

## Correction Exercice d'application chapitre 5 Pile :

### Algorithme Chiffre

#### Var

nb, chiffre : entier

#### Début

**Ecrire** ("Donner nb")

**Lire** (nb)

Crée\_pile ()

#### Répéter

chiffre <- nb mod 10

Empiler (chiffre)

nb <- nb div 10

**Jusqu'à** (nb=0)

**Tantque**(non (pile\_vide()))

**Ecrire** (dernier())

Dépiler()

**Fin tantque**

**fin**

### Correction exercice 1 TD pile

```
// Modification appropriées
Nb : entier static
//pour la procedure crée pile
Procédure crée_pile()
Début
    ...
    Nb <- 0
Fin proc

//pour la procédure empiler
procédure empiler(Info : Entier)
Début
    ...
    Nb <- Nb + 1
Fin proc

//pour la procédure dépiler
Procédure dépiler()
Début
    ...
    Nb <- Nb - 1
Fin proc

// Ajout de la fonction nb_element
Fonction Nb_élément () : entier
Début
    Nb_élément <- Nb
Fin Fn
```

```
//Ajout de la procédure remplacer sommet
// solution 1
Procédure remplace_sommet (Info: entier)
Début
    Assure ( ! pile_vide() )
    Dépiler()
    Empiler(Info)
Fin proc

// solution 2
Procédure remplace_sommet (Info: entier)
Début
    Assure ( ! pile_vide() )
    sommet^.cle ← info
Fin proc

//Ajout de la procédure qui efface tout les
éléments
Procédure effacer_tout ()
Début
    Tantque ( ! pile_vide() ) faire
        Dépiler()
    Fin tantque
Fin proc
```

## Correction exercice 2 TD pile

### Algorithme segmentation

**var**

ch : **chaîne** /\* expression proposée \*/

sch : **chaîne** /\* expression courant entre parenthèses \*/

i: **entier**

**debut**

**ecrire** (“donner une expression”);

**lire** (ch);

**pour** i **de** 1 **à** length(ch) **faire**

**si** ( ch[i]= '(' ) **alors**

        empiler(i);

**sinon si** ( ch[i]= ')' ) **alors**

        sch <- **sous\_chaine** (ch, dernier(), i);

/\*RQ : la fonction Chaîne sous\_chaine(Ch : chaîne, debut : entier , fin : entier)

    rend la sous\_chaine comprise entre ch[debut] et ch[fin] \*/

**ecrire\_Ln** (sch);

        depiler();

**fin si**

**fin Pour**

**fin**