DEUG MIAS 2e année
Projet de programmation fonctionnelle
Le compte est bon

Pierre Boulet Jean-Marc Nicod Frédéric Vivien

Le but de ce projet est de programmer le jeu du « compte est bon».

Dans ce jeu, l'ordinateur sélectionne 6 cartes sur lesquelles sont inscrits des nombres. On dispose de l'ensemble des cartes suivantes :

- chaque nombre entre 1 et 10, en 2 exemplaires ;
- 1 exemplaire des nombres 25, 50, 75 et 100.

L'ordinateur choisit aussi au hasard un nombre de 3 chiffres entre 101 et 999. Le but du jeu est de trouver une suite d'opérations composées des nombres présents sur les cartes sélectionnées et des 4 opérations : addition, soustraction, multiplication et division.

1. Tirage du problème à résoudre

Question 1. *Construisez une fonction objectif qui retourne un nombre entre 101 et 999.*

Indication. On dispose de la fonction (*random n*) qui retourne un entier compris entre 0 et $n-1$.

Question 2. *Construisez une fonction tirage qui retourne une liste de 6 nombres représentant les 6 cartes sélectionnées.*

Note. Pensez aux restrictions sur l'ensemble des cartes possibles.

2. Recherche d'une solution

On va construire ici pas à pas le programme de recherche exhaustive d'une solution. L'idée de ce programme est la suivante :

> On manipule une liste d'expressions arithmétiques. Au départ, cette liste contient les nombres retournés par *tirage*. À chaque étape de l'algorithme, on remplace

deux expressions de cette liste par une autre expression construite à partir de celles-ci et d'une opération.

Note. Un nombre est une expression arithmétique.

Question 3. *Comme on manipule des expressions et qu'on veut les comparer à l'objectif, il faut pouvoir les évaluer pour calculer le nombre qu'elles représentent. Construisez une telle fonction d'évaluation : evalue.*

Exemple. (*evalue* '(* (+ 2 5) (− 25 10))) ⟹105

Question 4. *On veut pouvoir tester toutes les solutions. Pour cela, à chaque étape, il faut construire la liste de toutes les listes d'expressions déductibles de la liste d'expressions courante. Construisez une fonction nouvelles-expressions qui prend une liste d'expressions et retourne la liste de toutes les listes d'expressions construites par remplacement de deux expressions de la liste originale par l'application d'une opération arthmétique à ces deux expressions.*

Note. Toutes les opérations sont symétriques : en effet, l'addition et la multiplication commutent, on soustrait toujours le plus petit nombre du plus grand et on divise le plus grand nombre par le plus petit et seulement si la division tombe juste.

Question 5. *À partir des deux fonctions précédentes, construisez une fonction resout qui prend en argument une liste d'expressions et un nombre (objectif) et retourne une expression s'évaluant en ce nombre construite à partir des expressions de la liste argument si une telle expression existe et* "pas de solution" *sinon.*

Indication. Une solution peut ne pas utiliser toutes les cartes.

3. Extensions

Vous devez répondre à au moins une des questions suivantes.

Question 6. *Proposez une modification de votre programme qui retourne une expression pour la valeur la plus proche de l'objectif.*

Question 7. *Optimisez votre recherche en diminuant l'espace de recherche : on peut supprimer toutes les solutions qui font les calculs suivants : multiplier ou diviser par* 1*, la soustraction de deux valeurs identiques et* 2×2 *(car on peut faire* $2 + 2$*). On diminue ainsi l'espace de recherche (et donc le temps de recherche) d'environ un tiers.*

Question 8. *Optimisez votre programme en évitant les appels inutiles à la fonction evalue.*

4. Planning

Vous disposez de trois semaines pour faire ce projet. Il est à rendre le 23 ou le 25 avril selon les groupes. Une attention toute particulière sera portée à la clarté du code (commentaires et indentation) et de la rédaction. Vous pouvez faire ce projet seul ou à deux.

Les documents à rendre sont :

- une disquette avec le code scheme de votre programme ;
- un rapport d'explications de 2 à 3 pages maximum (sur papier).