

Exercice 1 : Que fait le programme 1 ?

```

1      MOV R0,#0
2      MOV R1,#100
3  ici:
4      CMP R1,#0
5      BEQ la
6      ADD R0,R0,R1
7      SUB R1,R1,#1
8      B  ici
9  la:
10     HALT

```

Code 1 – Que fait ce programme?

Exercice 2 : Traduire les instructions suivantes en assembleur :

```

1  x=y+42
2  if x==42:
3      z=1
4  else:
5      z=2

```

Exercice 3 : Les microprocesseurs Intel utilisent le langage d'assemblage x86. Ce langage a huit registres : eax, ebx, ecx, edx, esi, edi, esp et ebp. La documentation (très incomplète) ci-dessous présente plusieurs instructions de ce langage :

Instructions	Utilisation	Exemple
mov destination, source	copie la source dans la destination	mov eax, 42
add destination, source	ajoute la source à la destination	add eax, 10
sub destination, source	soustraie la source à la destination	sub eax, 10
cmp source1, source2	effectue la soustraction source1-source2 et indique le résultat de comparaison dans des registres spéciaux (flag). Si le flag <i>zf</i> vaut 1 c'est que <i>source1</i> est égal à <i>source2</i> . Si le flag <i>sf</i> vaut 1 c'est que <i>source1</i> est inférieure à <i>source2</i> .	cmp eax, ebx
je adr	saute à l'adresse si le flag <i>zf</i> vaut 1. L'adresse est une étiquette ou label.	je label
jne adr	saute à l'adresse si <i>zf</i> vaut 0. L'adresse est une étiquette ou label.	jne label
jmp adr	saute à l'adresse. L'adresse est une étiquette ou label.	jmp label

Traduire le code 1 dans le langage assembleur x86.