

## Dpt Informatique-IGMO

### EMD2 Architecture des ordinateurs 2001

#### Klouche-Djedid A.

**Exo1** : les piles offrent un moyen d'accéder à des données en mémoire principale, qui est utilisée pour stocker temporairement des valeurs

Une pile est une zone de mémoire qui utilise un pointeur qui conserve l'adresse du sommet de la pile. En utilisant les instructions PUSH et PULL, écrire une séquence d'instructions qui permet de transférer le contenu du registre RO vers R1. utiliser seulement PUSH et PULL et éviter d'écraser le contenu de R1.

**Exo2** : Voici les caractéristiques du processeur 80x86

CPU :bus de données 16 bits et bus d'adresse 32 bits.

REGISTRES : ACC 16 bits BX et CX 16 bits ; pointeur d'instruction PC 16 bits ;pointeur de pile SP 16 bits

1	Adressage implicite (zéro adresse)	A	Placer dans BX la valeur contenue à @13
2	Adressage immédiat	B	Ajouter la valeur 5 à CX
3	Adressage direct	C	Incrémenter PC
4	Adressage relatif	D	Branchement à une @ par rapport au PC

Relier deux à deux chaque proposition 1, 2, 3, 4 avec la proposition A, B, C, D qui lui correspond et justifier votre réponse.

#### Exo3 :

Le tableau EMD2 présente quelques instructions du 80x86. le code instruction est donné en hexadécimal dans la deuxième colonne. La colonne qui suit précise le nombre d'octets nécessaires pour coder l'instruction complète (opérande inclus).

On note **valeur** une valeur sur 16 bits et **adr** une adresse sur 16 bits .

1. classer les instructions du tableau EMD2 (01 à 20) par format ; format à 0 adresse, format à 0.5 adresse, format à 1 adresse, à 2 adresses, etc.....
2. classer les instructions du tableau EMD2 (01 à 20) par modes d'adressage ; immédiat, direct, indirect, indexé, relatif.
3. décrire le déroulement des instructions 01, 02, 07, et 14 du tableau EMD2 (phase de recherche et d'exécution).
4. en utilisant les instructions du tableau EMD2, écrire un programme qui calcule la somme de deux emplacements mémoire **adr1** et **adr2** et stocke le résultat dans l'emplacement mémoire indiqué par le contenu de l' **adr2**. (simplifier le programme)
5. en utilisant les instructions du tableau EMD2, écrire un programme qui permet de trier par ordre décroissant les 10 éléments d'un vecteur de type entier. L'adresse du premier élément est **val0**. chaque mot est codé sur 2 octets.

**Tableau EMD2**

.....	Symbole	Code Op.	Octets	signification
01	MOV AX, valeur	B8	3	$AX \leftarrow \text{valeur}$
02	MOV AX,[adr]	A1	3	$AX \leftarrow \text{contenu de l'adresse adr}$
03	MOV [adr], AX	A3	3	Range AX à l'adresse adr.
04	ADD AX , valeur	05	3	$AX \leftarrow AX + \text{valeur}$
05	ADD AX, [adr]	03 06	4	$AX \leftarrow AX + \text{contenu de adr.}$
06	SUB AX, valeur	2D	3	$AX \leftarrow AX - \text{valeur}$
07	SUB AX,[adr]	2B 06	4	$AX \leftarrow AX - \text{contenu de adr}$
08	SHR AX, 1	D1 E8	2	Décale AX à droite
09	SHL AX, 1	D1 E0	2	Décale AX à gauche
10	INC AX	40	1	$AX \leftarrow +1$
11	DEC AX	48	1	$AX \leftarrow -1$
12	CMP AX,valeur	3D	3	Compare AX et valeur
13	CMP, AX, [adr]	3B 06	4	Compare AX et contenu de adr
14	JMP adr	EB	2	Saut inconditionnel(adrelatif)
15	JE adr	74	2	Saut si =
16	JNE adr	75	2	Saut si $\neq$
17	JG adr	7F	2	Saut si $>$
18	JLE adr	7E	2	Saut si $\leq$
19	JA adr			Saut si CF = 0
20	JB adr			Saut si CF = 1
<b>Fin du programme (retour au DOS)</b>				
	MOV AH,4C	B4 4C	2	
	INT 21	CD 21	2	