Travaux Dirigés d'Algorithmique n°2

Cours d'Informatique de Deuxième Année

—DEUG SCIENCES S4-MI option Informatique—

Structure de pile - calculatrice

Dans cette séance de travaux dirigés, nous abordons une nouvelle structure : la pile. Nous définissons le type *C* `pile` et nous implémentons les fonctions de base de manipulation de pile. Nous continuons par une étude des expressions arithmétiques écrites en notation polonaise inversée. Nous finissons par une application de la structure `pile` en implémentant une calculatrice de type `HP`.

Nous introduisons la structure de donnée `pile` pour représenter une pile d'éléments de type `data`.

```
#define TAILLE_PILE 100

typedef struct {
      data tab[TAILLE_PILE]; /* tableau de TAILLE_PILE nombre */
      int sommet;            /* sommet de la pile */
      } pile ;
```

▸ **Exercice 1. (Fonctions de base)** Écrire les fonctions suivantes :

- `void InitialiserPile(pile *)` : initialise une pile ;
- `int Empiler(pile *, data)` : ajoute un élément dans une pile ;
- `int Depiler(pile *, data *)` : retire un élément d'une pile ;
- `int EstVidePile(pile)` : teste si une pile est vide.

Avant d'aller plus loin, tester si tout fonctionne.

▸ **Exercice 2. (Notation polonaise inversée)** Donner la notation en polonaise inversée des expressions suivantes :

- 25*4+3 ;
- 25*(4+3) ;
- 2+3+5 ;
- (23+4/3)+4*5*(3+2).

▸ **Exercice 3.** L'évaluation d'une expression écrite en notation polonaise inversée peut s'effectuer à l'aide d'une pile. Toute expression sera préalablement découpée en « mots » (opérateur ou nombre). L'algorithme d'évaluation fonctionne de la manière suivante : soit `m` le mot en cours,

- si `m` est un opérande (nombre), on l'empile ;

- si `m` est un opérateur, on dépile les opérandes, on applique l'opérateur à ces dernières, et on empile le résultat.

Finalement, la pile sert donc à stocker les résultats intermédiaires du calcul.

Nous allons maintenant appliquer cet algorithme afin de réaliser une petite calculatrice de type `HP`. On pourra s'inspirer de la commande unix `dc`.

Dans un premier temps, nous décomposons l'expression saisie par l'utilisateur en « mots ». Ensuite, nous appliquons l'algorithme précédent à l'aide des fonctions de gestion de pile.

- Écrire une fonction `EstOperateur` prenant en argument une chaîne de caractères `ch` et renvoyant 1 si le premier mot est un opérateur, 0 si c'est un nombre. Dans les deux cas, le programme principal devra pouvoir récupérer l'opérateur ou le nombre, et devra savoir quelle partie de la chaîne a déjà été traitée.
- Écrire une fonction `Evaluation` prenant en argument une expression (en polonaise inversée) sous la forme d'une chaîne de caractères et qui retourne l'évaluation de cette expression ;
- En s'inspirant de la commande `dc`, modifier les fonctions précédentes afin de réaliser une calculatrice évaluant des expressions arithmétiques écrites en notation polonaise inversée.