

DS 1

Informatique pour tous, deuxième année

Julien REICHERT

Durée : 2 heures maximum.

Exercice 1 : Réaliser une implémentation de la structure de pile : expliquer la structure choisie et écrire les trois fonctions de base associées, ainsi que les fonctions `sommet`, `est_vide` et `taille`.

Exercice 2 : D'après la formule du cours, quelle est la complexité c_n d'un algorithme en fonction de la taille n de son entrée si cette complexité s'obtient par la formule de récurrence $c_n = c_{n/2} + 1$? Donner un exemple d'algorithme ayant cette complexité.

Exercice 3 : Écrire un programme qui, étant donné une liste d'entiers représentant une expression utilisant diverses parenthèses ouvrantes et fermantes, revoie la liste des couples correspondant aux positions des ouvertures et fermetures des paires de parenthèses associées. Par convention, l'entier positif i correspond à une parenthèse ouvrante de type i , l'entier négatif $-i$ correspond à une parenthèse fermante de type i , et l'entier 0 correspond à autre chose qu'une parenthèse et qui est donc à ignorer. La fonction doit renvoyer un message d'erreur si la liste ne correspond pas à un bon parenthésage.

La fonction, prenant comme argument la liste `[1,0,2,0,-2,3,-3,-1]`, renverra alors `[(0,7),(2,4),(5,6)]`. L'ordre d'apparition des couples n'est pas important pour cette question.

Exercice 4 : Écrire un programme qui, étant donné une configuration intermédiaire possible dans le problème des tours de Hanoï, imprime la liste des opérations nécessaires pour arriver à une configuration finale (peu importe sur lequel des trois piquets les anneaux seront placés). Les arguments de la fonction seront alors trois listes décroissantes contenant à elles trois une et une seule fois tous les nombres de 0 à un certain entier naturel n . Il est essentiel de bien voir comment lancer la récursion. On ne demande pas que la séquence d'opérations soit optimale.

Exercice 5 : Robozzle.

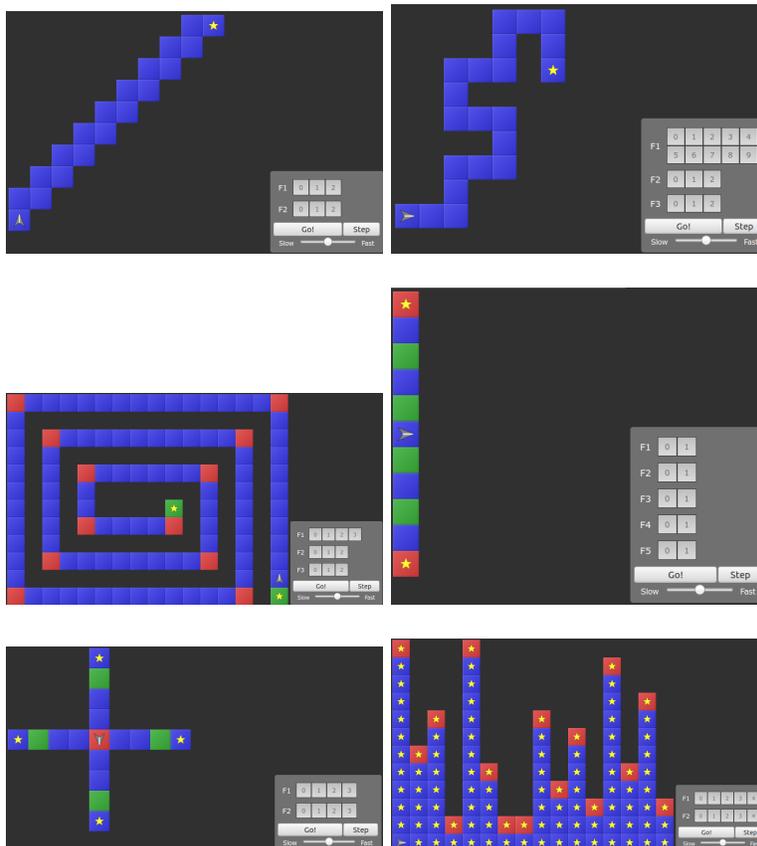
Une instance de Robozzle est un plateau de jeu contenant des cases de trois couleurs différentes : des bleues, des rouges et des vertes, certaines de ces cases étant marquées d'une étoile, d'un robot présent sur une des cases avec une orientation quelconque parmi les quatre directions cardinales, et d'une liste de fonctions à remplir avec pour chaque fonction une limite d'instructions. Les instructions possibles sont de la forme `si condition alors action`, avec 4 conditions possibles : vrai, la case du robot est bleue, la case du robot est rouge, la case du robot est verte. Les actions possibles sont l'avancée d'une case, un virage à gauche sur place, un virage à droite sur place et un appel récursif à n'importe laquelle des fonctions. Une instance est résolue si on parvient à renseigner les fonctions disponibles (pas nécessairement toutes) dans la limite de la place disponible de sorte que le robot passe par toutes les cases marquées d'une étoile (ce qui interrompt immédiatement l'exécution des instructions) sans jamais sortir du plateau (par exemple en avançant à un endroit où il n'y a pas de case). L'exécution doit démarrer sur F1, en outre.

À titre d'exemple, l'instance suivante peut se résoudre en fournissant, avec une fonction de 5 instructions disponible :

F1 : si vrai alors avancer (qu'on écrira simplement « avancer »), avancer, tourner à gauche, si la case du robot est verte alors tourner à droite (qu'on écrira simplement « droite vert ») et répéter F1.



Résoudre les six autres instances du jeu ci-dessous.



Résoudre la dernière instance signifie que les piles et la récursivité sont pleinement maîtrisées...