# T.P. 7 Calculatrice (partie 4)

Pour ce TP, le sous-programme **GetInput** est mis à votre disposition. Il permet de récupérer une chaîne de caractères saisie par l'utilisateur. **GetInput** contient les entrées suivantes :

Entrées : **A0.L** pointe sur un *buffer* de 60 octets où sera stockée la chaîne après la saisie.

**D1.B** contient le numéro de colonne où sera affichée la chaîne à saisir.

**D2.B** contient le numéro de ligne où sera affichée la chaîne à saisir.

**D3.L** contient l'indice de temps d'attente avant une première répétition.

**D4.L** contient l'indice de temps d'attente après une première répétition.

Le *buffer* sera réservé par la directive d'assemblage DS.B.

- Tapez le code source ci-dessous et sauvegardez-le sous le nom "GetInputTest.asm".
- Copiez le fichier "GetInput.bin" dans le même dossier.

```
Initialisation des vecteurs
                    $0
                    $ffb500
              dc.l
vector_000
vector_001
              dc.l
                    Main
               ______
               Programme principal
               ______
                    $500
              огд
Main
              movea.l #sBuffer,a0
              clr.b
              clr.b
                    d2
              move.l #60000,d3
              move.l #8000,d4
                    GetInput
              jsr
              illegal
               ______
               Sous-programmes
               _____
              incbin "GetInput.bin"
GetInput
              ; =============
               Données
               _____
sBuffer
              ds.b
                    60
```

T.P. 7

• Exécutez ce code à l'aide du débogueur en activant la fenêtre de sortie vidéo. Saisissez une chaîne de caractères puis appuyez sur la touche [Entrée]. Observez alors ce que contient la mémoire à l'adresse buffer. Recommencez jusqu'à bien comprendre le fonctionnement de GetInput. Attention, il n'est pas demandé d'exécuter pas à pas le contenu de GetInput afin de comprendre son fonctionnement interne, mais simplement de savoir l'utiliser.

### **Remarque:**

Les paramètres dans **D3** et **D4** doivent être ajustés en fonction de la puissance de votre ordinateur. Si la répétition des touches est trop rapide, vous devrez augmenter ces valeurs.

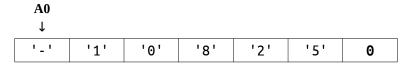
## Étape 1

Réalisez le sous-programme **Itoa** qui convertit une valeur numérique entière, codée sur 16 bits signés, en une chaîne de caractères ASCII.

<u>Entrées</u> : **A0.L** pointe sur un *buffer* où sera stockée la chaîne après la conversion.

**D0.W** contient une valeur numérique entière signée à convertir.

Par exemple, si  $\mathbf{D0.W} = -10825$ , la chaîne suivante devra être placée à l'adresse pointée par  $\mathbf{A0}$ :



#### **Indications:**

- Si le nombre est positif, il suffit d'appeler le sous-programme **Uitoa**.
- Si le nombre est négatif, il faut positionner le caractère '-' dans la chaîne puis appeler le sous-programme **Uitoa** avec l'opposé du nombre à convertir.

## <u>Étape 2</u>

Réalisez le programme Main de la calculatrice en respectant l'affichage de l'exemple ci-dessous :

```
Veuillez saisir une expression :
2*5-5*2+18/2
Resultat :
14
```

## <u>Étape 3</u>

Modifiez votre programme de façon à prendre en compte la priorité des opérateurs. Utilisez la méthode de votre choix.

```
Veuillez saisir une expression :
2*5-5*2+18/2
Resultat :
9
```

T.P. 7