TP réalisation d'une calculatrice

Ce mini-projet consiste à réaliser en binôme une calculatrice en script. Afin de vous guider, le projet est scindé en plusieurs questions dont l'ordre est à suivre impérativement. Chaque partie devra faire l'objet d'un script à remettre. Au final, vous obtiendrez le script permettant un calcul de nombres entiers proposé par l'utilisateur. Ce calcul devra être de la forme :

```
./calc.sh \( 2 + 3 \) x \( 2 - 4 \)
```

Pour simplifier le projet chaque élément du calcul devra être séparé par un espace. La parenthèse devra être précédée par un "\" afin d'éviter qu'elle soit interprétée comme un début de commande SHELL. Pour rappel le "\" est un caractère d'échappement. De même le caractère "*" a une signification spéciale pour le SHELL (l'échappement ne fonctionne malheureusement pas dans ce cas), il est donc important d'utiliser le caractère "x" pour spécifier la multiplication.

Exercice 1 :

Ecrire un script nommé *question*1.*sh* permettant d'effectuer un calcul simple. N'oubliez pas de mettre des espaces pour séparer chaque caractère. Votre script devra contenir les fonctions effectuant les opérations de base : addition, soustraction, multiplication et division. Il n'est pas demandé dans cet exercice de procéder aux vérifications des paramètres envoyés au script. On se limitera à deux opérandes.

```
Exemple  :  $./question1.sh  3  2
            somme 3 2 = 5
            sous 3 2 = 1
            mult 3 2 = 6
            div 3 2 = 1
```

Exercice 2 :

Ecrire un script nommé *question*2.*sh* permettant de faire l'analyse des paramètres donnés au lancement du script. Ce script devra identifier la nature de chaque élément. Pour simplifier l'exercice on admettra que tout autre caractère que les parenthèses ou les opérateurs est un entier. On se limitera à deux opérandes.

```
$./question2.sh \( 2 + 3 \)
para ouvrante
entier
somme
entier
para fermante
```

Exercice 3 :

Ecrire un script (à partir des deux exercices précédents) nommé *question*3.*sh* permettant d'effectuer un calcul simple. N'oubliez pas de mettre des espaces pour séparer chaque caractère.

Votre script devra contenir les fonctions effectuant les opérations de base : addition, soustraction, multiplication et division. Il n'est pas demandé dans cet exercice de procéder aux vérifications des paramètres envoyés au script. La gestion des parenthèses n'est pas à faire non plus à ce niveau. On se limitera à deux opérandes pour le moment.

```
Exemple 1:  $./question3.sh  2  +  3
            5

Exemple 2:  $./question3.sh  2  x  3
            6
```

Exercice 4 :

Ecrire un script nommé *question*4.*sh* capable de vérifier la syntaxe du calcul demandé par l'utilisateur. Cette syntaxe doit respecter une alternance entre les opérations et les entiers. Le script devra comporter une fonction capable de détecter une mauvaise alternance.

```
Exemple 1:  $./question4.sh  2  +  +  3
            entier
            somme
            somme
            ** erreur d'alternance ! **

Exemple 2:  $./question4.sh   2  2 + 3
            entier
            entier
            ** erreur d'alternance ! **
```

Exercice 5 :

Ecrire un script nommé *question*5.*sh* capable de vérifier la syntaxe du calcul demandé par l'utilisateur en tenant compte du parenthèsage. Cette syntaxe doit respecter une alternance entre les opérations, les entiers et les parenthèses. Le script devra comporter une fonction capable de détecter une mauvaise alternance.

Pour simplifier le travail, un mauvais parenthèsage ne sera notifié qu'à la fin de la vérification de l'alternance.

```
Exemple 1:  $./question5.sh  \(  \( 2  +  3  \)
            para ouvrante
              para ouvrante
            entier
            somme
            entier
            para fermante
            ** erreur parenthèse ! **

Exemple 2:  $./question5.sh  \(  2  +  \(  3  \)
            para ouvrante
            entier
            somme
            para ouvrante
            entier
            para fermante
            ** erreur parenthèse ! **
```

Exercice 6 :

A partir du script *question*3.*sh*, proposez un script *question*6.*sh* permettant de gérer plusieurs opérandes. Il n'est pas demandé de gérer la priorité des opérateurs.

```
Exemple 1:  $./question6.sh  2  +  3  x  4
            résultat = 20

Exemple 2:  $./question6.sh  2  x  3  +  2
            résultat = 8
```

Exercice 7 :

En vous servant de tous les exercices précédents, écrire le script nommé *calc.sh* permettant de simuler une calculatrice simple. Pour rappel, cette calculatrice ne doit pas gérer la priorité des opérateurs. D'autre part, pour simplifier, on ne tiendra compte que d'un seul niveau de parenthèsage. Le script pourra générer une erreur dans le cas d'un parenthèsage à plusieurs niveaux.

Exemple à ne pas traiter :

```
./calc.sh  \(  2  +  3  \)  x  \(  \( 2  -  4 \)  +  3  \)
```

Exemple à traiter :

```
./calc.sh  \(  2  +  3  \)  x  \(  2  -  4  \)
résultat = -10
```

Il est à noter que la priorité des opérateurs peut être gérée par l'intermédiaire des parenthèses.

```
./calc.sh  2  +  3  x  4
résultat = 20

./calc.sh  2  +  \(3  x  4 \)
résultat = 14
```

Exercice 8 :

Proposez une méthode pour gérer plusieurs niveaux de parenthèsage. Modifier le script précédent pour permettre une gestion du parenthèsage globale.

Vous pouvez également implémenter la gestion de la priorité entre opérateur.