


```

        .ORIG x3000      ; adresse de début de programme
; partie dédiée au code
        LEA R0,debut    ; charge l'adresse du premier élément dans R1
        LEA R1,fin      ; charge l'adresse du dernier élément dans R4
        JSR mul2        ; appel la routine add
        HALT           ; termine le programme

; partie dédiée aux données
debut:  .FILL #1
        .FILL #3
        .FILL #5
        .FILL #7
fin:    .FILL #9

; Routine mul2, pour multiplier par 2 le contenu des cases dont les adresses
; sont comprises entre l'adresse contenue dans R0 et celle contenue dans R1.
; On suppose que R0 <= R1
; paramètres d'entrée : R0 et R1, adresses de début et de fin du tableau
; registres temporaires : R2 registre dans lequel on effectuera des calculs
;
;                               R3 servira de pointeur pour parcourir les cases du tableau
;                               R4 contiendra l'opposé de l'adresse contenue dans R1
mul2:   NOT R4,R1
        ADD R4,R4, 1    ; R4 <- -R1
        ADD R3,R0,0    ; R3 <- R0
loop:   ADD R2,R3,R4    ; R2 <- R3+R4 = R3-R1
        BRp endloop    ; si R3-R1 > 0, i.e., R3>R1, on sort de la boucle
        LDR R2,R3,0    ; R2 <- Mem[R3]
        ADD R2,R2,R2    ; R2 <- 2*R2
        STR R2,R3,0    ; Mem[R3] <- R2
        ADD R3,R3,1    ; R3++
        BR loop        ; on boucle
endloop: RET           ; retour à l'appelant
        .END

```