Contrôle NSI

Exercice 1 :

Que fait le mini programme suivant :

```
MOV   R0,#42  :
STR   R0,150  :
HALT          :
```

Exercice 2

Donner les lignes de commandes en assembleur pour effectuer les actions suivantes :

1) Place la Valeur stockée à l'adresse mémoire 78 dans le register R1 (par souci de simplification, comme durant le cours, nous utiliserons des adresses codées en base 10)
2) Soustraire le nombre 128 de la Valeur stockée dans le register R0, place le resultat dans le register R1
3) Place la Valeur stockée dans le register R3 en mémoire vive à l'adresse 125
4) Place le nombre 23 dans le register R1.
5) Compare R0 et le nombre 17, et si R0 est plus grand que 17, on va à l'adresse 50 sinon on va à l'adresse 120.

Exercice 3

Considérons le programme suivant :

```
1     MOV R0,#2
2     MOV R1,#1
3     B 25
…
25    ADD R0,R0,R0
26    ADD R1,R1,#1
27    CMP R1,#5
28    BNE 25
29    STR R0,100
30    HALT
```

1) Détailler ligne par ligne ce qu'il fait , puis en déduire son utilité
2) Modifiez ce programme pour calculer 2^{10}

Exercice 4 : Listes définies par compréhension

1) Quelle est la valeur de l'expression [2*k + 1 for k in range(4)] ?
2) On exécute le script suivant :

```
L = [12,0,8,7,3,1,5,3,8]
arkana = [elt for elt in L if elt<4]
```

   Quelle est la valeur de akana à la fin de son exécution ?
3) Comment définir par compréhension la liste des cubes des entiers compris entre 10 et 20 ?
4) Soit array2 une variable contenant une liste d'entiers positifs. Comment définir par compréhension array2uneven la liste contenant tous les éléments de array2 qui sont impairs, autrement dit dont le reste par la division euclidienne par 2 est 1.
5) On définit :

```
stock = [ {'nom': 'flageolets', 'quantité': 50, 'prix': 5.68},
          {'nom': 'caviar', 'quantité': 0, 'prix': 99.99},
          {'nom': 'biscuits', 'quantité': 100, 'prix': 7.71} ]
```

   Quelle expression permet d'obtenir la liste des noms des produits effectivement présents dans le stock (c'est-à-dire ceux dont la quantité n'est pas nulle) ?

   A ['nom' for p in stock if 'quantité' != 0]
   B [p for p in stock if p['quantité'] != 0]
   C [p['nom'] for p in stock if 'quantité' != 0]
   D [p['nom'] for p in stock if p['quantité'] != 0]

Exercice 5 (bonus ???)

Considérons le programme suivant :

```
        MOV R0, #4
        STR R0,30
        MOV R0, #8
        STR R0, 75
        LDR R0, 30
        CMP R0, #10
        BNE label1
        MOV R0, #9
        STR R0, 75
        B label2
label1 :
        LDR R0, 30
        ADD R0, R0, #1
        STR R0, 30
label2 :
        MOV R0, #6
        STR R0, 23
        HALT
```

1) Que fait ce programme ?
2) En écrire une version en python (les adresses mémoires 30, 75 et 23 correspondrons à trois variables que nous appellerons respectivement x, y et z.

Correction

Exercice 1

```
MOV R0,#42  : Place le nombre 42 dans le registre R0
STR R0,150  : Stocke le contenu de R0 dans la mémoire 150
HALT        : on arrête le processus
```

Exercice 2

1) LDR R1, 78
2) SUB R1,R0,#128
3) STR R3, 125
4) MOV R1, #23
5) CMP R0, #17 BGT 50 B 120

Exercice 3

```
MOV R0,#2          mettre la valeur 2 dans le registre R0
MOV R1,#1          mettre la valeur 1 dans le registre R1
B 25               aller à l'emplacement 25

ADD R0,R0,R0       mettre le double de R0 dans R0
ADD R1,R1,#1       ajouter 1 à R1
CMP R1,#5          comparer R1 et 5
BNE 25             s'ils ne sont pas égaux aller à l'emplacement 25
STR R0,100         stocker R0 dans l'emplacement mémoire 100
HALT               fin du programme
```

1) Ce programme stocke la Valeur de 2^5 à l'emplacement mémoire 100
2) Il suffit de remplacer #5 par #10 à la ligne 27

Exercice 4 : Listes définies par compréhension

1) [1,3,5,7]
2) arkana contiendra la liste [0,3,1,3]
3) [n**3 for n in range(10,21)]
4) array2uneven=[valeur in array2 if valeur%2==1]
5) D [p['nom'] for p in stock if p['quantité'] != 0]

Exercice 5 (bonus ???)

```
        MOV R0, #4          placer le nombre 4 dans le registre R0
        STR R0,30           placer la valeur du registre R0 dans la mémoire 30
        MOV R0, #8          placer le nombre 8 dans le registre R0
        STR R0, 75          placer la valeur du registre R0 dans la mémoire 75
        LDR R0, 30          charger le contenu de la mémoire 30 dans R0
        CMP R0, #10         comparer 10 et le contenu de R0
        BNE else            s'ils ne sont pas égaux aller à l'emplacement label1
        MOV R0, #9          placer la valeur 9 dans le registre R0
        STR R0, 75          stocker la valeur de R0 dans l'emplacement 75
        B endif             aller à l'emplacement label2
label1 :                    emplacement label1
        LDR R0, 30          charger le contenu de la mémoire 30 dans R0
        ADD R0, R0, #1      augmenter R0 de 1
        STR R0, 30          stocker le contenu de R0 dans la mémoire 30
label2 :                    emplacement label2
        MOV R0, #6          mettre la valeur 6 dans le registre R0
        STR R0, 23          stocker la valeur de R0 dans la mémoire 23
        HALT                arrêt du programme
```

En python ça donne :

```
x=4
y=8
if 10 !=x :
        x=x+1
else :
        y=9
z=6
```

Simulateur d'assembleur
Cours plutôt bien fait sur cette partie : https://www.math93.com/lycee/146-nsi/986-nsi-numerique-et-sciences-informatiques-memoire-et-langage-machine.html

http://www.peterhigginson.co.uk/AQA/

des devoirs de NSI
https://qkzk.xyz/docs/nsi/cours_premiere/devoirs_premiere/