

Ne vous connectez pas avec votre mot de passe habituel !

☞ Utiliser l'identifiant `interro396` et le mot de passe `blabla`

## Contexte (à lire si vous voulez)

Après une analyse minutieuse du comportement des financiers de la planète, un constat évident s'impose : leur confier de l'argent est dangereux pour tous et malsain pour leur santé mentale.

Les robots-traders doivent donc apprendre à vivre sans argent. Tout contact avec une somme d'argent aggrave désespérément leur cas et s'avère néfaste pour le reste du monde.

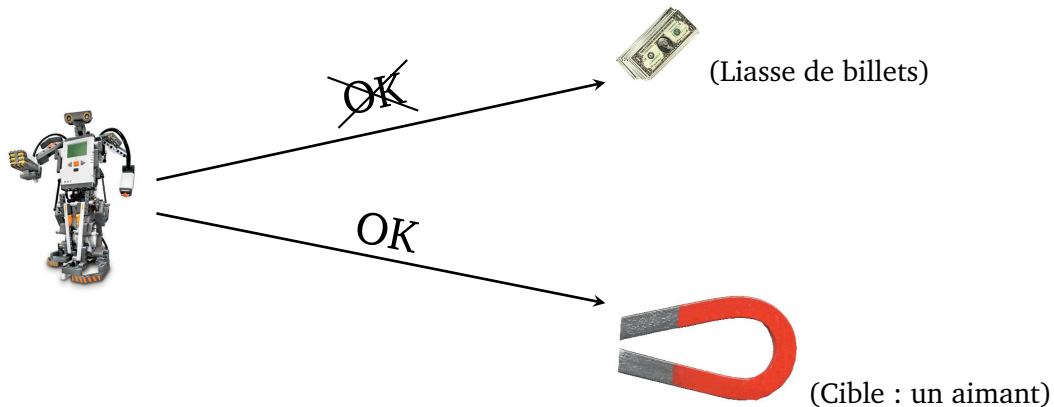
Vous allez maintenant aider les braves robots-trader à rentrer dans le droit chemin.



## Consignes

### Consignes techniques

- Vous devez remplir trois missions. Chaque mission consiste en un plateau sur lequel se situent des robots, des liasses de billets, et des cibles.
- Lors des déplacements des robots, il faut toujours éviter les liasses (seule la position des pieds du robot est importante).



### Consignes administratives

- Le barème donné est approximatif, il pourra être ajusté.
- Les missions sont indépendantes, mais la difficulté est croissante. Commencez par la mission 1.
- PENSEZ À METTRE VOTRE NOM AU DÉBUT DE CHAQUE PROGRAMME

### Principaux critères d'évaluation

- Le programme compile sans erreur et remplit correctement la mission.
- Le code est indenté (touche TAB) et contient des commentaires judicieux et dosés.
- L'algorithme est le plus simple possible.

---

# SUJET N°396 (verso)

---

## Mission 1    ★ ★    6,5 points

- Ouvrez votre dossier personnel (icône de la maison) ; ouvrez le fichier mission1b.adb, compilez-le.
  - Pour exécuter le programme, lancer d'abord ecoload-exe et choisissez l'exécutable que vous venez de compiler (comme en TP).
  - Observez la situation, il y a un chemin en forme d'escalier. Attention, le nombre et la longueur des "marches" de l'escalier est aléatoire : exécuter une deuxième fois le programme, et constatez que le chemin est différent.
  - Examinez l'acteur EcoRobot sur commetud (1ere Annee, ADA, Acteurs).
  - **Mission** : L'objectif est de faire atteindre la cible aux deux robots, en respectant les étapes suivantes :
    - ▷ Complétez le programme mission1b.adb avec une procédure `Jump` qui reçoit en argument un numéro de robot et une direction, et qui déplace le robot dans cette direction.
    - ▷ Écrivez une procédure de test `Test_Jump` **sans argument** qui fait bouger chaque robot successivement au sud, puis à l'ouest, au nord, puis à l'est.
    - ▷ En vous inspirant de `Jump`, écrivez une fonction `Safe_Jump` qui reçoit un numéro de robot et une direction et qui fait bouger le robot seulement si la case de destination est sûre (elle ne contient pas de billet et n'est pas en dehors du plateau). Cette fonction renvoie Vrai si le robot a bougé, Faux sinon. Inutile de tester cette fonction, car elle sera testée directement avec la question suivante.
    - ▷ Écrivez une procédure `Progress` qui reçoit un numéro de robot et une direction, et qui avance autant que possible dans cette direction. Écrivez une procédure de test `Test_Progress` sans argument, dans le style de `Test_Jump`.
    - ▷ Terminez la mission en amenant tous les robots sur la cible.
- 

## Mission 2    ★ ★    7,5 points

- Ouvrez le fichier mission2b.adb, compilez-le, exécutez-le, observez la situation.
  - Pour cette mission, la position des robots est aléatoire.
  - **Mission** : L'objectif est de déplacer les robots selon une chorégraphie précise :
    - ▷ Écrire une fonction `Num_Danseur` qui renvoie le numéro d'un robot mécontent ayant le moins de voisins. Un voisin est un robot situé sur la case adjacente (mais pas en diagonale).
    - ▷ Écrivez une procédure de test de cette fonction.
    - ▷ En écrivant si besoin de nouvelles procédures ou fonctions, terminer la mission en respectant la consigne suivante :
      - Le robot qui effectue sa chorégraphie en premier est celui qui a le moins de voisins.
      - La chorégraphie consiste à aller sur un bord du plateau (par exemple le bord Nord), puis à revenir à sa place (conseil : compter les pas), et à recommencer avec les autres bords. L'ordre dans lequel les bords doivent être visités est indiqué par les **préférences** de chaque robot (bien lire l'acteur).
      - Lorsque la chorégraphie est entièrement terminée, le robot peut rejoindre la cible, qui est toujours au même endroit. (Penser à cliquer sur le mode éclair pour aller plus vite pendant l'exécution).
      - Recommencer avec les robots restants, celui qui commence étant de nouveau celui qui a le moins de voisins. Pendant la chorégraphie, des robots peuvent en pousser d'autres, c'est normal.
- 

## Mission 3    ★ ★ ★ ★    6 points

- Ouvrez le fichier mission3a.adb, compilez-le, exécutez-le, observez la situation.
- Faites sortir le robot du labyrinthe (qui est aléatoire) avec l'algorithme de votre choix. Examinez bien l'acteur EcoRobot, il contient des fonctions utiles. (La sortie est la cible).
- Si vous n'avez pas le temps de programmer votre algorithme, écrivez-le en commentaires dans votre programme Ada, de manière aussi claire et détaillée que possible.