UNIVERSITE MENTOURI
FACULTE DES SCIENCES DE L'INGENIEUR
Année universitaire : 2007 – 2008
Formation : LMD (MI)
Module : info2
Durée: 1 heure 30 mn

Mai 2008

Contrôle de longue durée

Exercice1: (6 points)

Soit une liste circulaire doublement chaînée où chaque élément (nœud) est constitué de trois champs : prec, info et suiv. Les champs prec et suiv contiennent respectivement l'adresse de l'élément précédent et l'adresse de l'élément suivant. Le champ info concerne les informations relatives à des pièces détachées :

- Label (nom)
- Référence (numéro)
- Prix

1) Donner une déclaration en Pascal de cette liste.
2) Ecrire une procédure paramétrée en Pascal de rajout au début de cette liste d'une pièce détachée X.

Exercice2: (7 points)

Soit une pile P1 de clients d'une banque. Un client est caractérisé par les informations suivantes: Nom; Prénom; Numéro de compte; Solde (montant de son avoir - الرصيد البنكي)

Le directeur de la banque désire éclater cette pile P1en 2 autres piles P2 et P3 où:

- P2 comporte les clients à découvert (solde ≤ 0)
- P3 comporte les clients qui ne sont pas à découvert (solde > 0)

1) Donner une déclaration en Pascal de ces trois piles
2) Ecrire une fonction booléenne en Pascal qui vérifie si une pile est vide.
3) Ecrire une procédure paramétrée en Pascal qui réalise l'éclatement en utilisant les procédures paramétrées empiler et dépiler (sans les écrire). La pile P1 reste inchangée.

Exercice3: (7 points)

Soit une matrice carrée MAT(N,N) (avec N=10) contenant des entiers strictement positifs inférieurs ou égaux à 250 (≤ 250). On désire mettre dans une file d'attente FP les nombres premiers contenus dans cette matrice.

1) Donnez une déclaration en Pascal de la matrice et de la file d'attente.
2) Ecrire une fonction **Premier(Nb : integer) : boolean** qui vérifie si un nombre Nb est premier ou non.
3) Ecrire une procédure en Pascal qui permet d'enfiler un élément dans la file FP.
4) Ecrire une procédure en Pascal qui utilise la fonction premier et la procédure enfiler pour réaliser le travail cité ci-dessus.

Remarques générales:

- Dans tout le contrôle, les piles et les files sont dynamiques.
- Il est permis d'utiliser les procédures paramétrées initpile et initfile (qui s'occupent de l'initialisation respectivement de la pile vide et de la file vide) ainsi que de la fonction booléenne filevide (qui teste si la file est vide).
- Les procédures et fonctions non paramétrées ne seront pas prises en compte.

Bonne Chance

MIAS - Informatique II - Mai 2008

- Solutions Exo I - 6pts

1) Implementation.

```
const
    N = 30;
Type
    Piece = Record
                Lab : String[N];
                Ref : Integer;          1
                Prix : real
            end;
    Liste = ^node;
    node = Record
                Prec : Liste;
                Info : Piece;           1
                Suiv : Liste
           end;
Var
   L : Liste;
```

2) Procedure Rajout (var L : Liste ; X : Piece);

```
Var
   E, Pt : Liste;
Begin
    new(E);
    E^.Info := X;                       1
    IF L = nil Then
      Begin
          E^.prec := E;                 1
          E^.Suiv := E
      end
      else
         Begin
             Pt := L;
             E^.Suiv := Pt;
             E^.prec := Pt^.prec;
             (Pt^.prec)^.Suiv := E;     2
             Pt^.prec := E
         end;
    L := E
end;
```

TIAS - Informatique II - TAI 2008

- Solution Exo II - 7pts

1/ Implementation.

```
Const N = 30;
Type
   Item = Record
            nom,
            Pnom : String [N];   1
            Num : Integer;
            Solde : real
          end;
   Liste = ^node;
   node = Record
            Info : Item;
            Suiv : Liste
          end;                    1
   Pile = Record
            Sommet : Liste
          end;
Var
   P1, P2, P3 : Pile;
```

2/ Function vide (P : Pile) : booleau;

```
Begin
   vide := (P.Sommet = nil)      1
end;
```

3/ Procedure Eclater (P1 : Pile ; Var P2, P3 : Pile); 1

```
Var
  X : Item;
Begin
     Initialise (P2);
     Initialise (P3);   1
     while not (vide(P1)) Do
          Begin
               Depiler (P1, X);
               IF X.Solde <= 0 Then   2
                  Empiler (P2, X)
               else
                  Empiler (P3, X)
          end
end;
```

3. $E=\{u\in IR^3 / f(u)=u\}$ sev de IR^3 ?

i) $f(0_{IR^3})=0_{IR^3}\Rightarrow 0_{IR^3}\in E\Rightarrow E\neq\phi$

ii) Soient $\alpha,\beta\in IR$ et $u,u'\in E$, nous avons

$f(\alpha u+\beta u')=\alpha f(u)+\beta f(u')$ (def. de f)

$=\alpha u+\beta u'$ $(u,u'\in E)$

$f(\alpha u+\beta u')=\alpha u+\beta u'\Rightarrow \alpha u+\beta u'\in E$

4. Recherche d'un supplémentaire G de E

$$E=\{u=(x,y,z)\in IR^3 / f(u)=u\}=\{(x,y,z)/(x+y,y+z,z)=(x,y,z)\}$$
$$=\{(x,y,z)/x+y=x\wedge y+z=y\wedge z=z\}=\{(x,y,z)/y=z=0\}=\{(x,0,0)/x\in IR^3\}$$
$$=[\{(1,0,0)\}]$$

Ainsi E est engendré par le seul vecteur $e_1=(1,0,0)$ qui en constitue une base, on complète cette dernière en une base de IR^3 en rajoutant, par exemple, les vecteurs $e_2=(0,1,0)$ et $e_3=(0,0,1)$ et l'on aura $G=[\{e_2,e_3\}]$

Autre possibilité :

$(\forall u=(x,y,z)\in IR^3)\ u=(x,0,0)+(0,y,z)=u_1+u_2\quad (u_1\in E\wedge u_2\in G)$

avec $G=\{(0,y,z)/y,z\in IR\}$ (sev de IR^3)

La décomposition étant unique, on a bien $IR^3=E\oplus G$

5.

$(\forall u\in IR^3)\qquad g(u)=f(u)-u\Leftrightarrow g(x,y,z)=(x+y,y+z,z)-(x,y,z)=(y,z,0)$

$$g^2(u)=gog(u)=g(g(u))=g(y,z,0)=(z,0,0)$$
$$g^3(u)=g(g^2(u))=g(z,0,0)=(0,0,0)$$

$(\forall u\in IR^3)\ g^3(u)=0_{IR^3}\Leftrightarrow g^3=0\Rightarrow(\forall n\geq 3)\ g^n=0$

$$[(\forall u)\ g(u)=f(u)-u]\Leftrightarrow[f=g+Id_{IR^3}]\Rightarrow f^n=(g+Id_{IR^3})^n=\sum_{p=0}^{n}C_n^p g^p$$

$$f^n=C_n^0g^0+C_n^1g+C_n^2g^2+\underbrace{C_n^3g^3+\cdots+C_n^ng^n}_{0}$$

$$=Id_{IR^3}+ng+\frac{n(n-1)}{2}g^2$$

$$f^n(x,y,z)=(x,y,z)+n(y,z,0)+\frac{n(n-1)}{2}(z,0,0)=(x+ny+\frac{n(n-1)}{2}z,y+nz,z)$$

MIAS - Informatique II - Mai 2008

Solution - Exo III - 7pts

1/ Implementation.

```
Const
    N = 10;
Type
    Item = 1..250;  (* ou bien Item = Byte *)
    Liste = ^node;
    node = Record
                Info : Item;
                Suiv : liste
           end;                                    2
    Filat = Record
                Tete,
                queue : Liste
           end;
    matrice = array [1..N, 1..N] of Item;
Var
    Mat : matrice;
    FP : filat;
```

2/

```
Function  Premier (Nb : Integer) : boolean;
Var
    D : 2.. Nb Div 2;   (* ou bien D : Integer; *)
    Prem : boolean;
Begin
    D := 2
    Prem := True;
    while (D <= Nb div 2) and (Prem) Do
        IF Nb mod D = 0 Then
            Prem := false                          2
        else
            D := D + 1;
    If Prem Then
        Premier := True
    else
        Premier := false
end;
```