Contrôle continu 2007-2008
INF251
Pointeurs - Récursivité – Listes – Piles - Files

Questions de cours (5 points)

- Quelle structure de données est utilisée pour gérer les appels récursifs ?

Une pile

- Quel type de liste peut-on utiliser pour gérer une pile ?

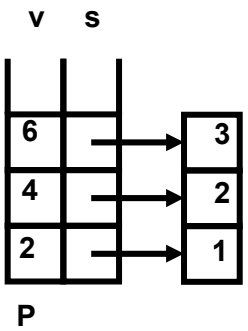
Une liste simplement chainée

- Quand on implémente en allocation dynamique, quelle est la différence entre une liste simplement chainé et une file ?

Pour la file il faut ajouter un pointeur sur le dernier élément.

- Dessinez la mémoire, après la suite d'opérations suivante :

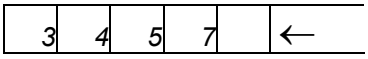
```
var p : pile de tag;
var t :tag ;
type tag=structure
        v :entier ;
        s :^entier ;
      finstructure
creerPile(P) ;
pour i=1 à 3 faire
  new(t.s) ;
  t.v=2*i;
  t.s^=i;
  empiler(P,t);
finpour
```



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,4), ;empiler(P,3),empiler(P,valeur(P)) ;empiler(P,5) ;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,4), ;enfiler(F,3),enfiler(F,valeur(F)) ;enfiler(F,5) ;défiler(F) ;enfiler(F,7) ;



Exercice 2 (12 points)

1- Ecrire une fonction qui crée une copie d'une file d'entiers non nuls.

La fonction étant mal formulée une réponse très correcte et juste est la suivante

```
fonction copieFile(ref F :file de entiers): file de entiers ;
  var FC :file de entiers ;
  début
    FC=F ;
    Retourner(FC) ;
  Fin
```

L'enseignant aurait du préciser « sans affectation directe »

```
fonction copieFile(ref F :file de entiers): file de entiers ;
  var FC :file de entiers ;
  var v :entier ;
  début
    creerFile(FC) ;
    enfiler(F,0) ;
    v=valeurFile(F) ;
    tantque v !=0 faire
      defiler(F) ;
      enfiler(FC,v) ;
      v=valeurFile(F) ;
    fintantque
    defiler(F) ;
   retourner(FC) ;
  fin
```

2- Ecrire une fonction qui crée une liste simplement chainée LC à partir d'une liste L simplement chainée en mettant en tête les nombres pairs dans le même ordre et ensuite les nombres impairs en ordre inverse de leur apparition dans L. Par exemple si L=(3,4,2,9,8,5,1,7,6) alors LC=(4,2,8,6,7,1,5,9,3).

```
fonction listeCopie(ref L :listeSC de entiers) :listeSC de entiers ;
  var LC :listeSC d'entiers ;
  var ind :^cellule de entier ;
  var v :entier ;
  var P :pile de entiers ;
  début
    creerPile(P) ;
    creerListe(LC) ;
    ind=premier(L) ;
    tant que ind !=NIL faire
      v= contenu(ind) ;
      si estPair(v) alors
        empiler(P,v) ;
      sinon
        ajouterEnTete(v,LC)
```

```
        finsi
        ind=suivant(L,ind) ;
      fintantque
      tantque !pileVide(P) faire
        ajouterEnTete(valeur(P),LC)
        depiler(P)
      fintantque
      retourner(LC)
    fin
```

Listes simplement chainées (listeSC)

```
fonction premier(val L:type_liste):^type_predefini;
fonction suivant(val L:type_liste; val P:^type_predefini):^type_predefini;
fonction listeVide(val L:type_liste):booléen;
fonction créer liste(ref L:type_liste):vide;
fonction insérerAprès(val x:type_prédéfini;ref L:type_liste; val P:^type_predefini):vide;
fonction insérerEnTete(val x:type_prédéfini;ref L:type_liste):vide;
fonction supprimerAprès(ref L:type_liste;val P:^type_predefini):vide;
fonction supprimerEnTete(ref L:type_liste):vide;
```

Listes doublement chainées (listeDC), On ajoute les primitives suivantes

```
fonction dernier(val L:type_liste):^type_predefini;
fonction précédent(val L:type_liste; val P:^type_predefini):^type_predefini;
```

Piles

```
fonction valeur(ref P:pile de type_predefini):type_predefini;
fonction pileVide(ref P:pile de type_predefini):booléen;
fonction empiler(ref P:pile de type_predefini; val v:type_predefini):vide;
fonction dépiler (ref P:pile de type_predefini):vide;
fonction créerPile(P:pile de type_predefini);
```

Files

```
fonction valeur(ref F:file de type_predefini):type_predefini;
fonction fileVide(ref F:file de type_predefini):booléen;
fonction enfiler(ref F:file de type_predefini; val v:type_predefini):vide;
fonction défiler (ref F:file de type_predefini):vide;
fonction créerFile(F:file de type_predefini);
```