

**EMD2 Mai 1999 Architecture des ordinateurs
Klouche-Djedid A.**

Soit le programme suivant :

PP			TOTO	
adresse	Instruction	signification	adresse	Instruction
@A0	CLEAR R0	Initialiser le R0 à la valeur 0	@B0	ADD R2(indirect), R0
@A1	MOVE N, R1		@B1	Incrémenter R2
@A2	MOVE NB1(immédiat), R2		@B2	Décrémenter R1
@A3	CALL TOTO		@B3	RTS
@A4	BRU R1,@A3	Branchement à @A3 si (R1)>0
@A5	MOVE R0,SOMME			
@A6	Halt			
.....			

Juste avant l'exécution de l'instruction @A0, nous avons (SP) = 2000, (R0) = 5000, (R1) = 5001, (R2) = 5002.

- 1) dérouler (phase de recherche et d'exécution) les instructions d'adresses suivantes : @A0, @A2, @A3, @A4, @B0, @B2.
- 2) déterminer le contenu de la pile, des registres SP, PC, R0, R1 et R2 juste après l'exécution des instructions d'@A3, @B0, @B3 et @A4.
- 3) dérouler le programme PP et le sous-programme TOTO instruction par instruction selon le tableau suivant :

PP		
Adresse	instruction	signification
@A0	CLEAR R0	Initialiser le R0 à la valeur 0
@A1	MOVE N, R1
@A2et ainsi de suite jusqu'à @A6
	TOTO	
@B0	ADD R2 (indirect),R0
.....
.....et ainsi de suite jusqu'à @B3

Que fait ce programme ?

4) réécrire le programme ci-dessus en utilisant le mode auto-incrémentation à R2 et le mode auto-décrémentation à R1.

5) réécrire un programme plus souple qui réalise la même tâche que celui cité ci-dessus, en utilisant le mode d'adressage indexé.

N.B.

Nom	Syntaxe	Signification
Registre	Rn	@eff= Rn, Op = (Rn)
Auto-incrémentation	(Rn)+	@eff = (Rn) et $Rn \leftarrow (Rn) + 1$
Auto-décrémentation	(Rn)-	$Rn \leftarrow (Rn)-1$ et @eff = (Rn)
Indexé	X(Rn)	@eff = X + (Rn) avec index = (Rn)
Pile	Empiler	Dans l'ordre croissant des adresses
	SP	Pointe sur la dernière case mémoire libre

PC (ou compteur ordinal CO) : program counter ; SP :Stack Pointeur (pointeur de sommet de pile ,libre ou empli selon le mode choisi)