Université Paris Diderot – Paris 7
Concepts informatiques (CI2)
Examen du 21 juin - Durée : 2 heures 30 minutes

Exercice 4. On dispose d'un type $Pile$ avec les méthodes *empiler* (*push*), *dépiler* (*pop*) et *estVide*.

4.1. Écrire une méthode *affiche1* qui prend une pile en argument et affiche son contenu sans la modifier, l'affichage commençant par ce qui se trouve au sommet.

4.2. Écrire une méthode *affiche2* qui réalise l'affichage en commençant par le fond.

Exercice 5.

Vous disposez d'une somme que vous souhaitez dépenser en totalité pour acquérir un certain nombre de produits dont le prix est particulièrement attractif : il y a N types de produits, tous ont un prix différent et le nombre d'exemplaire de chaque type est illimité. Vous vous autorisez à acheter un nombre quelconque d'exemplaires de chacun des N produits.

5.1. Vous souhaitez tout d'abord connaître le nombre de combinaisons possibles (sans les énumérer) vous permettant de dépenser exactement la somme .

Exprimer sous forme récursive un procédé de calcul de ce nombre et en donner la traduction en java. On supposera les prix des produits donnés sous forme triée par prix croissant dans un tableau d'entiers *prix*.

5.2. Exécuter à la main le programme précédent pour 4 types d'articles de prix respectifs 3, 4, 5 et 7 et $= 14$.

5.3. Pourquoi ce problème peut-il être résolu par *backtracking* ?

5.4. Exprimer en quoi consisterait une stratégie de résolution de ce type (solution partielle, contrainte d'allongement d'une solution, ...) et dessiner l'arbre décrivant la suite de solutions partielles générées et toutes les solutions possibles.