

TP : ASSEMBLEUR

I) Comprendre ce que font des programmes simples

Affichez simultanément le "[simulateur AQA](#)" ainsi que l'aide mémoire en français proposé sur mon site.

Pour chacun des programmes ci-dessous :

- 1) Chercher dans l'aide mémoire ce que fait chaque instruction du programme
- 2) Copier-coller et exécuter au pas à pas le programme dans le simulateur en essayant de bien comprendre ce que vous voyez
- 3) Trouver ce que fait le programme

PROGRAMME 1

```
inp R0,2
mvn R0,R0
add R0,R0,#1
out R0,4
halt
```

PROGRAMME 2

```
inp R0,2
lsl R0,R0,#3
out R0,4
halt
```

PROGRAMME 3

```
inp R0,2
inp R1,2
mov R2,#0
boucle:
  cmp R1,#0
  beq fin
  sub R1,R1,#1
  add R2,R2,R0
  b boucle
fin:
  out R2,5
  halt
```

PROGRAMME 4

```
inp R0,2
and R1,R0,#1
cmp R1,#1
bne pair
mov R2,#0x69
out r2,7
mov R2,#0x6D
out r2,7
pair:
  mov R2,#0x70
  out r2,7
  mov R2,#0x61
  out r2,7
  mov R2,#0x69
  out r2,7
  mov R2,#0x72
  out r2,7
  halt
```

II) Comprendre comment est codée une instruction

- 1) Afficher le contenu de la mémoire en binaire (En bas : Options, puis binary)
- 2) Dans la zone "Assembly Language", taper l'instruction "mov R0,R1" et effacer tout le reste. Noter par quelle suite de 0 et de 1 cette instruction est codée en mémoire.
- 3) Garder l'instruction mov et modifier les registres indiqués en paramètre.
En déduire comment est codé l'instruction mov, comment est codé le premier registre passé en paramètre, comment est codé le deuxième registre passé en paramètre.
- 4) En déduire quelle est en assembleur l'instruction : 11100001 10100000 11000000 00000111
- 5) Taper maintenant : "add R0,R1,R2". Chercher de même comment cette nouvelle instruction est codée.
- 6) Essayer maintenant : "add R0,R1,#2". Voyez-vous une différence avec "add R0,R1,R2" ?
- 7) En déduire quelle est en assembleur l'instruction : 11100010 10000111 01110000 00000001

III) Comprendre mieux le fonctionnement des drapeaux

- 1) Dans ce simulateur, quelle est l'instruction qui modifie les drapeaux NZCV dans la zone Status ?
- 2) Écrire un programme qui met le flag N à 1, puis à 0
- 3) Écrire un programme qui met le flag Z à 1, puis à 0

IV) Écrire des programmes simples

- 1) Écrire un programme qui demande 2 entiers positifs et affiche le plus grand
- 2) Écrire un programme qui demande 3 entiers positifs et affiche le plus grand
- 3) Écrire un programme qui demande 1 entier relatif et affiche sa valeur absolue
- 4) Écrire un programme qui demande 1 entier positif n et affiche 2^n
- 5) Écrire un programme qui demande 1 entier positif n et affiche la somme des entiers de 0 à n compris
- 6) Écrire un programme qui demande 1 entier positif n et affiche son carré
- 7) Écrire un programme qui demande 1 entier positif n et affiche le nombre de chiffres de l'écriture binaire de n
- 8) Écrire un programme qui demande 3 entiers positifs et les affiche par ordre croissant
- 9) Écrire un programme qui fait une division euclidienne de 2 entiers positifs et affiche le quotient et le reste