Contrôle continu 2008-2009
INF251
Pointeurs - Récursivité – Listes – Piles - Files

Questions de cours (6 points)

- La primitive dépiler doit-elle contenir l'appel à pileVide ? Pourquoi ?
  Non. Tout d'abord la primitive dépiler(telle que décrite en cours) n'est pas de type booléen, par suite on ne pourrait pas distinguer le cas où on dépile dans une pile vide du cas standard. Ensuite, dans des algorithmes on pourrait être amené à tester avant de dépiler il y aurait donc un second test de pile vide au sein de la primitive. Les langages de programmations actuels permettent d'insérer un tel test qui pourra être supprimé dans une compilation finale (instruction assert en C par exemple)
- Donnez une définition du type file en implémentation dynamique.
  Type file = structure
  premier, dernier:^cellule ;
  finstructure
  Type cellule = structure
  Info :objet ;
  Suivant :^cellule ;
  Finstructure
  On peut aussi utiliser comme dans le cours des listes doublement chainées.
- Pour une pile ayant N objets empilés, peut-on connaitre l'élément N-3 (quatrième élément sous le sommet de pile)?

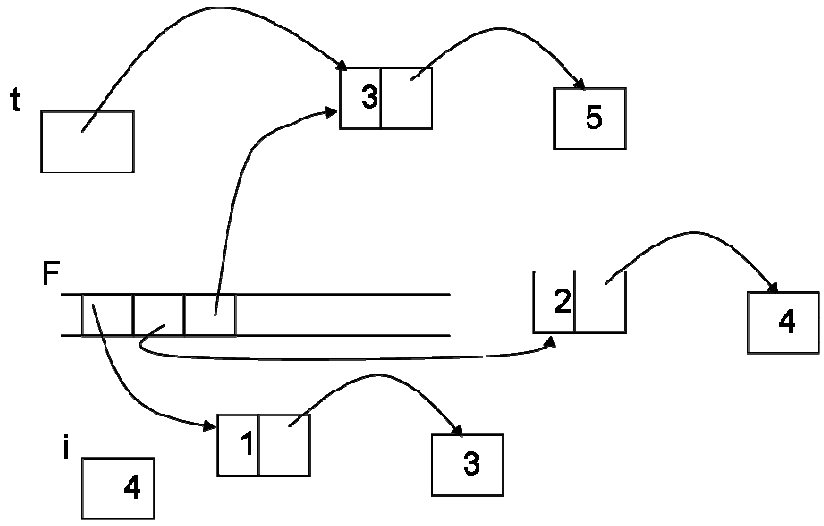
En général une pile n'est pas adaptée à ce genre d'opération. Cependant, il suffit d'utiliser 3 variables temporaires pour stocker les éléments en sommet de pile, puis on consulte le sommet de pile et on empile dans l'ordre inverse les trois valeurs des variables temporaires. S'il s'agit de consulter le premier élément de la pile, il faut une seconde pile.

- Soit la fonction :

```
Fonction essai() :entier ;
  var t : ^tag ;
   type tag=structure
            v : entier ;
            s :^entier ;
          finstructure
   var F :file de ^tag ;
   var i :entier ;
   creerFile(F) ;
   pour i=1 à 3 faire
     new(t) ;
     new(t^.s) ;
     t^.v=i;
     t^.s^=i+2;
     enfiler(F,t);
   finpour
   retourner(valeur(F) ^.v) ;
finfonction
```

Dessinez la mémoire au moment de l'exécution de l'instruction retourner.

Que retourne la fonction ? Pourquoi cette fonction est-elle mal écrite ?



Cette fonction retourne 1.
La fonction est mal écrite car la file créée par cette fonction n'est ni retournée à la fonction appelante ni détruite. De plus, de la mémoire dynamique a été préemptée qui ne sera pas détruite par un delete.

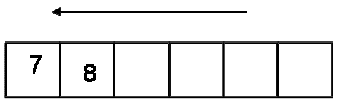
Exercice 1 (3 points)

1- Soit P une pile d'entiers, quel est l'état de la pile après les opérations suivantes :

créerPile(P) ;empiler(P,7), ;empiler(P,4) ; empiler(P,5) ; empiler(P,valeur(P)) ;dépiler(P) ; empiler(P,7) ;

2- Soit F une file d'entiers, quel est l'état de la pile après les opérations suivantes :

créerFile(F) ;enfiler(F,7), ;enfiler(F,4),enfiler(F,valeur(F)) ;enfiler(F,8) ;défiler(F) ;defiler(F) ;



Exercice 2 (11 points)

1- Ecrire une fonction qui détruit une file de cellule du type
Cellule=^entier ;

```
fonction detruire(ref F :file de Cellule) :vide ;
début
    Tant que !fileVide(F) faire
        delete(valeur(F)) ;
        depiler(F) ;
    fintantque
    detruireFile(F)
fin
```

2- Ecrire une fonction qui prend en entrée une pile d'entier et qui fournit en sortie une liste simplement chainée telle que tous les nombres multiples de 3 se retrouvent en tête de liste et dans le même ordre que la pile. La pile d'entrée ne doit pas être modifiée en sortie. Par exemple, [1,3,5,4,2,6,8 (1 est le fond de pile) est transformée en (3,6,1,5,4,2,8).

```
fonction pileToListeSC3(val  P :pile de entier) listeSC de entier ;
    var L :listeSC d'entier ;
    var fileTmp :file d'entier ;
    var n :entier ;
    début
        creerListe(L) ;
        creerFile (fileTmp);
        tant que !pileVide(P) faire
            n=valeur(P) ;
            depiler(P);
            si n % 3 alors
                enfiler(filetmp,n) ;
            sinon
                ajouterEnTete(L,n)
            finsi
        finTantQue
        tantque !fileVide(fileTmp) faire
            ajouterEnTete(L, valeur(fileTmp))
            defiler(fileTmp);
        fintantque
        detruireFile(fileTmp) ;
        retourner(L) ;
    fin
```

Primitives et Fonctions

Liste Simplement Chainées

fonction valeur(val L:liste d'objet):objet;
fonction debutListe(ref L:liste d'objet);
fonction suivant(ref L:liste d'objet);
fonction listeVide(val L:liste d'objet): booleen;
fonction créerListe(ref L:liste d'objet):vide;
fonction insérerAprès(ref L:liste d'objet; val x:objet;):vide;
fonction insérerEnTete(ref L:liste d'objet,val x:objet):vide;
fonction supprimerAprès(ref L:liste d'objet):vide;
fonction supprimerEnTete(ref L:liste d'objet):vide;
fonction setCléListe(ref L: liste d'objet;val c:curseur):vide;
fonction detruireListe(ref L:lisetd'objet):vide;

Liste Doublement Chainées (ajout de)

fonction finListe(ref L:liste d'objet):vide;
fonction précédent(ref L::liste d'objet): vide;

Fonctions sur les listes

fonction estFinListe(val L:liste d'objet):booléen;
fonction chercher(ref L:liste d'objet; ref x:objet): booleen;
fonction supprimer(ref L:liste d'objet; ref x:objet): vide;

Piles

fonction valeur(ref P:pile de objet):objet;
fonction fileVide(ref P:pile de objet):booléen;
fonction créerPile(P:pile de objet);
fonction empiler(ref P:pile de objet;val x:objet):vide;
fonction dépiler(ref P:pile de objet):vide;
fonction detruirePile(ref P:pile de objet):vide;

Files

fonction valeur(ref F:file de objet):objet;
fonction fileVide(ref F:file de objet):booléen;
fonction créerFile(F:file de objet);vide;
fonction enfiler(ref F:file de objet;val x:objet):vide;
fonction défiler (ref F:file de objet):vide;
fonction detruireFile(ref F:file de objet):vide;