

Algorithmique et Programmation

Licence Sciences, Santé et Technologies mention Sciences et Technologies

Semestre 2

Vendredi 13 mars 2015

13h30 - 15h

Les implantations demandées dans cette épreuve sont à réaliser en langage algorithmique à l'exception des questions 3a et 3c pour lesquelles les codages en langage Java et en langage algorithmique sont demandés.

Le seul document autorisé est la fiche PDL-Java.

Exercice n°1

- Dans quelle mesure est-il possible de développer un type agrégé dont un ou plusieurs champs sont eux-mêmes de type agrégé?
- Quel est le rôle des paramètres d'entête d'un sous-algorithme?
- De quels types peuvent être définis les composantes élémentaires d'un tableau?
- Un tableau étant défini de taille N entière, quel est l'intervalle de validité des indices de ses composantes?

Exercice n°2

Lorsque l'on génère une série de 20 valeurs booléennes calculées individuellement avec la probabilité p d'obtenir un "vrai", le nombre nbv de vrais comptabilisés sur cette série peut être 0, 1, 2, 3, 4, ..., 20. On souhaite analyser la distribution des valeurs nbv sur un échantillon de 1000000 de séries de 20 booléens. On souhaite donc calculer combien de fois 0 vrai, combien de fois 1 vrai, combien de fois 2 vrais, ..., combien de fois 20 vrais ont été obtenus.

Développer sous la forme d'un **algorithme principal** faisant appel à des **sous-algorithmes de résolution de problème élémentaire** le programme informatique de réalisation des traitements suivants:

- Déclaration d'un tableau tnbv de 21 entiers.
- Initialisation de ce tableau avec des 0.
- Génération de 1000000 de séries de 20 booléens tirés au sort avec la probabilité p d'obtenir la valeur "vrai" (p acquise au clavier dans l'algorithme principal). Pour générer des nombres aléatoires, on pourra utiliser la fonction **réel random()** qui retourne un réel tiré au sort avec équiprobabilité entre 0.0 et 1.0.
- Concurremment à la génération de chacune des 1000000 séries, calcul, pour chacune d'elles, du nombre nbv de "vrai" qu'elle contient et comptabilisation (addition de 1) de cette valeur dans tnbv.
- Affichage du tableau tnbv.

Exercice n°3

Un porte-monnaie peut contenir des pièces de 1, 2, 5, 10, 20, 50 centimes et des pièces de 1 et 2 euros. On souhaite modéliser et manipuler des porte-monnaie dans une application informatique.

- Définir un type agrégé **porteMonnaie**. Donner les codages en langage algorithmique et en langage Java.

On s'intéresse au problème consistant à déterminer si, pour une somme donnée sm comptée en euros et centimes d'euros, il existe une combinaison de tout ou partie des pièces contenues dans un **porteMonnaie** donné pm telle que le total des pièces de cette combinaison est égal à sm. Il est possible que 0, 1 ou plusieurs combinaisons existent suivant les valeurs respectives de sm et pm. Dit autrement, le problème consiste à déterminer si le contenu d'un porte-monnaie permet de rendre exactement une certaine somme.

- Décrire sans l'implanter informatiquement une technique utilisable pour tester si, une somme sm comptée en euros étant donnée, il existe une combinaison de tout ou partie des pièces contenues dans un **porteMonnaie** lui aussi donné telle que le total des pièces de cette combinaison est égal à sm.

- Définir l'entête (pas d'implantation du corps) du sous-algorithme que vous créeriez pour calculer, si elle existe, l'une des combinaisons de pièces dont l'existence est testée à la question (b). Donner les codages en langage algorithmique et en langage Java.

- Développer entièrement un sous-algorithme permettant, si c'est possible en fonction des pièces qui y sont présentes, de retirer une somme exacte donnée d'un **porteMonnaie** donné.