Université Bordeaux 1 Licence Informatique 2005/2006

INF102 Initiation à l'algorithmique

Corrigé du Devoir surveillé

Durée : 1h20. Aucun document autorisé. Les algorithmes qu'on vous demandera d'écrire dans les exercices suivants doivent utiliser le langage EXALGO défini en cours.

Exercice 1.1

Question de cours : Quelle est l'instruction obligatoire dans une fonction récursive ? Pourquoi ?

La récursivité consiste à remplacer une boucle par un appel à la fonction elle-même. Afin d'éviter de générer un mécanisme qui tourne indéfiniment il est impératif de prévoir au moins un cas d'arrêt certains des appels recursifs dans l'algorithme.

Quel est le danger au niveau complexité d'une fonction récursive ? Pourquoi ?

Si le ou les cas d'arrêt sont oubliés on génére un algorithme qui tourne indéfiniment. Même chose si les appels successifs n'aboutissent jamais aux cas d'arrêt. Autre problème lié à la complexite des fonctions récursives : la complexité d'une fonction récursive se mesure en comptant le nombre des appels or dans certains cas un appel à la fonction en génère systématiquement deux autres (ou plus) dans ces cas la fonction croît de façon exponentielle et l'algorithme est inutilisable car non efficace en temps de calcul. Il est alors nécessaire de trouver une version itérative au problème posé (exemples : calcul des nombres de Fibonaci, calcul des coefficients binomiaux)

Exercice 1.2

Soit la structure *Etudiant* suivante :

```
Etudiant = structure
         Nom: chaine de caracteres;
         NoteEpreuve1: entier;
         NoteEpreuve2: entier;
         Moyenne : reel;
       finStructure
```

avec

- $0 \leq NoteEpreuve1 \leq 20$,
- $0 \leq NoteEpreuve2 \leq 20$.

1. Soit *ZEP : tableau [1..NMAX] d'étudiants.* Supposons que ce tableau contiennent les 6 éléments suivants : *Hugo, Jean-Claude, Manu, Nadia, Titeuf, Vomito* classés par ordre alphabétique de leur *Nom* :

   Compléter le tableau suivant :

Description	Nom de la variable	Type de la variable
Moyenne d'Hugo	ZEP[1].Moyenne	reel
Element "Nadia"	ZEP[4]	Etudiant
Note de l'épreuve1 de Nadia	ZEP[4].NoteEpreuve1	entier

2. Soit T un tableau de N étudiants. Ecrivez une fonction qui compte le nombre d'étudiants ayant obtenu une note supérieure ou égale à 10/20 à au moins une des deux épreuves.

```
fonction Reussi1sur2 (ref T : tableau[1..N] d'Etudiant):entier;

var i, compt: entier;

debut
    compt = 0;
    pour i allant de 1 à n par pas de 1 faire :
        si (T[i].NoteEpreuve1 >= 10 ou T[i].NoteEpreuve2 >= 10)
        faire :
        compt = compt+1;
        finsi
    finpour
    retourner (compt)
fin
finfonction
```

3. Ecrivez une fonction d'en-tête *reussi (ref T : tableau[1..N] d'Etudiants, val numero : entier) : booleen* qui renvoie VRAI quand les 2 notes de l'Etudiants sont supérieures ou égales à 10/20 et FAUX sinon. *numero* est l'indice de l'étudiant dans le tableau T.

```
fonction reussi (ref T: tableau[1..N] d'Etudiants,
                val numero : entier): booleen;
debut
    retourner (T[numero].NoteEpreuve1>=10 et
    T[numero].NoteEpreuve2>=10)
fin

finfonction
```

4. En supposant que chaque étudiant dispose de 2 notes, écrivez une fonction qui calcule la moyenne pour chacun d'eux.

```
fonction Moyenne (ref T: tableau [1..N]d'Etudiant):vide;

var i : entier;

debut
    pour i allant de 1 à N par pas de 1 faire :
        T[i].Moyenne = (T[i].NoteEpreuve1+T[i].NoteEpreuve2)/2
    finpour
fin

finfonction
```

Exercice 1.3

```
fonction mystere (val y : entier):entier
var s: entier;
debut
   s=0
   tantque y>0 faire :
       s=s+1;
       y=y/3; /* division entiere */
   fintantque
   retourner(s);
fin
finfonction
```

Que retourne la fonction *mystere* pour y=90 ?
La fonction retourne 5
Quel est le résultat de cette fonction dans le cas général ?
La fonction retourne s tel que $3^{(s-1)} <= y < 3^s$ *soit le plus petite puissance de 3 strictement supérieure à y.*
Quelle est sa complexité, justifier ?
La complexité de cette fonction depend du nombre de fois où la boucle `tantque` *sera effectuée c'est à dire* `s`*. En l'exprimant par rapport à y la complexité est donc de l'ordre de* $ln(y)/ln(3)$ *ou en négligeant la constante de l'ordre de* $ln(y)$

Exercice 1.4

On considère le tableau $T012 = \{1, 0, 2, 2, 0, 1, 0, 2, 1\}$ et la fonction `Que_ fais_ je` définie par :

```
fonction Que_fais_je (ref T : tableau [1..NMAX] d'entiers,
                        val n : entier) : vide;
var i, j, k : entier;
debut
  i=1; j=1; k=n;
  Tant que (i<=k) faire:
    si T[i]==0 alors :
        echanger(T[i],T[j]);
        j=j+1;
        i=i+1;
    sinon
        si T[i]==2 alors :
            echanger (T[i],T[k]);
            k=k-1;
        sinon :
            i=i+1;
        finsi
    finsi
  fintantque
fin
finfonction
```

1. Quel est le résultat de l'appel `Que_ fais_ je(T012, 9)` ?
   $T012 = \{0, 0, 0, 1, 1, 1, 2, 2, 2\}$
2. Dans le cas général si `T` est un tableau de `n` entiers dont les valeurs sont des $\{$`0, 1 ou 2`$\}$ quel est le résultat de l'appel `Que_ fais_ je( T, n)` ?

   La fonction trie le tableau en séparant les 0 des 1 et des 2. Après l'exécution de la fonction le tableau est organisé de la façon suivante : une zone regroupant les 0 au début du tableau, suivie d'une zone de 1 et enfin une zone de 2 à la fin du tableau. Quelle est la complexité de cette fonction ?

   Chaque élément du tableau n'est examiné qu'une fois et l'algorithme s'arrête lorsque tous les élements ont été étudiés. La complexité est donc lineaire par rapport au nombre d'éléments dans le tableau soit de l'ordre de n.