

# 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
; =====

org      $0
vector_000 dc.l    $ffb500
vector_001 dc.l    Main

; =====
; Programme principal
; =====

org      $500
Main     movea.l #sBuffer,a0
         clr.b   d1
         clr.b   d2
         move.l  #60000,d3
         move.l  #8000,d4
         jsr     GetInput

         illegal

; =====
; Sous-programmes
; =====

GetInput incbin  "GetInput.bin"

; =====
; Données
; =====

sBuffer  ds.b    60

```

- 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.

Entrées : **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 **D0.W** = -10825, la chaîne suivante devra être placée à l'adresse pointée par **A0** :

<b>A0</b>						
↓						
' - '	' 1 '	' 0 '	' 8 '	' 2 '	' 5 '	0

### 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.

### Étape 2

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
```

### Étape 3

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
```