

Enseignant : C.A. ABID	Année universitaire : 2025-2026
TD 1	
Module : Architectures avancées des processeurs (programmation assembleur ARM) - Section ICE3	

Exercice 1.

Insérer le nombre minimum de directives `align` dans le programme ci-après afin d'avoir un alignement correct.

```
.data
.align 2
a: .byte 0
b: .word 32
c: .byte 3
d: .hword 45
e: .hword 0
f: .byte 0
g: .word 128
```

- Réordonner les directives dans le programme précédent afin d'avoir un alignement correct sans utiliser la directive `.align`. Quel est le nombre d'octets gaspillés dans la première version ?

Exercice 2.

Écrire un programme en assembleur qui permute le contenu de deux variables `x` et `y` de type mot.

Exercice 3.

Écrire un programme en assembleur qui inverse l'ordre des octets d'un mot.

Exercice 4.

a) Donner une traduction possible en assembleur ARM de l'expression :

```
if (a==b) c=a else c=b*8 ;
```

b) Écrire un un programme assembleur qui renvoie dans le registre `r3` - le contenu de `r0/8` si sa valeur est divisible par 8. - la valeur 0 sinon.

c) Écrire un algorithme, puis un programme assembleur qui renvoie le maximum de trois valeurs contenues dans `r0`, `r1` et `r2`, dans le registre `r3`.

d) Il est possible d'écrire plus efficacement la structure conditionnelle de l'exemple (`if (a>b) c=a else c=b*8;`) sans utiliser de branchements, en utilisant la forme conditionnelle des instructions. Donner le code assembleur correspondant et comparer les deux versions.

Exercice 5.

Écrire un programme assembleur qui calcule la somme $1+2+\dots+N$ et range le résultat dans la variable `resultat`. On donne :

```
.data
N .byte 10
.bss
```

`resultat .word`

Exercice 6.

On se propose de trier trois valeurs signées par ordre croissant. L'utilisateur choisit trois nombres au hasard et le programme les triera dans l'ordre croissant, en mettant la plus petite dans le registre `R0` et la plus grande dans le registre `R2`.

Exercice 7.

On se propose de calculer le poids de Hamming d'une valeur binaire, c'est-à-dire le nombre de 1 que la valeur contient. À titre d'exemple, le nombre de Hamming de la valeur 00101011 est 4.

Pour effectuer ce calcul, nous proposons d'examiner la valeur du bit ayant le poids fort, nous incrémentons un compteur représentant le poids de Hamming si la valeur du bit est égale à un 1, puis nous décalons chaque bit du nombre vers la gauche. L'opération sera répétée jusqu'à ce que le nombre soit égal à 0.

Exercice 8.

Écrire un programme permettant de calculer le quotient et le reste d'une division entière non signée par soustraction successive.

Exercice 9.

Écrire un programme permettant de ranger les nombres pairs d'un tableau donné dans un deuxième tableau.