

# Pointeurs et listes chaînées

M2 AIGEME, année 2008-2009

## Exercice 1 : Bidouillage

1. Qu'affiche le code suivant ?

```
Var p1,p2 : *Entier;  
    i,j : Entier;  
  
Debut  
i ← 4;  
j ← 7;  
p1 ← &j  
p2 ← &i  
Afficher(*p1+*p2)  
Fin
```

2. Et celui-ci ?

```
Var p1,p2 : *Entier;  
    i,j : Entier;  
  
Debut  
i ← 4;  
j ← 7;  
p1 ← &j  
p2 ← &i  
Afficher(p1+p2)  
Fin
```

3. Cette écriture est-elle correcte ?

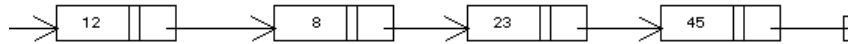
```
Var p1,p2 : *Chaîne de caractères;  
    mot1,mot2 : Chaîne de caractères;  
  
Debut  
mot1 ← "debut";  
mot2 ← "fin";  
p1 ← mot1  
p2 ← mot2  
Afficher(*p1+*p2)  
Fin
```

## Exercice 2 : Liste chaînée

On utilise dans tout cet exercice des listes chaînées dont le type est le suivant :

```
Type Cellule={
  info : Entier;
  suivant : *Cellule;
}
Type Liste = *Cellule;
```

On représentera graphiquement une liste chaînée de la manière suivante (le pointeur NULL étant représenté par le trident à droite de la figure):



Pour les questions 3 à 5, il est **fortement conseillé** de faire des dessins pour bien voir ce qu'il faut faire.

1. Ecrire une fonction qui affiche successivement tous les éléments d'une liste chaînée :

```
Fonction affiche_liste(l : Liste)
```

2. Ecrire une fonction qui recherche un élément (entier du champ **info**) dans une liste chaînée. La fonction renvoie un pointeur vers la cellule contenant cet élément. Si l'élément n'apparaît pas dans la liste, la fonction renvoie le pointeur NULL.

```
Fonction recherche_liste(x : Entier, l : Liste) : *Cellule
```

3. Ecrire une fonction qui ajoute une cellule au début d'une liste chaînée.

```
Fonction ajoutd_liste(c : Cellule, l : Liste) : Liste
```

4. Ecrire une fonction qui ajoute une cellule à la fin d'une liste chaînée.

```
Fonction ajoutf_liste(c : Cellule, l : Liste) : Liste
```

5. Ecrire une fonction qui supprime une cellule contenant une certaine information d'une liste chaînée (et ne modifie pas la liste si cette information n'est pas présente).

```
Fonction supp_liste(x : Entier, l : Liste) : Liste
```