

# T.P. 2 – Corrigé

## Les branchements et les boucles

### Étape 1

```

    org     $4
Vector_001 dc.l   Main

    org     $500
Main      clr.l   d1           ; 0 -> D1
          move.l  #$80000007,d0 ; $80000007 -> D0.L (D0.W = $0007 = 7)
loop1     addq.l  #1,d1       ; D1 + 1 -> D1
          subq.w  #1,d0       ; D0.W - 1 -> D0.W ; Seul D0.W est décrémenté.
          bne    loop1        ; Saut si Z = 0 (D0.W ≠ 0)
          ; D1 = 7

          clr.l   d2           ; 0 -> D2
          move.l  #$fe2310,d0  ; $fe2310 -> D0.L (D0.B = $10 = 16)
loop2     addq.l  #1,d2       ; D2 + 1 -> D2
          subq.b  #2,d0       ; D0.B - 2 -> D0.B ; Seul D0.B est décrémenté.
          bne    loop2        ; Saut si Z = 0 (D0.B ≠ 0)
          ; D2 = 8

          clr.l   d3           ; 0 -> D3
loop3     moveq.l #125,d0     ; 125 -> D0
          addq.l  #1,d3       ; D3 + 1 -> D3
          dbra   d0,loop3     ; DBRA = DBF
          ; D0.W - 1 -> D0.W
          ; Saut si D0.W ≠ -1 (D0.W ≠ $FFFF)
          ; D3 = 126

          clr.l   d4           ; 0 -> D4
loop4     moveq.l #10,d0      ; 10 -> D0
          addq.l  #1,d4       ; D4 + 1 -> D4
          addq.l  #1,d0       ; D0 + 1 -> D0
          cmpi.l  #30,d0      ; Compare D0 à la valeur 30.
          bne    loop4        ; Saut si Z = 0 (D0.L ≠ 30)
          ; D4 = 20

          illegal

```

**Étape 2**

```

VALUE      equ      18

           org      $4

Vector_001 dc.l     Main

           org      $500

Main       move.b   #VALUE,d1

           tst.b    d1          ; Mise à jour de N et de Z en fonction de D1.B
           bne     next1       ; Si Z = 0 (D1.B ≠ 0), saut à Next1
           move.l  #200,d0     ; Sinon (D1.B = 0), 200 -> D0.L
           bra     quit        ; Sortie

next1      bmi     next3       ; Si N = 1 (D1.B < 0), saut à Next3
           cmp.b   #$61,d1     ; Sinon (D1.B ≥ 0), D1.B est comparé à $61 ($61 = 97)
           blt     next2       ; Si D1.B < $61, saut à Next2
           move.l  #400,d0     ; Sinon (D1.B ≥ $61), 400 -> D0.L
           bra     quit        ; Sortie

next2      move.l  #600,d0     ; D1.B < $61, 600 -> D0.L
           bra     quit        ; Sortie

next3      move.l  #800,d0     ; D1.B < 0, 800 -> D0.L

quit      illegal

```

1. Quelle valeur renvoie le programme lorsque l'étiquette VALUE est initialisée à la valeur 18 ?

Le programme renvoie la valeur **600**.

2. Quelle valeur renvoie le programme lorsque l'étiquette VALUE est initialisée à la valeur -5 ?

Le programme renvoie la valeur **800**.

3. Quelle valeur renvoie le programme lorsque l'étiquette VALUE est initialisée à la valeur 0 ?

Le programme renvoie la valeur **200**.

4. Quelle valeur renvoie le programme lorsque l'étiquette VALUE est initialisée à la valeur 96 ?

Le programme renvoie la valeur **600**.

**Étape 3**

```

                org     $4
Vector_001     dc.l     Main

                org     $500
Main          ; Initialise D0.
              move.l   #-1,d0

Abs          ; Mise à jour des flags Z et N en fonction de D0.
              ; Si D0 est positif ou nul, alors N = 0.
              ; Si D0 est négatif, alors N = 1.
              tst.l    d0

              ; Saut à quit si N = 0 (donc si D0 est positif).
              bpl     quit

              ; Sinon N = 1 (donc D0 est négatif).
              ; 0 - D0 -> D0
              neg.l   d0

quit         ; Arrêt du programme.
              illegal

```

**Étape 4**

```

                org     $4
Vector_001     dc.l     Main

                org     $500
Main          ; Initialise A0 avec l'adresse de la chaîne.
              movea.l  #STRING,a0

StrLen       ; Initialise le compteur de caractères à 0
              ; (D0 = compteur de caractères).
              clr.l    d0

loop        ; On teste si le caractère de la chaîne est nul.
              ; On profite également de ce test pour faire pointer
              ; le registre A0 sur le caractère suivant.
              tst.b    (a0)+

              ; Si le caractère testé est nul, il s'agit de la fin de la chaîne.
              ; On peut quitter.
              beq     quit

              ; Sinon, on incrémente le compteur de caractères.
              ; Puis on reboucle.
              addq.l   #1,d0
              bra     loop

quit        ; Arrêt du programme.
              illegal

                org     $550
STRING       dc.b     "Cette chaîne comporte 36 caractères.",0

```

**Étape 5**

```

                org     $4
Vector_001     dc.l     Main

                org     $500
Main           ; Initialise A0 avec l'adresse de la chaîne.
               movea.l #STRING,a0

SpaceCount    ; Initialise le compteur d'espaces à 0.
               ; (D0 = compteur d'espaces).
               clr.l   d0

loop          ; Le caractère de la chaîne est chargé dans D1.
               ; L'instruction MOVE modifie les flags de la
               ; même façon que l'instruction TST.
               ; Par conséquent :
               ; - si D1 est non nul, Z est mis à 0 ;
               ; - si D1 est nul, Z est mis à 1.
               ; On peut dès lors utiliser l'instruction BEQ
               ; qui sautera à quit si Z = 1 (c'est-à-dire si D1 = 0).
               move.b  (a0)+,d1
               beq    quit

               ; Si le caractère contenu dans D1 n'est pas un espace,
               ; alors on reboucle.
               cmp.b  #' ',d1
               bne   loop

               ; Sinon, le caractère est un espace.
               ; On incrémente le compteur d'espaces,
               ; puis on reboucle.
               addq.l #1,d0
               bra    loop

quit          ; Sortie du programme.
               illegal

                org     $550
STRING       dc.b     "Cette chaîne comporte 4 espaces.",0

```

**Étape 6**

```

                org     $4
Vector_001     dc.l     Main

                org     $500
Main           ; Initialise A0 avec l'adresse de la chaîne.
               movea.l #STRING,a0

LowerCount    ; Initialise le compteur de minuscules à 0.
               ; (D0 = compteur de minuscules).
               clr.l   d0

loop          ; Le caractère de la chaîne est chargé dans D1.
               ; L'instruction MOVE modifie les flags de la
               ; même façon que l'instruction TST.
               ; Par conséquent :
               ; - si D1 est non nul, Z est mis à 0 ;
               ; - si D1 est nul, Z est mis à 1.
               ; On peut dès lors utiliser l'instruction BEQ
               ; qui sautera à quit si Z = 1 (c'est-à-dire si D1 = 0).
               move.b  (a0)+,d1
               beq     quit

               ; Si le code ASCII du caractère est inférieur
               ; à celui de 'a', le caractère n'est pas une minuscule.
               ; On reboucle.
               cmp.b   #'a',d1
               blo     loop

               ; Si le code ASCII du caractère est supérieur
               ; à celui de 'z', le caractère n'est pas une minuscule.
               ; On reboucle.
               cmp.b   #'z',d1
               bhi     loop

               ; Sinon, le caractère est une minuscule.
               ; On incrémente le compteur de minuscules,
               ; puis on reboucle.
               addq.l  #1,d0
               bra     loop

quit          ; Sortie du programme.
               illegal

                org     $550
STRING        dc.b     "Cette chaîne comporte 28 minuscules.",0

```