T.P. 2 – Corrigé
Calculatrice (partie 1)

Étape 1

```
DelSpace            ; Sauvegarde les registres.
                    movem.l a0/a1,-(a7)

                    ; A1 = pointeur sur la chaîne destination
                    ; (même chaîne que la source).
                    movea.l a0,a1

\loop               ; Si fin de chaîne, on quitte.
                    tst.b   (a0)
                    beq.s   \quit

                    ; Si le caractère est un espace,
                    ; on ne le copie pas.
                    cmpi.b  #' ',(a0)
                    beq.s   \noCopy

                    ; Sinon, on copie le caractère dans la destination
                    ; et on incrémente le pointeur destination.
\copy               move.b  (a0),(a1)+

\noCopy             ; Passage au caractère suivant
                    ; en incrémentant le pointeur source
                    ; puis en rebouclant.
                    addq.l  #1,a0
                    bra.s   \loop

\quit               ; Ne pas oublier le caractère nul
                    ; en fin de chaîne destination.
                    clr.b   (a1)

                    ; Restaure les registres puis sortie.
                    movem.l (a7)+,a0/a1
                    rts
```

Étape 2

```
ErrChar             ; Sauvegarde les registres.
                    move.l  a0,-(a7)

\loop               ; Compare le caractère de la chaîne au caractère '0'.
                    ; S'il est inférieur, on renvoie une erreur.
                    cmpi.b  #'0',(a0)
                    blt.s   \true

                    ; Compare le caractère de la chaîne au caractère '9'.
                    ; S'il est supérieur, on renvoie une erreur.
                    ; (On fait pointer A0 sur le caractère suivant.)
                    cmpi.b  #'9',(a0)+
                    bgt.s   \true

                    ; Fin de chaîne ?
                    ; Si non, on reboucle.
                    ; Si oui, on quitte sans erreur.
                    tst.b   (a0)
                    bne.s   \loop

\false              ; Sortie qui renvoie Z = 0 (aucune erreur).
                    ; (L'instruction BRA ne modifie pas le flag Z.)
                    andi.b  #%11111011,ccr
                    bra.s   \quit

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

\quit               ; Restaure les registres puis sortie.
                    ; (Les instructions MOVEA et RTS ne modifient pas le flag Z.)
                    movea.l (a7)+,a0
                    rts
```

Étape 3

```
ErrMax              ; Sauvegarde les registres.
                    movem.l d0/a0,-(a7)

                    ; On récupère la taille de la chaîne (dans D0)
                    ; et on la compare à 5 caractères.
                    jsr     StrLen
                    cmpi.l  #5,d0

                    ; Si la chaîne contient plus de 5 caractères,
                    ; on quitte en renvoyant une erreur.
                    bgt.s   \true

                    ; Si la chaîne contient moins de 5 caractères,
                    ; on quitte sans renvoyer d'erreur.
                    blt.s   \false

                    ; Si la chaîne contient exactement 5 caractères :
                    ; comparaisons successives de '3', '2', '7', '6' et '7'
                    ; Si supérieur, on quitte en renvoyant une erreur.
                    ; Si inférieur, on quitte sans renvoyer d'erreur.
                    ; Si égal, on compare le caractère suivant.
                    cmpi.b  #'3',(a0)+
                    bgt.s   \true
                    blt.s   \false

                    cmpi.b  #'2',(a0)+
                    bgt.s   \true
                    blt.s   \false

                    cmpi.b  #'7',(a0)+
                    bgt.s   \true
                    blt.s   \false

                    cmpi.b  #'6',(a0)+
                    bgt.s   \true
                    blt.s   \false

                    cmpi.b  #'7',(a0)+
                    bgt.s   \true

\false              ; Sortie qui renvoie Z = 0 (aucune erreur).
                    ; (L'instruction BRA ne modifie pas le flag Z.)
                    andi.b  #%11111011,ccr
                    bra.s   \quit

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

\quit               ; Restaure les registres puis sortie.
                    ; (Les instructions MOVEM et RTS ne modifient pas le flag Z.)
                    movem.l (a7)+,d0/a0
                    rts
```

Étape 4

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

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

                    ; La chaîne est valide, il ne reste plus qu'à la convertir
                    ; puis à quitter sans renvoyer d'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 5

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

\loop               ; Copie le caractère à afficher dans D0.
                    move.b  (a0)+,d0

                    ; Si le caractère est nul, il s'agit d'une fin de chaîne.
                    ; On peut sortir du sous-programme.
                    beq.s   \quit

                    ; Affiche le caractère.
                    jsr     PrintChar

                    ; Incrémente la colonne d'affichage du caractère.
                    addq.b  #1,d1

                    ; Passe au caractère suivant.
                    bra.s   \loop

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