეზღუდულია. იაკობი თქვენ გთხოვთ დახმარებას.

შეზღუდვები

- $5 \leq n \leq 100$

იმპლემენტაციის დეტალები

თქვენ გამოგზავნილ კოდში უნდა იყოს დაწერილი შემდეგი ფუნქცია:

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

- ეს ფუნქცია მხოლოდ ერთხელ გამოიძახება გრეიდერის მიერ
- *n*: წვეროების რაოდენობა მრავალკუთხედში
- თქვენ უნდა დააბრუნოთ დიაგონალი რომელიღაც a-ს და b-ს შორის. ამისთვის დააბრუნეთ ერთი რიცხვი $a \cdot n + b$
- თუკი მინიმალურ მანძილიანი დიაგონალი რამდენიმე არსებობს მაშინ დააბრუნეთ ნებისმიერი მათ შორის

ზედა ფუნქციას შეუძლია გამოიძახოს შემდეგი ფუნქცია:

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

- *x:* ერთი წვეროს ინდექსი
- y: მეორე წვეროს ინდექსი
- $0 \leq x, y \leq n - 1$
- ფუნქცია აბრუნებს 1-ს თუკი ამ ორ წვეროს შორის დიაგონალი არსებობს ხოლო 0-ს წინააღმდეგ შემთხვევაში

სამაგალითო ინტერაქცია

ქვემოთ მოცემულია სამაგალითო შემავალი მონაცემები გრეიდერისთვის.
ერთადერთ ხაზში უნდა შეუტანოთ ერთი რიცხვი n.
სამაგალითო გრეიდერი დაბეჭდავს ყველა query-ის გამოძახებას და თქვენ 1 ან 0 უნდა შეუყვანოთ იმის მიხედვით, დიაგონალი არსებობს თუ არა.
ეს ინტერაქცია შეესაბამება ზემოთ მოცემულ სურათს.

Sample Input to grader	Sample Calls			
	Calls	Returns	Calls	Returns
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!		

შეფასება

თქვენ მიერ გამოყენებული კითხების რაოდენობა აღვნიშნოთ როგორც q. ხოლო $w = \frac{n \cdot (n-3)}{2}$.

- თქვენ თუ ცუდი ფორმატით იკითხავთ კითხვას ან არასწორ პასუხს გასცემთ მაშინ მიიღებთ ტესტის ქულების 0%-ს
- თუ $w < q$ თქვენ მიიღებთ ტესტის ქულების 0%-ს
- თუ $n < q \leq w$ თქვენ მიიღებთ ტესტის ქულების $10 + 60 \cdot \frac{w-q}{w-n}$%-ს
- თუ $q \leq n$ თქვენ მიიღებთ ტესტის ქულების 100%-ს

ქვეამოცანები

ამ ამოცანაში არის მხოლოდ ერთი ქვეამოცანა, რომელშიც თქვენი ქულა იქნება ცალკეული ტესტების ქულების ჯამი. თუმცა კონტესტის განმავლობაში თქვენ დაინახავთ ქულას მხოლოდ ტესტების ნახევარზე(მაქსიმუმ 50 ქულა). დანარჩენი ტესტების ქულებს გაიგებთ კონტესტის დამთავრების შემდგომ. **საბოლოო ქულა ამოცანაში იქნება საუკესო ქულა ყველა გაგზავნას შორის.**