

T.P. 5 – Corrigé

Calculatrice (partie 2)

Étape 1

```

Convert      ; Si la chaîne est nulle,
              ; on quitte en renvoyant false (erreur).
tst.b    (a0)
beq      \false

              ; (À ce stade, la chaîne n'est pas nulle.)
              ; S'il existe une erreur sur les caractères,
              ; on quitte en renvoyant false (erreur).
jsr      IsCharError
beq      \false

              ; (À ce stade, la chaîne n'est pas nulle
              ; et ne contient que des chiffres.)
              ; Si le nombre que contient la chaîne est supérieur à 32767,
              ; on quitte en renvoyant false (erreur).
jsr      IsMaxError
beq      \false

              ; La chaîne est valide, il ne reste plus qu'à la convertir
              ; puis à quitter en renvoyant true (aucune erreur).
jsr      Atoui

\true        ; Sortie qui renvoie Z = 1 (aucune erreur).
ori.b    #%00000100,CCR
rts

\false       ; Sortie qui renvoie Z = 0 (erreur).
andi.b   #%11111011,CCR
rts

```

Étape 2

```

Print      ; Sauvegarde les registres dans la pile.
           movem.l d0/d1/a0,-(a7)

\loop      ; Charge un caractère de la chaîne dans D0.
           ; Si le caractère est nul, il s'agit de la fin de la chaîne.
           ; On peut sortir du sous-programme.
           move.b (a0)+,d0
           beq     \quit

           ; Affiche le caractère.
           jsr     PrintChar

           ; Incrémente la colonne d'affichage du caractère,
           ; et reboucle.
           addq.b #1,d1
           bra     \loop

\quit      ; Restaure les registres puis sortie.
           movem.l (a7)+,d0/d1/a0
           rts

```

Étape 3

```

NextOp     ; Si le caractère est nul (fin de chaîne),
           ; il n'y a pas d'opérateur dans la chaîne.
           ; A0 pointe sur le caractère nul. On quitte.
           tst.b  (a0)
           beq    \quit

           ; Comparaisons successives du caractère aux 4 opérateurs.
           ; Si le caractère est un opérateur, on peut quitter.
           ; (A0 contient l'adresse de l'opérateur.)
           cmpi.b #'',(a0)
           beq    \quit

           cmpi.b #'-',(a0)
           beq    \quit

           cmpi.b #'*',(a0)
           beq    \quit

           cmpi.b #'/',(a0)
           beq    \quit

           ; Passage au caractère suivant.
           addq.l #1,a0
           bra    NextOp

\quit      ; Sortie.
           rts

```