

Les Piles – Comprendre la gestion d’une pile – L’implémenter sous forme d’une liste chaînée – PUSH et POP

Solution :

Parmi les différentes notations possibles d’expressions mathématiques, deux sont parmi les plus utilisées :

- La notation infixée, dans laquelle les opérateurs sont entourés par les opérandes ; par exemple $3 * 4$ et $(3 + 4) * 5$ sont des expressions en notation infixée.

- La notation postfixée, dans laquelle les opérateurs suivent les opérandes ; par exemple $3 4 *$ et $3 4 + 5 *$ sont les expressions en notation postfixée correspondant aux opérations précédentes.

L’algorithme général de calcul d’une expression en notation postfixée s’effectue simplement à l’aide d’une pile.

L’algorithme général de calcul s’explique comme suit :

- Si le mot courant est un nombre, l’empiler.
- Si le mot courant est un opérateur, dépiler les deux opérandes, effectuer l’opération, puis empiler le résultat.
- L’algorithme s’arrête lorsqu’il n’y a plus d’éléments en entrée

Question 1 :

Dessiner l’état de la pile à chaque étape de traitement de l’expression postfixée suivante : $2 5 * 3 + 2 *$

Donner l’équivalent de la notation postfixée en notation infixée.

- Départ du programme : la pile est vide
- Lecture du 2 : c’est un nombre => on empile 2
- Lecture du 5 : c’est un nombre => on empile 5
- Lecture de * : c’est un opérateur => on dépile 2 et 5, on fait l’opération $(2*5)$ et on empile 10
- Lecture de 3 : c’est un nombre => on empile 3
- Lecture de + : c’est un opérateur => on dépile 3 et 10, on fait l’opération $(3+10)$ et on empile 13
- Lecture de 2 : c’est un nombre, on empile 2
- Lecture de * : c’est un opérateur, on dépile 2 et 13, on fait l’opération $(2 * 13)$ et on empile le résultat 26

Il n’y a plus de caractères en entrée : le résultat est 26.

Soit les schémas successifs suivants :

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|
| | | | | | | | |
| | | | | | | | |
| | | 5 | | 3 | | 2 | |
| | 2 | 2 | 10 | 10 | 13 | 13 | 26 |
| Etape 1 | Etape 2 | Etape 3 | Etape 4 | Etape 5 | Etape 6 | Etape 7 | Etape 8 |

La notation infixée est : $((2 * 5) + 3) * 2$

Question 2 :

On désire implémenter la pile sous forme de liste chaînée. La tête de liste correspondant au sommet de la pile. Etablir les déclarations de type nécessaires (les éléments de la pile sont stockés sous forme de chaînes de caractères)

```
Type ptElement = ^Pile
Type Pile = Structure
    Element : chaîne
    Suivant : ptElement
FinStructure
```

Question 3 :

Ecrire la procédure PUSH qui empile un élément PUSH et la procédure POP qui dépile le sommet de pile.

```
Procédure PUSH (E/S sommet : ptElement, E x : chaîne)
// Procédure qui place un élément x en sommet de la pile pointée par sommet
//
```

Déclaration

```
p : ptelement
```

Début

```
p ← Allouer (Pile)           // on alloue un espace pour ce nouvel élément
p^.Element ← x               // on lui affecte la valeur contenue dans x
p^.Suivant ← sommet         // on insère cet élément en tête de liste
sommet ← p                   // le sommet pointe sur ce nouvel élément
```

Fin Procédure

Procédure POP (E/S sommet, S x : chaîne)

```
// Procédure qui enlève l'élément en sommet de pile
// Renvoie « erreur » dans x si la pile est vide
```

Déclaration

```
p : ptelement
```

Début

```
Si sommet <> null alors // Attention de ne pas vouloir dépiler une pile vide
```

Début

```
p ← sommet // mémoriser l'adresse de l'élément au sommet
Sommet ← Sommet^.Suivant // changer l'adresse du sommet
x ← p^.Element // mémoriser la chaîne dépilée
désallouer (p) // libérer l'espace
```

Fin

Sinon

```
x ← « erreur » // La pile est vide, on ne peut dépiler
```

Finsi

Fin Procédure