

Nom, prénom:	<b>Contrôle court n°2 - Durée 1 heure</b>
Grp:    A        B        C	<b>Calculatrice et documents interdits - répondre sur la feuille</b>

**EXERCICE 1. : PILE**

1.1. Quelles sont les instructions permettant d'accéder à la pile ? Précisez la taille des opérandes.

*Mettre sur la pile **PUSH** registre 16 bits*  
*Retirer de la pile **POP** registre 16 bits*

1.2. Quels éléments du processeur permettent de gérer la pile ?

*Registres **SS** (segment pile),*  
***SP** (stack pointer),*  
***BP** (base pointer)*  
*Tous 16 bits*

1.3. A partir de la taille de ces éléments, déterminez la taille maximum d'une pile.

*Registre **SP** 16 bits*  
*Pile de 2<sup>16</sup> octets (64ko)*

1.4. Donnez tous les éléments de déclaration et d'initialisations utiles pour une pile.

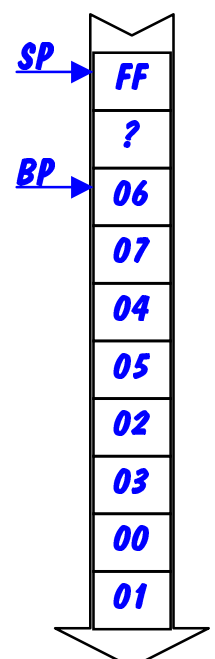
***ASSUME SS:Pile***  
***Pile SEGMENT STACK***  
***DW FFFFH DUP (?)**                   *# les 10000H octets, c'est avec 'bas'**  
**bas EQU THIS WORD**  
***Pile ENDS***  
***Dans le code segment :***  
***MOV AX,Pile***  
***MOV SS,AX***  
***MOV AX,bas***  
***MOV SP,AX***  
***MOV BP,SP***

1.5. Représentez l'état de la pile et des registres utiles à la fin des instructions suivantes :

```
MOV  BX,OFFSET tab      ;tab contient octets -1,-2,-3...
PUSH BX                  ;tab rangé à l'@ 1H
PUSH n                   ;N contient 203H
CALL fonction
```

La procédure fonction implantée en 0405H commence par les instructions :

```
PUSH BP                  ;à cet instant BP à 607H
MOV  BP,SP
SUB  SP,2
MOV  BX,[BP+6]
MOV  BYTE PTR [BP-2],[BX]
```



## EXERCICE 2. : PROCEDURE ET INTERRUPTIONS LOGICIELLES

2.1. Donnez les directives assembleur nécessaires à l'établissement d'une procédure `fonction`.

***fonction PROC NEAR***  
***fonction ENDP***

2.2. Complétez la procédure `fonction` qui change les éléments d'un tableau, dont l'adresse et le nombre sont passé en paramètre par la pile, en leur opposé (modifiez un minimum de registres).

```
        PUSH BP
        MOV  BP, SP
        SUB  SP, 2
        MOV  BX, [BP+6]
repet:  MOV  BYTE PTR [BP-2], [BX]
        CMP  [BP+4], 00H
        JEQ  retour
        XOR  [BP-2], 11111111b
        ADD  [BP-2], 1
        MOV  [BX], [BP-2]
        INC  [BP+4]
        JMP  repet
        ADD  SP, 2
        POP  BP
retour: RET
```

2.3. S'il s'agissait d'un traitant d'interruption...

a) Quelle serait la principale différence dans les instructions de la procédure ?

***IRET au lieu de RET***

b) Comment serait appelé ce traitant d'interruption ?

***INT XX au lieu de CALL***

c) Expliquez précisément les différences (expliquez le fonctionnement de l'appel).

***INT N va chercher l'@ du traitant en 4xN***  
***Il empile le registre d'état***  
***et CS:IP au lieu de IP seul puis appelle le traitant***  
***IRET dépile CS:IP et le registre d'état***

### EXERCICE 3. : INTERRUPTIONS MATERIELLES

3.1. Quelle est la différence fondamentale entre une interruption matérielle et une interruption logicielle...?

**Déclenché par un dispositif extérieur au processeur**

3.2. Quelles sont les bornes du processeur liées aux interruptions ? Précisez leur rôle.

**INT            interruption simple**

**INTA        ack**

**NMI        interruption non masquable**

3.3. Connexions

a) En connexion directe, combien de dispositifs peuvent faire des interruptions ?

**1**

b) Quelle est la solution adoptée ?

**PIC**

c) Représentez le branchement de deux interfaces travaillant par interruption (clavier et souris).

**INT PIC -> INT CPU, INTA CPU -> INTA PIC**

**INT interfaces -> IRQ PIC**

**@ et IO/M CPU -> @ décodeur**

**@ decodeur > @ et CS interfaces**

**data CPU <-> PIC <-> interfaces**

3.4. Donnez les étapes du traitement d'une interruption (par exemple appui d'une touche sur le clavier).

**Identification n° par le PIC, mise en attente**

**INT PIC -> CPU**

**CPU : vérif IF**

**si ok INTA CPU -> PIC**

**CPU sauve registre état et CS:IP**

**PIC n°IRQ sur data**

**Lit n°IRQ**

**(calcule correspondance INT)**

**CPU IF = 1**