

Fonctions à appeler

Vous pouvez appeler deux fonctions pour localiser votre chèvre :

- `horizontal_information(x)` : cette fonction renvoie -1 si votre chèvre est à l'ouest de vous, 1 si votre chèvre est à l'est de vous et 0 si votre chèvre est sur la même colonne que vous sur la grille.
- `vertical_information(y)` : cette fonction renvoie -1 si votre chèvre est au sud de vous, 1 si votre chèvre est au nord de vous et 0 si votre chèvre est sur la même ligne que vous sur la grille.

Ces fonctions peuvent être appelées **au plus 120 fois au total** (après cela, le soleil se couche).

Une fois que vous avez localisé votre chèvre, vous devez appeler `found_goat(x, y)` une fois avec les coordonnées que vous avez trouvées.

Fonctions à implémenter

`find_goat(N, M)` : cette fonction reçoit les dimensions de la grille comme paramètres. Vous devez appeler les fonctions `horizontal_information` et `vertical_information` pour localiser votre chèvre. Une fois que vous avez trouvé votre chèvre, vous devez appeler `found_goat` une fois avec les coordonnées de votre chèvre.

Remarque : en Java, il faut appeler `grader.horizontal_information(x)`, `grader.vertical_information(y)` et `grader.find_goat(N, M)`, se référer au squelette de code fourni.

Contraintes supplémentaires

Sous-tâche	Points	Contraintes
A	20	$N = 120, M = 1$
B	20	$N = M = 60$
C	20	$N = 10^9, M = 1$
D	40	$N = M = 10^9$

Exemple 0

Voici une séquence d'évènements possibles pour une grille 4×5 ($N = 5$ et $M = 4$).

Votre appel	Réponse reçue
horizontal_information(3)	1
horizontal_information(5)	-1
vertical_information(4)	-1
horizontal_information(4)	0
vertical_information(2)	0
found_goat(4, 2)	Chèvre trouvée!

Informations de compilation

Cette tâche est interactive. Vous ne pouvez pas utiliser la compilation et l'exécution automatique de `gedit`. Suivez les étapes suivants pour compiler et exécuter votre programme. Demandez de l'aide à un responsable si nécessaire.

- Ouvrez un terminal (l'application LXTerminal)
- Pour afficher la liste des fichiers présents dans un dossier, utilisez la commande `ls`
- Pour changer de dossier courant (vers le dossier contenant les codes à modifier), utilisez la commande `cd nom.du.dossier`
- Pour ceux qui utilisent C++, les commandes sont les suivantes :
 - Compilation :
`g++ -std=c++11 -Wall -Werror -Wshadow grader.cpp goat.cpp`
 - Exécution : `./a.out < input.txt`
- Pour ceux qui utilisent Java, les commandes sont les suivantes :
 - Compilation : `javac grader.java`
 - Exécution : `java grader < input.txt`

Le résultat de votre programme sera affiché à la console : « FOUND THE GOAT AFTER X CALLS » si votre chèvre a été trouvée ou « WRONG COORDINATES WERE GIVEN » si vous fournissez des coordonnées incorrectes. D'autres messages peuvent être affichés si les arguments que vous donnez aux autres questions sont incorrects. Si vous voulez modifier le fichier `input.txt`, sachez que la première ligne contient N et M et que la seconde ligne contient les coordonnées de la chèvre (entre 1 et N et 1 et M respectivement).

Remarques

- Vous devez seulement soumettre le fichier `goat.cpp` ou `goat.java`. Il doit implémenter la fonction `find_goat` décrite précédemment.
- **N'affichez rien à la console dans le programme que vous soumettez (`printf`, `cout`).** Vous pouvez utiliser `cerr` en C++ et `System.err.println` en Java pour afficher de l'information de debug.