

TP Algo 2

Partie I: Pile et File

1. Pile

Le fichier cours2.c contient la déclaration d'une pile ainsi que de plusieurs fonctions opérant sur cette pile.

1. Copiez le code propre à la pile dans un nouveau fichier puis complétez les fonctions (vous pouvez reprendre le code de l'exercice "Vecteur" du TP 4 du cours de C).
2. Écrivez une fonction print_pile qui affiche le contenu de la pile à l'écran puis écrivez une fonction main qui teste toutes ces fonctions.

2. File

1. Copiez le code propre à la file du fichier cours2.c dans un nouveau fichier puis complétez les fonctions (Réutilisez la liste chaînée du TP précédent).
2. Écrivez une fonction print_file qui affiche le contenu de la file à l'écran puis écrivez une fonction main qui teste toutes ces fonctions.

Partie II Algorithmes

3. Notation polonaise inversée

1. Créez une fonction appelée calcul qui accepte en argument une chaîne de caractères contenant l'expression au format Notation polonaise inversée et renvoie le résultat de l'expression. Les seules opérations acceptées seront la multiplication, la division, la soustraction et l'addition
2. Créez une fonction qui accepte en argument une expression en notation standard et qui la transforme en Notation Polonaise Inversée.
3. Créez un programme qui demande à l'utilisateur d'entrer une expression et qui lui affiche le résultat
4. Sauvegardez le code de ces fonctions pour l'exercice suivant

4. Condition

Le but de cet exercice est de modifier le programme précédent pour qu'il permette à l'utilisateur de renvoyer un résultat différent selon une condition

- Modifiez le code de traduction de l'expression standard pour qu'il accepte un opérateur de plus: l'opérateur ?
- Modifiez le code de la fonction calcul pour qu'elle prenne en compte cet opérateur:
 - L'opérateur prend les 3 dernières valeurs sur la pile et renvoie la troisième si la première est 0, sinon il renvoie la seconde

Vous pouvez tester ces expressions :

- L'expression $123?$ renverra 2 car 3 est différent de 0
- L'expression $120?$ renverra 1 car 0 est égal à 0
- L'expression $1234+?$ renverra 2 car le résultat de $3 + 4$ est différent de 0
- L'expression $1234+0*?$ renverra 1 car le résultat de $0 * (3 + 4)$ est 0

5. Variables

Le but de cet exercice est de modifier le programme précédent pour qu'il permette à l'utilisateur d'utiliser 3 variables : a, b et c.

- La notation standard comprendra un nouvel opérateur : le signe égal
 - L'utilisateur pourra alors écrire des expressions de la forme suivante $(a=4)(b=a/2)a+b*(3-1)$
1. Modifiez le code de traduction de la notation standard pour qu'il fonctionne avec l'opérateur égal
 - L'expression $(a=4)(b=a/2)a+b*(3-1)$ sera transformée en $a4=ba2/=ab31-*+$
 2. Modifiez la fonction calcul pour qu'elle utilise deux variables a et b et qu'elle exécute les expressions générées précédemment (un opérateur égal modifiera la valeur de la variable ciblée avec la valeur trouvée dans la pile).
 - L'expression $a4=a1+$ renverra la valeur 5.
 - L'expression $a4=b5=ab-$ renverra la valeur 1
 - L'expression $a4=ba2/=ab31-*+$ renverra la valeur 4
 3. Modifiez le programme pour qu'il fonctionne avec n'importe quelle lettre de l'alphabet (Indice: on rappelle que 'c' - 'a' = 3)

Partie III Avancé

6. Fonctions

Le but de cet exercice est de donner à l'utilisateur la possibilité de créer une fonction. Nous proposerons d'utiliser l'opérateur #

- Modifiez le programme pour qu'il demande à l'utilisateur d'entrer une expression sous forme NPI pour cet opérateur et qui l'enregistre dans une chaîne de caractères
 - L'utilisateur pourra par exemple dire que l'expression associée à l'opérateur # est $4+$
- Modifiez la fonction de traduction pour qu'elle fonctionne avec cet opérateur (il sera traité avec la même priorité que les multiplications).
- Modifiez le programme pour qu'il demande à l'utilisateur de saisir une expression utilisant cet opérateur.
- Modifiez la fonction calcul pour qu'elle accepte le tableau de variables en argument.
- Modifiez la fonction pour qu'elle prenne en compte l'opérateur #:
 - L'opérateur # est traité comme un nombre ou une variable et ajouté dans la pile comme tel.
 - Lorsqu'on souhaite l'additionner/le soustraire/le multiplier/le diviser/le comparer ou autre on calcule sa valeur. Pour cela on appelle la fonction calcul récursivement.

Vérifiez que pour $\# = 1+$, l'expression $4\#1+$ renvoie 6 Essayez pour $\# = 23?$

7. Programmation

- Trouvez une valeur de # pour laquelle la fonction permettra de calculer la factorielle d'un nombre (Indice: utilisez la récursion)