

R106 – Architecture des ordinateurs

Assembleur sur

TD 4

Exercice 1 Branchement conditionnel

Que fait le code suivant ?
Expliquez-le en détail.

```
SECTION INTVEC

B main

SECTION CODE
mon_tableau ASSIGN32 0xCA330B0, 0xE7FA1998, 0x93C2C5DE, 0x9DC73798

main
LDR R0, =mon_tableau
LDR R3, [R0]
MOV R1, #0x04
boucle
SUBS R1, R1, #0x01
BEQ fin
ADD R0, R0, #0x04
LDR R2, [R0]
CMP R2, R3
MOVGT R3, R2
B boucle

fin
B fin

SECTION DATA
```

Exercice 2 Utilisation du registre LR

Développez la fonction Puissance qui calcule R0 à la puissance R1 et qui stocke le résultat dans R0.

Exercice 3

Voici un exemple, en assembleur, de fonction permettant de calculer le triple d'une valeur transmise par R0.

Observez comment la pile est utilisée afin de préserver les registres R1 et LR :

```

SECTION INTVEC

B main

SECTION CODE

main

LDR SP, =maPile ; Préparons une pile (de 16 octets)
ADD SP, SP, #16 ; La pile descend, donc il faut commencer à la fin

B debut
NOP

debut
MOV R1, #2
PUSH {R0} ; Sauvegarde R0
MOV R0, #8
BL Triple ; Appel de fonction
MOV R2, R0
POP {R0} ; Restaure R0
CMP R1, R2

fin
B fin

; Espace vide pour que le code de la fonction débute à l'adresse 0xC0
; comme dans les notes.
tmp2 ALLOC32 5

Triple
; paramètre: R0
; valeur de retour: R0
PUSH {R1,LR} ; Sauvegarde R1 et LR

ADD R1, R0, R0 ; Calcule 3*R0 en utilisant des additions.
ADD R1, R1, R0

MOV R0, R1 ; Place résultat dans R0 (valeur de retour)
POP {R1,LR} ; Restaure R1 et LR
BX LR ; Fonction terminée

SECTION DATA

maPile ALLOC32 4

```

- ▷ Expliquez comment est allouée la pile.
A quelle adresse mémoire se situe-t-elle? Quelle est sa taille?
- ▷ Dans quelle sens Evolue la pile? Où est son début? Sa fin?
- ▷ Expliquez ce que font les instructions PUSH et POP.

Exercice 4

A quoi peut servir la pile? Donnez un exemple pratique. (Vous pourrez reprendre la question 2 par exemple)