

Algorithmes et Structures de Données

Mardi 8 Novembre 2016

Durée 2H – Cours et TD **NON** autorisés

Exercice 1 : Pile (5 pts)

Considérons les 3 suites suivantes pour $n \geq 0$,

$$u_0=1, v_0=2, w_0=3$$

$$u_{n+1}=2u_n+3v_n+w_n$$

$$v_{n+1}=u_n+v_n+2w_n$$

$$w_{n+1}=u_n+4v_n+w_n$$

1.1. Ecrire les fonctions U, V et W en pseudo langage.

```
Fonction u (n :entier) : entier
Var res : entier
Début
  Si n=0 Alors res←1
  Sinon res←2*u(n-1){@1}+3*v(n-1){@2}+w(n-1){@3}
FinSi
Retourner(res)
Fin
```

```
Fonction v (m :entier) : entier
Var res : entier
Début
  Si m=0 Alors res←2
  Sinon res←u(n-1){@4}+v(n-1){@5}+2*w(n-1){@6}
FinSi
Retourner(res)
Fin
```

```
Fonction w (p :entier) : entier
Var res : entier
Début
  Si p=0 Alors res←3
  Sinon res←u(n-1){@7}+4*v(n-1){@8}+w(n-1){@9}
FinSi
Retourner(res)
Fin
```

1.2. Simuler la pile sur l'appel écrire(u(2)){@0} dans le programme principal.

```
@9, p=0 ----- res=3
@8, m=0 ----- res=2
@7, n=0 ----- res=1
@6, p=1 ----- res=1+4+2+3=12
@5, p=0 ----- res=3
@4, m=0 ----- res=2
@3, n=0 ----- res=1
@2, m=1 ----- res=1+2+2+3=9
@1, p=0 ----- res=3
@0, m=0 ----- res=2
@1, n=0 ----- res=1
@1, n=1 ----- res=2*1+3+2+3=11
@0, n=2 ----- res=2*11+3*9+12=61
```

Exercice 2 : Chaines de caractères (5 pts)

Soit une chaîne de caractères contenant des chiffres et éventuellement une virgule. On souhaite transformer cette chaîne en un réel correspondant à la valeur de cette chaîne.

Exemple : "436,78" → 436,78 ; "79,0" → 79 ; "45" → 45 ; "0,57" → 0,57 ; "4" → 0,4

2.1. Expliquer en français le principe pour réaliser cette transformation.

On doit parcourir la chaîne de gauche à droite et tant qu'on n'a pas rencontré la virgule, on multiplie le résultat par 10 (initialisé à 0) et on ajoute la traduction du caractère en entier. Une fois la virgule rencontrée, on ajoute au résultat la traduction du caractère en entier divisée par les multiples successifs de 10 (un à un) jusqu'à la fin de la chaîne.

2.2. Ecrire en pseudo-langage la fonction qui réalise cette transformation :

```
Fonction chaîne-en-réel (c : chaîne) : réel
Rappel : pour transformer un caractère c en chiffre n, on utilise le calcul suivant :
n ← ord(c)-ord('0')
```

```
Fonction chaîne-en-réel (c : chaîne) : réel
Var i : entier
  r : réel
Début
  r←0
  i←1
  TantQue c[i]≠',' et i≤lg[c] Faire
    r ← r * 10 + ord(c[i])-ord('0')
    i←i+1
  FinTantQue
  i←i+1
  j←10
  TantQue i≤lg[c] Faire
    r ← r + (ord(c[i])-ord('0'))/j
    j←j*10
    i←i+1
  FinTantQue
Retourner(r)
Fin
```

Exercice 3 : Récursivité (5 pts)

Soit un nombre entier **strictement positif**, nous souhaitons afficher les chiffres le constituant.

3.1. Expliquer en français la méthode pour réaliser cet affichage de gauche à droite de façon **récursive**. Ex : pour l'entier 45 071, les chiffres 4, 5, 0, 7 et 1 sont affichés successivement.

Ecrire en pseudo-langage la Procédure afficher-rec-gd (E n : entier)

Le principe récursif est le suivant : on doit isoler chaque chiffre composant le nombre . Ceci est possible en prenant le reste de la division entière du nombre par 10, puis en rappelant la fonction sur le nombre divisé par dix. Le principe s'arrête quand le nombre est nul. On fait les affichages après les appels récursifs pour que les chiffres s'affiche de gauche à droite.

```
Procédure afficher-rec-gd (E n : entier)
Début
  Si n>0 Alors
    Afficher-rec(n div 10)
    Ecrire(n mod 10)
  FinSi
Fin
```

3.2. Expliquer en français la méthode pour réaliser cet affichage de droite à gauche de façon **récur­sive**. Ex : pour l'entier 45 071, les chiffres 1, 7, 0, 5 et 4 sont affichés successivement.
Ecrire en pseudo-langage la Procédure afficher-rec-dg (E n : entier)

Le principe reste le même sauf qu'il faut maintenant écrire le chiffre avant de faire les appels récur­sifs.

```
Procédure afficher-rec-dg (E n : entier)
Début
  Si n>0 Alors
    Ecrire(n mod 10)
    Afficher-rec(n div 10)
  FinSi
Fin
```

Exercice 4 : Tableau et fichier texte (5 pts)

Nous souhaitons calculer la fréquence de chacune des notes (entières sur vingt) contenues dans un fichier texte (ce fichier contient une note par ligne).

4.1. Expliquer en français comment effectuer cette tâche de façon optimale en décrivant les structures de données utilisées.

On utilise un tableau de réels indicé de 0 à 20 (et initialisé à 0) pour représenter le nombre de notes pour chacune d'elles. Ensuite, il faut lire le fichier texte ligne par ligne (note par note) et pour chacune d'elles incrémenter le réel dont l'indice est la note lue. On compte au fur et à mesure le nombre de notes lues, puis quand le fichier est terminé, on divise les réels du tableau par ce nombre.

4.2. Ecrire en pseudo-langage le programme qui réalise ce calcul.

```
Programme fréquences
Type tab = tableau [0..20] de réel
Var t : tab
  i, n : entier
  nomf, c : chaîne
  f : FT

Début
  Pour i←0 à 20 inc +1 Faire
    t[i]←0
  FinPour
  Ecrire('Entrez le nom du fichier de notes : ')
  Lire(nomf)
  f←OuvrirEnLecture(nomf)
  n←0
  TantQue ~Finfichier(f) Faire
    c←LireChaîne(f)
    i←partie-entière(chaîne-en-réel(c))
    t[i]←t[i]+1
    n←n+1
  FinTantQue
  Pour i←0 à 20 inc +1 Faire
    t[i]←t[i]/n
  FinPour
  Fermer(f)
Fin
```