

T.P. 2

Les branchements et les boucles

Étape 1

- Déterminez manuellement (sans l'aide du débogueur) les valeurs que prendront les registres **D1**, **D2**, **D3** et **D4** en sortie des boucles suivantes :

```

                                org      $4
Vector_001  dc.l      Main

                                org      $500
Main
                                clr.l     d1
loop1      move.l     #$80000007,d0
                                addq.l    #1,d1
                                subq.w    #1,d0
                                bne       loop1

                                clr.l     d2
loop2      move.l     #$fe2310,d0
                                addq.l    #1,d2
                                subq.b    #2,d0
                                bne       loop2

                                clr.l     d3
loop3      moveq.l    #125,d0
                                addq.l    #1,d3
                                dbra      d0,loop3      ; DBRA = DBF

                                clr.l     d4
loop4      moveq.l    #10,d0
                                addq.l    #1,d4
                                addq.l    #1,d0
                                cmpi.l    #30,d0
                                bne       loop4

                                illegal

```

- Assemblez et exécutez le programme ci-dessus afin de vérifier que vos réponses à la question précédente sont correctes.

Étape 2

Soit le programme ci-dessous :

VALUE	equ	18
	org	\$4
Vector_001	dc.l	Main
	org	\$500
Main	move.b	#VALUE,d1
	tst.b	d1
	bne	next1
	move.l	#200,d0
	bra	quit
next1	bmi	next3
	cmp.b	#\$61,d1
	blt	next2
	move.l	#400,d0
	bra	quit
next2	move.l	#600,d0
	bra	quit
next3	move.l	#800,d0
quit	illegal	

Ce programme charge une valeur dans le registre **D0.L** (registre de sortie du programme) en fonction d'une valeur contenue dans le registre **D1.B** (registre d'entrée du programme). La valeur qui sera chargée dans le registre **D1.B** est initialisée au début du code source à l'aide de l'étiquette **VALUE**.

Déterminez manuellement (sans l'aide du débogueur) les réponses aux questions suivantes :

1. Quelle valeur renvoie le programme lorsque l'étiquette **VALUE** est initialisée à la valeur 18 ?
2. Quelle valeur renvoie le programme lorsque l'étiquette **VALUE** est initialisée à la valeur -5 ?
3. Quelle valeur renvoie le programme lorsque l'étiquette **VALUE** est initialisée à la valeur 0 ?
4. Quelle valeur renvoie le programme lorsque l'étiquette **VALUE** est initialisée à la valeur 96 ?

Assemblez et exécutez le programme ci-dessus pour chaque valeur de l'étiquette **VALUE** afin de vérifier que vos réponses sont correctes.

Étape 3

Réalisez le programme **Abs** qui renvoie la valeur absolue d'un entier signé.

Entrée : **D0.L** = Entier signé sur 32 bits.

Sortie : **D0.L** = $|\mathbf{D0.L}|$

Vous utiliserez la structure suivante pour tester votre programme (essayez plusieurs valeurs significatives dans le registre **D0**).

```

                                org      $4
Vector_001 dc.l      Main

                                org      $500
Main      move.l    #-1,d0      ; Initialise D0.
Abs       ; ...              ; Programme Abs à développer.
          ; ...              ; En sortie du programme, D0.L doit contenir
          ; ...              ; la valeur absolue de sa valeur initiale.

          illegal

```

Indication : Jetez un coup d'œil à l'instruction NEG.

Étape 4

Réalisez le programme **StrLen** qui renvoie la taille d'une chaîne de caractères. Une chaîne de caractères se termine par un caractère nul.

Entrée : **A0.L** pointe sur le premier caractère d'une chaîne de caractères.

Sortie : **D0.L** renvoie le nombre de caractères de la chaîne (sans le caractère nul).

Vous utiliserez la structure suivante pour tester votre programme :

```

                                org      $4
Vector_001 dc.l      Main

                                org      $500
Main      movea.l   #STRING,a0   ; Initialise A0 avec l'adresse de la chaîne.
StrLen    ; ...              ; Programme StrLen à développer.
          ; ...              ; En sortie du programme, D0.L doit contenir
          ; ...              ; la taille de la chaîne.

          illegal

                                org      $550
STRING    dc.b      "Cette chaine comporte 36 caracteres.",0

```

Remarque : Afin d'éviter tout problème lié à l'encodage des caractères, ne pas mettre d'accents dans les chaînes de caractères.

Repérer où se trouve la chaîne de caractères en mémoire à l'aide de l'onglet **[Mémoire]**.

Étape 5

Réalisez le programme **SpaceCount** qui renvoie le nombre d'espaces dans une chaîne de caractères. Une chaîne de caractères se termine par un caractère nul.

Entrée : **A0.L** pointe sur le premier caractère d'une chaîne de caractères.

Sortie : **D0.L** renvoie le nombre d'espaces de la chaîne.

Vous utiliserez la structure suivante pour tester votre programme :

```

                org      $4
Vector_001     dc.l      Main

                org      $500
Main          movea.l    #STRING,a0 ; Initialise A0 avec l'adresse de la chaîne.
SpaceCount    ; ...          ; Programme SpaceCount à développer.
              ; ...          ; En sortie du programme, D0.L doit contenir
              ; ...          ; le nombre d'espaces de la chaîne.

              illegal

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

```

Indication : Le code ASCII du caractère *espace* peut être obtenu à l'aide de la syntaxe `#' '`.

Remarque : Afin d'éviter tout problème lié à l'encodage des caractères, ne pas mettre d'accents dans les chaînes de caractères.

Étape 6

Réalisez le programme **LowerCount** qui renvoie le nombre de lettres minuscules dans une chaîne de caractères. Une chaîne de caractères se termine par un caractère nul.

Entrée : **A0.L** pointe sur le premier caractère d'une chaîne de caractères.

Sortie : **D0.L** renvoie le nombre de lettres minuscules de la chaîne.

Vous utiliserez la structure suivante pour tester votre programme :

```

                org      $4
Vector_001     dc.l      Main

                org      $500
Main          movea.l    #STRING,a0 ; Initialise A0 avec l'adresse de la chaîne.
LowerCount    ; ...          ; Programme LowerCount à développer.
              ; ...          ; En sortie du programme, D0.L doit contenir
              ; ...          ; le nombre de lettres minuscules de la chaîne.

              illegal

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

```

Indications :

- Le code ASCII du caractère *a* peut être obtenu à l'aide de la syntaxe `#'a'`.
- Le code ASCII du caractère *z* peut être obtenu à l'aide de la syntaxe `#'z'`.
- Une lettre est minuscule si son code ASCII est compris entre les codes ASCII de *a* et de *z*.

Remarque : Afin d'éviter tout problème lié à l'encodage des caractères, ne pas mettre d'accents dans les chaînes de caractères.