

**Exercice 1 :**

```
1     LDR R1, val1
2     LDR R2, val2
3     STR R1, val2
4     STR R2, val1
5 val1: 10
6 val2: 12
```

**Exercice 2 :**

```
1     LDR R0, x
2     LDR R1, y
3     LDR R2, z
4     ADD R1,R1,#42
5     CMP R0,R1
6     BEQ egal
7     MOV R2,#2
8     B fin
9 x:0
10 y:0
11 z:0
12 egal:
13     MOV R2,#1
14 fin:
15     STR R2,z
16     HALT
```

**Exercice 3 :**

```
1     MOV R0, #tab //charge l'adresse du début du tableau
2     LDR R1, longueur //charge la longueur du tableau
3     MOV R2, #0 //prépare un compteur
4     MOV R4,#-1 //stocke le max
5 boucle:
6     LDR R3, [R0] //charge la valeur à l'adresse [R0] de la mémoire
7     CMP R3,R4 //compare la valeur courante avec R4
8     BGT maxi
9 finboucle:
10    ADD R0, R0, #4 //passe à l'adresse suivante
11    ADD R2, R2, #1 //augmente le compteur de 1
12    CMP R2, R1 // vérifie si on est en fin de tableau
13    BLT boucle
14    STR R4, .WriteUnsignedNum //affiche la valeur
15    HALT
16 maxi: // stocke la valeur max
17    MOV R4,R3
18    B finboucle
19 longueur:10
20 tab: 13
21     9
22     10
23     12
```

```
24     51
25     8
26     71
27     19
28     3
29     11
```

**Exercice 4 :** Le programme fait la somme des entiers de 1 à 10.

```
1 //somme entiers en x86
2     MOV EAX, 0
3     MOV ECX, 10
4 ici:
5     CPM ECX, 0
6     JE la
7     ADD EAX, ECX
8     SUB ECX, 1
9     JMP ici
10 la:
```