```
; Programme de démarrage TP1.S
.include "p24FJ256GB106.inc" ; Définitions pour ucontrôleur
.equiv PLL96DIS_ON,
                      0xF7FF ; Enabled
; Définition des paramètres de configuration (horloges, USB, watchdog)
; CONFIG 1 : JTAG off, watchdog timer off
config __CONFIG1, JTAGEN_OFF & ICS_PGx2 & FWDTEN_OFF
; CONFIG 2 : Using the 12MHz clock provided by the PIC18F67J50 (debug)
config __CONFIG2, IESO_OFF & PLL96DIS_ON & PLLDIV_DIV3 & FNOSC_PRIPLL & POSCMOD_HS
; Ces paramètres conduisent à une fréquence horloge système de 32 MHz
                               ;The label for the first line of code.
       .global __reset
;Declaration du mecanisme d'interuption en fonction d'un Timer
        .global __TlInterrupt ;Declare Timer 1 ISR name global
;Code Section in Program Memory
;Start of Code section
.text
    ORG 000H ; a reset redirects program to this point
    GOTO __reset
    ORG 004H ; an interrupt redirects the program to here
    GOTO __TlInterrupt
;Affiche une ligne a l'ordonnee W1 d'intensite W2 (modifie W0)
_TlInterrupt:
       BCLR IFS0, #T1IF
                           ;Clear interrupt flag
       MOV #0XFFFF, W9
       MOV W4, W0
       MOV W5, W1
       MOV W6, W2
       CALL PutPixel
       ;On laisse le pixel affiche un certain temps
       MOV #5, W0
       CALL DelayMs
       ;On eteints le pixel
       MOV W4, W0
       MOV W5, W1
       MOV #0, W2
       CALL PutPixel
       ;On decale la position de 1
       ADD W4, #1, W4
       MOV #132, W13
       RETFIE
                           ;Return from Interrupt Service routine
LINE_DELAY:
       ;On sauvegarde les registres a utiliser
       MOV #0, W4
       MOV W1, W5
       MOV W2, W6
       ;On place les parametres de PutPixel
       \textbf{MOV} \ \ \text{W4} \ , \ \ \text{W0}
       MOV W5, W1
       MOV W6, W2
       CALL PutPixel
       ;On laisse le pixel affiche un certain temps
       MOV #5, W0
       CALL DelayMs
       ;On eteints le pixel
       MOV W4, W0
       MOV W5, W1
       MOV #0, W2
       CALL PutPixel
       ;On decale la position de 1
       ADD W4, #1, W4
       MOV #132, W13
       SUB W4, W13, W13
       BRA NZ, TqL
```

RETURN

```
; Affiche une ligne a l'ordonnee W1 d'intensite W2 (modifie W0)
LINE_INPUT:
       ;On sauvegarde les registres a utiliser
       MOV #0, W4
       MOV W1, W5
       MOV W2, W6
   TqLI:
       ;On place les parametres de PutPixel
       \textbf{MOV} \ \ \text{W4} \, , \ \ \text{W0} \\
       MOV W5, W1
       MOV W6, W2
       CALL PutPixel
       ;On laisse le pixel affiche un certain temps
       BSET AD1PCFG, #1
   TNb:MOV #10000, W0
   TsI:BTST PORTB, #1
       BRA Z, TsI
       SUB W0, #1, W0
       BRA NZ, TsI
       ;On eteints le pixel
       MOV W4, W0
       MOV W5, W1
       MOV #0, W2
       CALL PutPixel
       ;On decale la position de 1
       ADD W4, #1, W4
       MOV #132, W13
       SUB W4, W13, W13
       BRA NZ, TqLI
       RETURN
;Affiche une ligne qui balaie l'ecran a une frequence WO
BALAY:
       MOV #0, W5
       CALL Timer23_on
   TaB:
       ;On place les parametres pour faire un balayage sur une ligne
       MOV W5, W1
       MOV #0xFFFF, W2
       CALL LINE_INPUT
       ;On change de ligne
       ADD W5, #1, W5
       MOV #64, W13
       SUB W5, W13, W13
       BRA NZ, TqB
       ;Si on a atteint la limite inferieure, on remonte
       MOV #0, W5
       CALL Timer23_off
       BRA TqB
       RETURN
reset:
       MOV #__SP_init, W15
                                ;Initalize the Stack Pointer
       MOV #__SPLIM_init, W0
                                ; Initialize the Stack Pointer Limit Register
       MOV W0, SPLIM
       BSET IECO, #1
       BSET IPC0, #7
       CALL ResetDevice
       CALL ClearDevice
;Exercice 1: Balayage par programme
       BRA $ ; boucle sans fin
.end
                                  ; End of program code in this file
```