RIVET Mathias
FONTANA Lucas

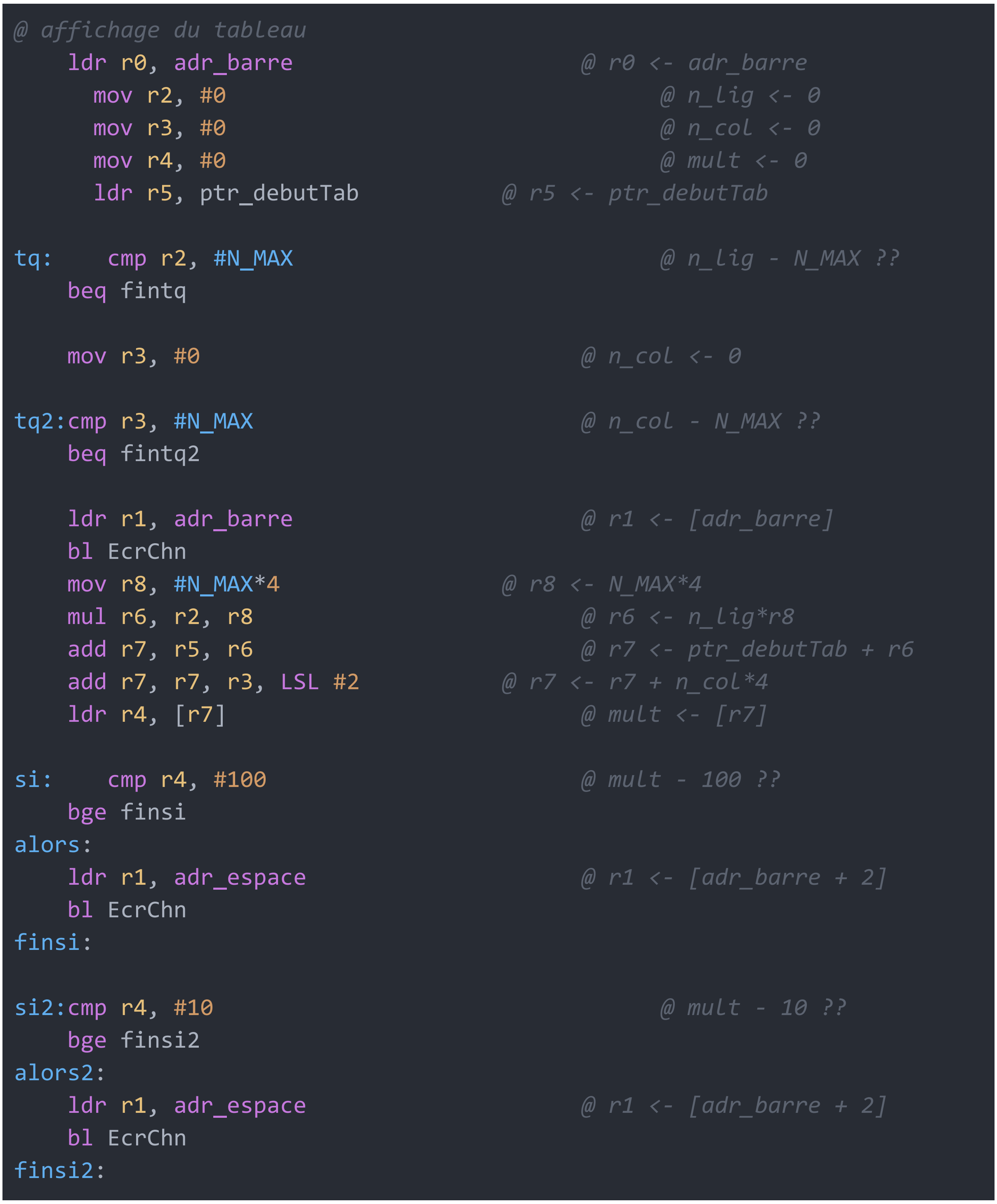
Compte-Rendu TP6/7

04/03/24
MIN-1

6.1

Les case du tableau sont comprises comme d'habitudes comme Tab[ligne - 1][colonne - 1].
Donc le produit 1*1 se trouve à Tab[0][0]
1*2 se trouve à Tab[0][1]
7*9 se trouve à Tab[6][8]
10*10 se trouve à Tab[9][9]

6.2



```
@ affichage du tableau
    ldr r0, adr_barre                      @ r0 <- adr_barre
      mov r2, #0                                 @ n_lig <- 0
      mov r3, #0                                 @ n_col <- 0
      mov r4, #0                                 @ mult <- 0
      ldr r5, ptr_debutTab           @ r5 <- ptr_debutTab

tq:    cmp r2, #N_MAX                            @ n_lig - N_MAX ??
    beq fintq

    mov r3, #0                             @ n_col <- 0

tq2:cmp r3, #N_MAX                         @ n_col - N_MAX ??
    beq fintq2

    ldr r1, adr_barre                      @ r1 <- [adr_barre]
    bl EcrChn
    mov r8, #N_MAX*4                 @ r8 <- N_MAX*4
    mul r6, r2, r8                         @ r6 <- n_lig*r8
    add r7, r5, r6                         @ r7 <- ptr_debutTab + r6
    add r7, r7, r3, LSL #2           @ r7 <- r7 + n_col*4
    ldr r4, [r7]                           @ mult <- [r7]

si:    cmp r4, #100                        @ mult - 100 ??
    bge finsi
alors:
    ldr r1, adr_espace                     @ r1 <- [adr_barre + 2]
    bl EcrChn
finsi:

si2:cmp r4, #10                                  @ mult - 10 ??
    bge finsi2
alors2:
    ldr r1, adr_espace                     @ r1 <- [adr_barre + 2]
    bl EcrChn
finsi2:
    mov r1, r4
    bl EcrNdecim32

    add r3, r3, #1
    b tq2
fintq2:

    ldr r1, adr_barre                     @ r1 <- [adr_barre]
    bl EcrChn
    bl AlaLigne

    mov r6, #0
tq3:cmp r6, #N_MAX                        @ r6 - N_MAX ??
    beq fintq3

    ldr r1, adr_barre                     @ r1 <- [adr_barre]
    bl EcrChn
    ldr r1, adr_tirets                    @ r1 <- [adr_barre + 4]
    bl EcrChn

    add r6, r6, #1
    b tq3
fintq3:

    ldr r1, adr_barre                     @ r1 <- [adr_barre]
    bl EcrChn
    bl AlaLigne

      add r2, r2, #1
      b tq
fintq:
```

6.3

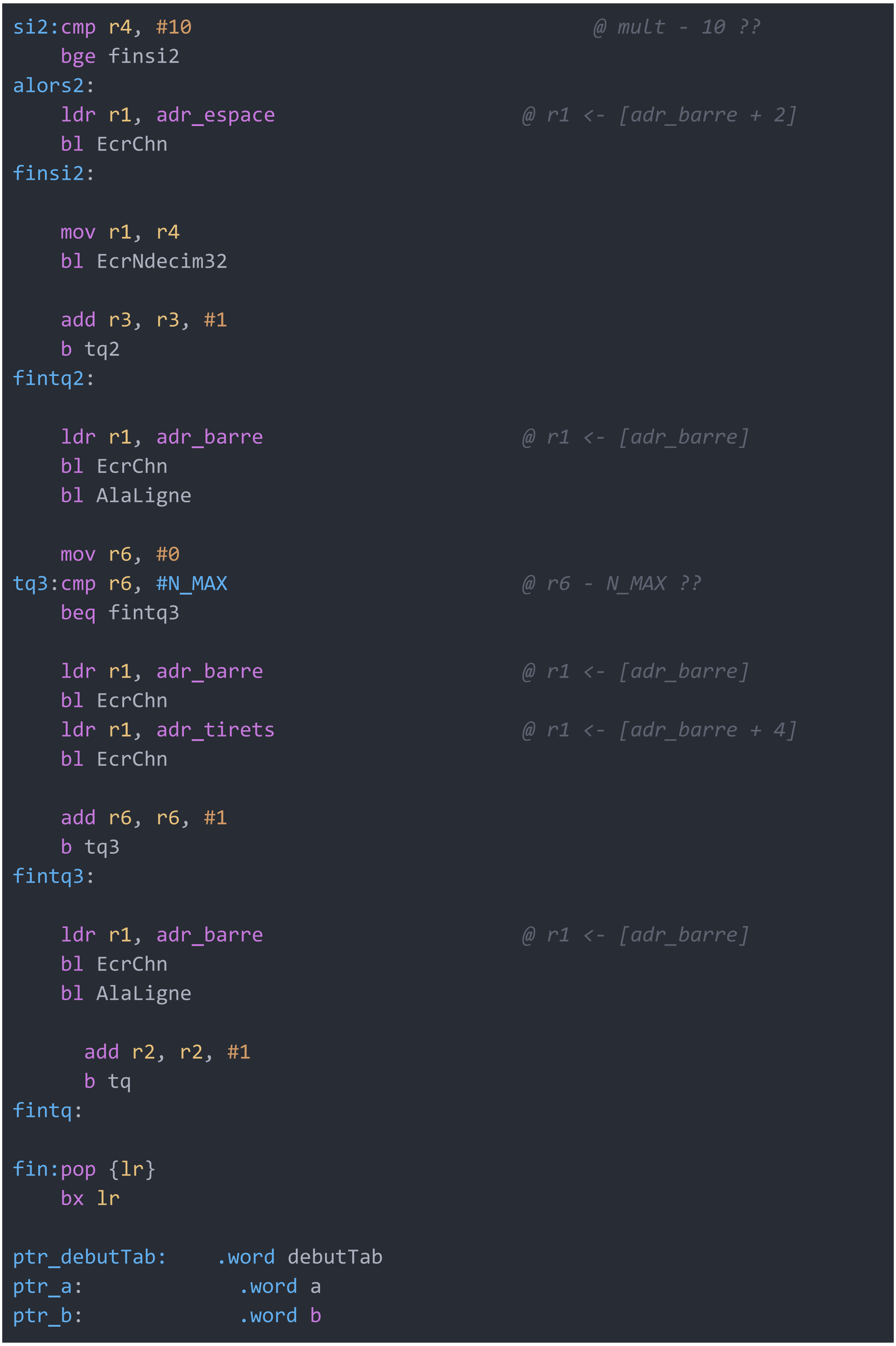
1) table + x * N_MAX * 4 + y * 4

2)

```
mov r8, #N_MAX*4                    @ r8 <- N_MAX*4
mul r6, r2, r8                      @ r6 <- n_lig*r8
add r7, r5, r6                      @ r7 <- ptr_debutTab + r6
add r7, r7, r3, LSL #2              @ r7 <- r7 + n_col*4
str [r7], r4                        @[r7] <-mult
```

6.3.2

```
@ Programme tabmult : Affiche les tables de multiplication de de 1 a 10
N_MAX = 10
    .data
barre:    .byte '|'
       .byte 0
espace:    .byte ' '
       .byte 0
tirets:    .asciz "---"
a:     .word 0
b:     .word 0


debutTab:    .skip N_MAX*N_MAX*4     @ adresse du debut du tableau

    .text
    .global main
main: push {lr}

@#######################################################################
########
      @ remplissage du tableau
      mov r2, #0                                 @ n_lig <- 0
      mov r3, #0                                 @ n_col <- 0
      mov r4, #0                                 @ mult <- 0
      ldr r5, ptr_debutTab           @ r5 <- ptr_debutTab

etq:cmp r2, #N_MAX                         @ n_lig - N_MAX ??
    beq efintq

    mov r3, #0                             @ n_col <- 0

etq2:
    cmp r3, #N_MAX                         @ n_col - N_MAX ??
    beq efintq2

                                                 @ r7 <- n_lig*N_MAX*4 +
n_col*4
    mov r8, #N_MAX*4                 @ r8 <- N_MAX*4
    mul r6, r2, r8                         @ r6 <- n_lig*r8
    add r7, r5, r6                         @ r7 <- ptr_debutTab + r6
    add r7, r7, r3, LSL #2           @ r7 <- r7 + n_col*4

    add r6, r2, #1                         @ r6 <- n_lig + 1
    add r8, r3, #1                         @ r8 <- n_col + 1
    mul r4, r6, r8                         @ mult <- r6 * r8

    str r4, [r7]                           @ [r7] <- mult

    add r3, r3, #1                         @ n_col++
    b etq2
efintq2:

      add r2, r2, #1                       @ n_lig++
      b etq
efintq:

@#######################################################################
########

      @ affichage du tableau
    ldr r0, adr_barre                      @ r0 <- adr_barre
      mov r2, #0                                 @ n_lig <- 0
      mov r3, #0                                 @ n_col <- 0
      mov r4, #0                                 @ mult <- 0
      ldr r5, ptr_debutTab           @ r5 <- ptr_debutTab

tq:    cmp r2, #N_MAX                            @ n_lig - N_MAX ??
    beq fintq

    mov r3, #0                             @ n_col <- 0

tq2:cmp r3, #N_MAX                         @ n_col - N_MAX ??
    beq fintq2

    ldr r1, adr_barre                      @ r1 <- [adr_barre]
    bl EcrChn
    mov r8, #N_MAX*4                 @ r8 <- N_MAX*4
    mul r6, r2, r8                         @ r6 <- n_lig*r8
    add r7, r5, r6                         @ r7 <- ptr_debutTab + r6
    add r7, r7, r3, LSL #2           @ r7 <- r7 + n_col*4
    ldr r4, [r7]                           @ mult <- [r7]

si:    cmp r4, #100                        @ mult - 100 ??
    bge finsi
alors:
    ldr r1, adr_espace                     @ r1 <- [adr_barre + 2]
    bl EcrChn
finsi:
```

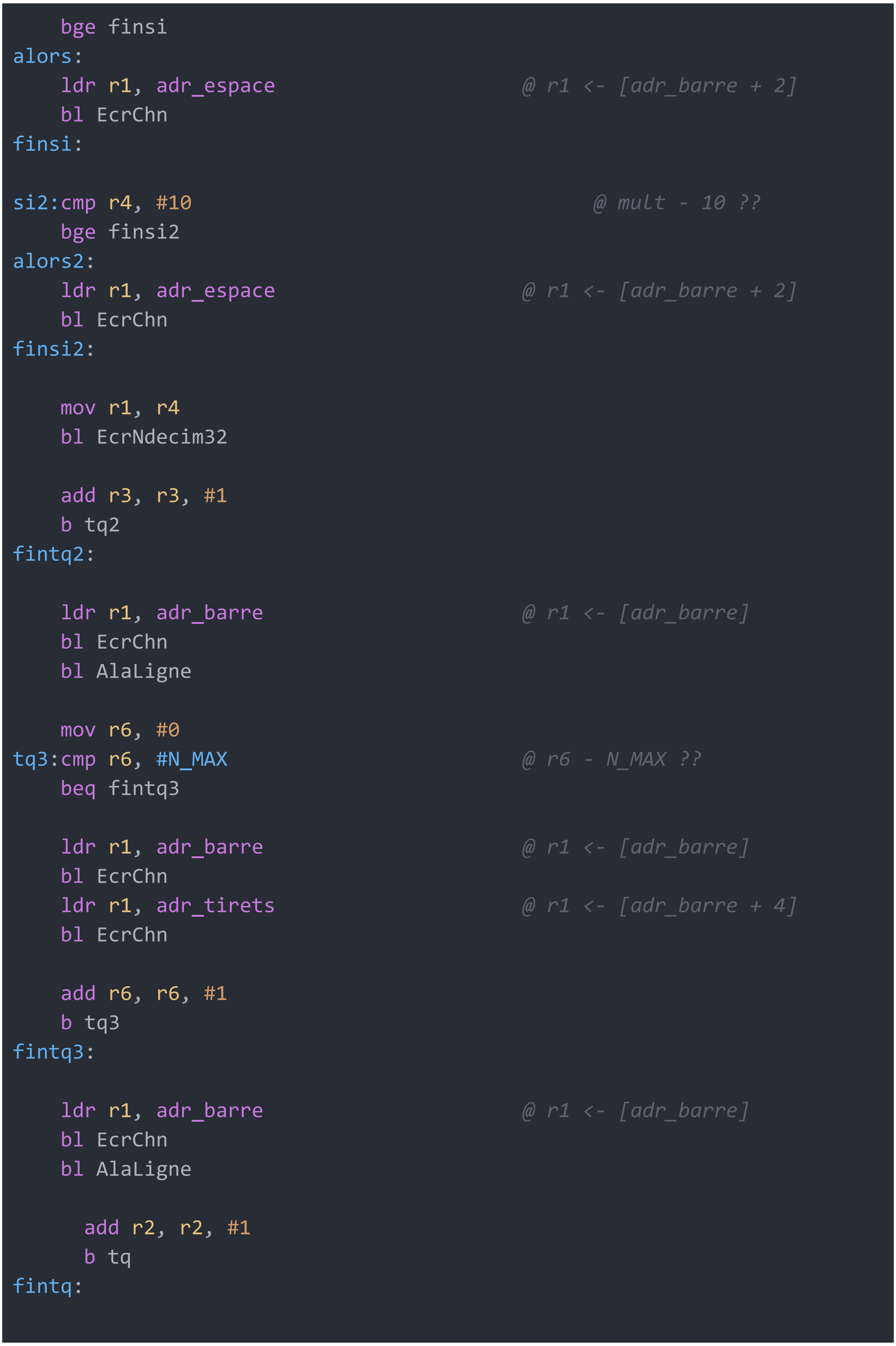


```
si2:cmp r4, #10                                  @ mult - 10 ??
    bge finsi2
alors2:
    ldr r1, adr_espace                     @ r1 <- [adr_barre + 2]
    bl EcrChn
finsi2:

    mov r1, r4
    bl EcrNdecim32

    add r3, r3, #1
    b tq2
fintq2:

    ldr r1, adr_barre                      @ r1 <- [adr_barre]
    bl EcrChn
    bl AlaLigne

    mov r6, #0
tq3:cmp r6, #N_MAX                         @ r6 - N_MAX ??
    beq fintq3

    ldr r1, adr_barre                      @ r1 <- [adr_barre]
    bl EcrChn
    ldr r1, adr_tirets                     @ r1 <- [adr_barre + 4]
    bl EcrChn

    add r6, r6, #1
    b tq3
fintq3:

    ldr r1, adr_barre                      @ r1 <- [adr_barre]
    bl EcrChn
    bl AlaLigne

      add r2, r2, #1
      b tq
fintq:

fin:pop {lr}
    bx lr

ptr_debutTab:    .word debutTab
ptr_a:             .word a
ptr_b:             .word b
```

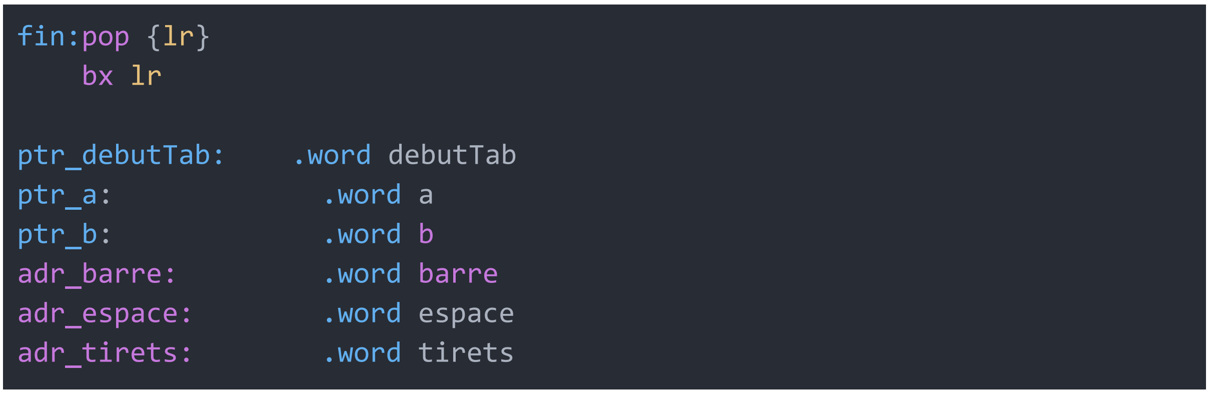
```
adr_barre:          .word barre
adr_espace:         .word espace
adr_tirets:         .word tirets
```

6.3.3

```
@ Programme tabmult : Affiche les tables de multiplication de 1 a 10
N_MAX = 10
    .data
barre:    .byte '|'
       .byte 0
espace:    .byte ' '
       .byte 0
tirets:    .asciz "---"
a:     .word 0
b:     .word 0


debutTab:    .skip N_MAX*N_MAX*4    @ adresse du debut du tableau

    .text
    .global main
main: push {lr}

@########################################################################
########
      @ remplissage du tableau
      mov r0, #0                                @ n <- 0
      mov r2, #0                                @ n_lig <- 0
      mov r3, #0                                @ n_col <- 0
      mov r4, #0                                @ mult <- 0
      ldr r5, ptr_debutTab          @ r5 <- ptr_debutTab

etq:cmp r0, #N_MAX*N_MAX            @ n - N_MAX*N_MAX ??
    beq efintq

                                                @ r7 <- n_lig*N_MAX*4 +
n_col*4
    mov r8, #N_MAX*4                @ r8 <- N_MAX*4
    mul r6, r2, r8                        @ r6 <- n_lig*r8
    add r7, r5, r6                        @ r7 <- ptr_debutTab + r6
    add r7, r7, r3, LSL #2          @ r7 <- r7 + n_col*4

    add r6, r2, #1                        @ r6 <- n_lig + 1
    add r8, r3, #1                         @ r8 <- n_col + 1
    mul r4, r6, r8                         @ mult <- r6 * r8

    str r4, [r7]                           @ [r7] <- mult

esi:cmp r3, #9                             @ n_col - 9 ??
    bne esinon
ealors:
    mov r3, #0                             @ n_col <- 0
    add r2, r2, #1                         @ n_lig++
    b efinsi
esinon:
    add r3, r3, #1                         @ n_col++
efinsi:
      add r0, r0, #1                       @ n++
      b etq
efintq:

@#########################################################################
########

      @ affichage du tableau
    ldr r0, adr_barre                      @ r0 <- adr_barre
      mov r2, #0                                 @ n_lig <- 0
      mov r3, #0                                 @ n_col <- 0
      mov r4, #0                                 @ mult <- 0
      ldr r5, ptr_debutTab           @ r5 <- ptr_debutTab

tq:    cmp r2, #N_MAX                            @ n_lig - N_MAX ??
    beq fintq

    mov r3, #0                             @ n_col <- 0

tq2:cmp r3, #N_MAX                         @ n_col - N_MAX ??
    beq fintq2

    ldr r1, adr_barre                      @ r1 <- [adr_barre]
    bl EcrChn
    mov r8, #N_MAX*4                 @ r8 <- N_MAX*4
    mul r6, r2, r8                         @ r6 <- n_lig*r8
    add r7, r5, r6                         @ r7 <- ptr_debutTab + r6
    add r7, r7, r3, LSL #2           @ r7 <- r7 + n_col*4
    ldr r4, [r7]                           @ mult <- [r7]

si:    cmp r4, #100                        @ mult - 100 ??
```



```
    bge finsi
alors:
    ldr r1, adr_espace                     @ r1 <- [adr_barre + 2]
    bl EcrChn
finsi:

si2:cmp r4, #10                                  @ mult - 10 ??
    bge finsi2
alors2:
    ldr r1, adr_espace                     @ r1 <- [adr_barre + 2]
    bl EcrChn
finsi2:

    mov r1, r4
    bl EcrNdecim32

    add r3, r3, #1
    b tq2
fintq2:

    ldr r1, adr_barre                      @ r1 <- [adr_barre]
    bl EcrChn
    bl AlaLigne

    mov r6, #0
tq3:cmp r6, #N_MAX                         @ r6 - N_MAX ??
    beq fintq3

    ldr r1, adr_barre                      @ r1 <- [adr_barre]
    bl EcrChn
    ldr r1, adr_tirets                     @ r1 <- [adr_barre + 4]
    bl EcrChn

    add r6, r6, #1
    b tq3
fintq3:

    ldr r1, adr_barre                      @ r1 <- [adr_barre]
    bl EcrChn
    bl AlaLigne

      add r2, r2, #1
      b tq
fintq:
```



```
fin:pop {lr}
    bx lr

ptr_debutTab:    .word debutTab
ptr_a:             .word a
ptr_b:             .word b
adr_barre:         .word barre
adr_espace:        .word espace
adr_tirets:        .word tirets
```