

Algorithmes et Structures de Données

Mardi 17 Novembre 2015

Durée 2H – Cours et TD NON autorisés

Exercice 1 : Pile (5 pts)

```

Fonction f (x : entier) : entier
Var res : entier
Début
Si x>100
    Alors res←x-10
    Sinon res←f(f(x+11){@1}){@2}
FinSi
Retourne(res)
Fin

```

Simulez la pile pour l'appel de f(94) - 15 appels

```

@2, x=101 res=91
@1, x=111 res=101
@2, x=100 res=91
@1, x=110 res=100
@2, x=99 res=91
@1, x=109 res=99
@2, x=98 res=91
@1, x=108 res=98
@2, x=97 res=91
@1, x=107 res=97
@2, x=96 res=91
@1, x=106 res=96
@2, x=95 res=91
@1, x=105 res=95
@0, x=94 res=91

```

Que fait cette fonction ?

- Si $x \geq 101$ alors $f(x) = x - 10$
- Si $0 \leq x \leq 100$ alors $f(x) = 91$

Exercice 2 : Récursivité (5 pts)

On souhaite écrire une **fonction récursive** qui effectue la recherche dichotomique d'un entier dans un tableau ordonné d'entiers. Cette fonction renvoie la position de l'entier dans le tableau s'il est présent, 0 sinon.

2.1. Expliquez en français le principe récursif de cette méthode en précisant les variables sur lesquelles porte la récursion et les critères d'arrêt.

Le principe récursif de cette fonction est de comparer l'élément cherché avec celui se trouvant au milieu du tableau. Si l'élément cherché est celui du milieu, alors la recherche s'arrête en renvoyant

l'indice de l'élément, s'il est plus petit, la recherche se poursuit sur la partie gauche du tableau (appel récursif) et sinon, sur la partie droite du tableau (appel récursif). Les variables de récursion sont alors les bornes entre lesquelles on cherche dans le tableau. Il y a plusieurs critères d'arrêt : si le tableau ne comporte qu'un élément, soit c'est l'élément cherché auquel cas la fonction renvoie l'indice de cet élément, soit ce n'est pas lui et la fonction renvoie 0 ; le dernier cas est celui mentionné précédemment (l'élément est celui du milieu du tableau).

2.2. Ecrivez en pseudo-langage cette fonction.

```

Fonction dichotomie(i,j : entier, e : element) : entier
Var res, m : entier
Début
Si i=j
    Alors Si t[i]=e
        Alors res←i
        Sinon res←0
    FinSi
Sinon m←(i+j) div 2
    Si e=t[m]
        Alors res←m
        Sinon Si est[m]
            Alors res←dichotomie(i,m)
            Sinon res←dichotomie(m+1,j)
        FinSi
    FinSi
FinSi
Retourne(res)
Fin

```

Exercice 3 : Fichier texte (5 pts)

On dispose d'un fichier texte de notes (une note par ligne). Nous souhaitons calculer la moyenne, la médiane, la meilleure et la moins bonne note. En utilisant un tableau de notes pour faire les calculs, écrivez un **programme** qui calcule ces valeurs (vous pouvez écrire plusieurs fonctions ou procédures comme bon vous semble). Vous disposez des fonctions réel-vers-chaine et chaine-vers-réel qui traduisent respectivement un réel en une chaîne de caractères et une chaîne de caractères en un réel.

Programme fichier

Type tab : tableau[1..50] de réel

```

Var t : tab
    nom-fich : chaine
    i, n : entier
    nmax, nmin, nmoy, nmed, s : reel

```

Procédure fich-vers-tab(E nom : chaine, S t : tab, S n : entier)

```

Var ft : FT
    c : chaine

```

```

Début
n←0
ft←OuvrirEnLecture(nom)

```

```

TantQue not Finfichier(ft) Faire
  n←n+1
  c←lirechaine(ft)
  t[n]←chaine-vers-réel(c)
FinTantQue
Fermer(ft)
Fin

Début
  écrire('entrer le nom du fichier de notes')
  lire(nom-fich)
  fich-vers-tab(nom-fich,t,n)
  tri-rapide(t,1,n)

  nmin←t[1]
  nmax←t[n]
  nmed←t[n div 2]
  s←0
  Pour i←1 à n inc +1 faire
    s←s+t[i]
  FinPour
  nmoy←s/n
  écrire('Note min : ', nmin)
  écrire('Note max : ', nmax)
  écrire('Moyenne : ', nmoy)
  écrire('Médiane : ', nmed)
Fin

```

Exercice 4 : Tableau d'enregistrements (5 pts)

```

Const max = 20
Type rationnel = Enregistrement
  num, den : entier
  FinEnregistrement
  tab-rat = tableau[1..max] de rationnel

```

Où num et den sont le numérateur et le dénominateur d'un **rationnel positif ou nul**.

Dans cet exercice, on se restreindra à l'arithmétique sur des entiers.

4.1. Ecrivez une fonction qui réduit un rationnel quand c'est possible. Vous utiliserez la fonction pgcd(x,y) qui renvoie le pgcd de 2 entiers x et y.

```

Fonction reduire(r : rationnel) : rationnel
Var res : rat
  d : entier
Début
  Si r.num=0
    Alors res←r
  Sinon d←pgcd(r.num,r.den)
    Si d≠1 alors
      res.num←r.num div d
      res.den←r.den div d
    FinSi
  FinSi
  Retourner(res)
Fin

```

4.2. Ecrivez une fonction qui calcule la somme de deux rationnels.

```

Fonction additionner (r1,r2 : rationnel) : rationnel
Var res : rat
Début
  Si r1.den=r2.den
    Alors res.num←r1.num+r2.num
      res.den←r1.den
  sinon
    res.num←r1.num*r2.den+r2.num*r1.den
    res.den←r1.den*r2.den
  FinSi
  Retourner(réduire(res))
Fin

```

4.3. Ecrivez une fonction qui retourne le plus petit rationnel contenu dans le tableau t.

```

Fonction min-rat (t : tab-rat) : rationnel
Var min : rationnel
Début
  min←t[1]
  Pour i←2 à max inc +1 Faire
    Si t[i].num*min.den<t[i].den*min.num
      Alors min←t[i]
    FinSi
  FinPour
  retourner(min)
Fin

```