

TD 3

TAD Semestre 2

1

```
1  VRAI    .equ 1
2  FAUX    .equ 0
3  TAILLE  .equ 1000
4          .bss
5          .align 2
6  tab     .space 4 * TAILLE
7  n       .space 4
8  else    .space 4
9  rang    .space 4
10 trouve  .space 1
11         .text
12
13 .pTab    .word  tab
14 .pN      .word  n
15 .pEle    .word  ele
16 .pRang   .word  rang
17 .pTrouve .word  trouve
```

Listing 1 – Glossaire assembleur

```
1  glossaire
2  R0 ≡ i
3  R1: .pTab ou .ptrouve ou .pRang
4  R2: .pN puis n
5  R3: .pEle puis ele
6  R4: tab[i]
7  debut
8  R0 <- #0;
9  tantque (R2 <- .pN, R2 <- R2↑, R0 < R2) et
10      (R1 <- .pTab, R4 <- (R1+DLG(2,R0)), R3 <- .pEle,
11      R3 <-R3↑, R4 /= R3) faire
12      R0 <- R0 + #1;
13  fin tantque;
14  si R0 < R2 alors
15      R1 <- .pRang;
16      R1 ↑ <- R0;
17      R1 <- .pTrouve;
18      R1↑.B <- #VRAI;
19  sinon
20      R1 <- .pTrouve;
21  R1
```

Listing 2 – Algorithme

Listing 3 – Programme assembleur

2

```
1  rechercherOccurence:
2      MOV R0, #0
3      LDR R1, .ptab
4      LDR R1, .ptab
5      LDR R2, .pEle
6      LDR R2, [R2]
7      LDR R3, .pn
8      LDR R3, [R3]
9  tantque    CMP R0, R3
10     BHS fintantque
11     LDR R4, [R1,R0,LSL#2]
12     CMP R4, R2
13     BEQ fintantque
14     ADD R0,r0,#1
15     B tantque
16 fintantque  CMP R0, R2
17     LDR R1, .ptrouve
18     MOV LO R3, #VRAI
19     MOV HS R3, #FAUX
20     STRB R3, [R1]
21     LDRLO R1, .pRang
22     STRLO R0, [R1]
23 fin
```

Listing 4 – Programme assembleur