

Triangulation

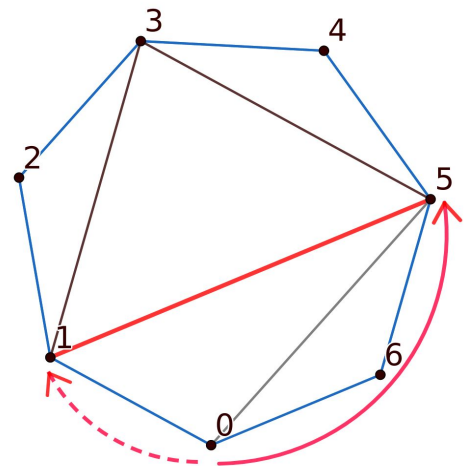
Enoncé

Anna a tracé un polygone régulier avec n sommets numérotés de 0 à $n - 1$ dans l'ordre des aiguilles d'une montre. Ensuite, elle l'a triangulé en traçant $n - 3$ diagonales qui ne s'intersectent pas, sauf éventuellement en leurs extrémités. Une diagonale est un segment qui relie deux sommets qui ne sont pas les extrémités d'un même côté du polygone.

Définissons d'abord la distance d'un sommet A à une diagonale D . Supposons qu'on commence au sommet A et qu'on passe au sommet suivant dans le sens des aiguilles d'une montre jusqu'à atteindre une extrémité de D . Le nombre de côtés traversés sera appelé **distance_gauche**.

De même, on nomme **distance_droite** le nombre de côtés traversés si on part de A et qu'on se déplace dans le sens inverse des aiguilles d'une montre jusqu'à atteindre D . La **distance** de A à D est le **maximum** entre **distance_gauche** et **distance_droite**.

Dans l'exemple illustré, la distance du sommet 0 à la diagonale $(1,5)$ est 2 avec **distance_gauche** qui vaut 1 et **distance_droite** qui vaut 2 . Pour la diagonale $(0,5)$ la distance depuis le sommet 0 est 5 , avec **distance_gauche** = 5 et **distance_droite** = 2 .



Anna veut mettre en difficulté Jacob. Jacob ne sait pas du tout quelles diagonales sont tracées. Il connaît uniquement la valeur de n , mais il peut demander à Anna plusieurs fois si une diagonale est tracée entre une paire de sommets. L'objectif de Jacob est de trouver la diagonale tracée la plus proche du sommet 0 (avec la distance définie ci-dessus). Vous devez l'aider à accomplir cet objectif en posant un nombre limité de questions à Anna.

Contraintes

- $5 \leq n \leq 100$

Détails d'implémentation

Vous devez implémenter la fonction suivante dans votre soumission :

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

- Cette fonction est appelée exactement une fois par l'évaluateur.
- n : nombre de sommets du polygone
- Cette fonction doit renvoyer la diagonale entre des sommets a et b sous la forme d'un entier de valeur $a \cdot n + b$
- S'il y a plusieurs diagonales à la distance minimale, vous pouvez afficher n'importe laquelle d'entre elles.

La fonction ci-dessus peut appeler la fonction suivante :

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

- x : le numéro du premier sommet
- y : le numéro du deuxième sommet
- $0 \leq x, y \leq n - 1$
- renvoie 1 si il y a une diagonale entre x et y et 0 sinon

Exemple d'interaction

Voici un exemple d'entrée pour l'évaluateur et les appels de fonctions correspondant. Cette entrée correspond à l'illustration précédente.

L'unique ligne de l'entrée contient un entier : n

L'évaluateur d'exemple va afficher chaque appel de requête sur la sortie standard (stdout) et vous devez y répondre manuellement avec 1 ou 0.

Exemple d'entrée pour l'évaluateur	Exemples d'appels			
	Appels	Renvois	Appels	Renvois
7	solve(7)			
			query(0, 3)	
				query renvoie 0
			query(0, 5)	
				query renvoie 1

			query(1, 5)	
				query renvoie 1
		solve renvoie $1 \cdot 7 + 5 = 12$		
		Correct !		

Score

Notons q le nombre de questions utilisées pour un test et $w = \frac{n(n-3)}{2}$.

- Si vous posez une question invalide ou que votre solution est incorrecte, vous recevrez 0% des points pour le test.
- Si $w < q$ vous recevrez 0% des points pour le test
- Si $n < q \leq w$ vous recevrez $10 + 60 \cdot \frac{w-q}{w-n}$ % des points pour le test
- Si $q \leq n$ vous recevrez 100% des points pour le test

Sous-tâches

Il y a une unique sous-tâche et votre score est la somme des scores individuels des tests. Toutefois pendant l'épreuve vous ne pourrez voir les scores que pour la moitié des tests (valant 50 points). L'autre moitié du score sera révélée après l'épreuve. Votre score final sera le **meilleur score total parmi toutes les soumissions**.