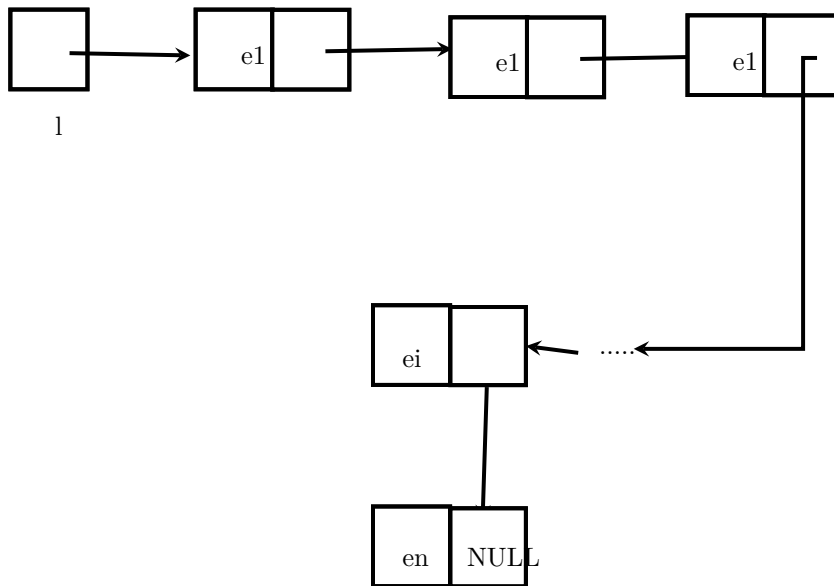


# TD 1

## Construction dynamique d'une liste

TAD  
Semestre 1

### 1 Représentation de la liste



### 2 Définition du type de la liste

```
1 type PtrCellule : pointeur sur <Cellule>;
2 type Cellule : enregistrement
3   valeur <Entier>,
4   suivant <PtrCellule>;
```

Listing 1 – Type de la liste

### 3 Construction de la liste

#### 3.1

#### 3.2

##### 3.2.1 Avant la première insertion

#### 3.3

```
1 glossaire
2 l <PtrCellule>;
```

```

3   dernier <PtrCellule>;
4   nouveau <PtrCellule>;
5   e <Entier>;
6   marqueur <Entier>;
7   debut
8     lire(marqueur);
9     lire(e);
10    l <- NULL;
11    dernier <- NULL;
12
13    tantque e /= marqueur faire
14      allouer(nouveau);
15      nouveau↑.valeur <- e;
16      nouveau↑.suivant <- NULL;
17      si l = NULL alors
18        l <- nouveau;
19      sinon
20        dernier↑.suivant <- nouveau;
21      fin si;
22      dernier <- nouveau;
23      lire(e);
24    fin tantque;
25  fin

```

## 4 Impression de la liste en détruisant au fur et à mesure ses éléments

### 4.1

```

1   se positionner sur le premier element de la liste;
2
3   tantque il reste un element faire
4     ecrire l'element courant;
5     detruire l'element courant;
6     passer a l'element suivant;
7   fin tantque;

```

### 4.2

#### 4.2.1 Après la première insertion

#### 4.2.2 Suppression autre que la première

même séquence de code que a (car suppression toujours en tête de la liste)

### 4.3

```

1   glossaire
2     l <PtrCellule>;
3     dernier <PtrCellule>;
4     nouveau <PtrCellule>;
5     courant <PtrCellule>;
6     e <Entier>;
7     marqueur <Entier>;
8   debut
9     tantque l /= NULL faire
10      ecrire(l↑.valeur);

```

```
11     courant <- l;  
12     l <- l↑.suivant;  
13     liberer(courant);  
14     fin tantque;  
15     dernier <- NULL;  
16     fin
```