

```

ORG      $1000
START:
    MOVE.L    #$12345678,D0
    MOVE.L    #$AAAAAAAA,D1
    MOVE.W    D0,D1
    SWAP      D0
    MOVE.W    D0,D2
    MOVE.L    #$12345678,D0
    BSR       AJOUTE
    MOVE.L    D0,D1
    BRA       FIN
AJOUTE:  MOVE.L    D0,-(A7)
    MOVE.W    #8,D0
    MOVE.W    #4,D1
    ADD.W     D0,D1
    MOVE.L    (A7)+,D0
    RTS
FIN:     END      START
    
```

Un appel à une sous-routine en lui passant des paramètres sur la Pile

```

ORG      $1000
START:  MOVE.W    #$1452,-(A7)
    BSR       ad1
    ADDA.L    #10,A7
    MOVE.W    #65,-(A7)
    MOVE.W    #2,-(A7)
    BRA       FIN
ad1:    ADDQ.L    #4,A7
    MOVE.W    #7,-(A7)
    RTS
FIN:
END      START
    
```

Application1:TP41 et TP42

1- Editer et assembler les programmes.

2-Interpretez

PEA - Push Effective Address.

PEA \$60(A4)

–adresse Effective est la somme de A4 et \$60.

– Résultat est placé dans la pile (pushed.)

Etudiez l'instruction MOVEM

PEA <EA> est équivalent à
 LEA <EA>,Ai
 MOVEA.L Ai,-(A7) a tester

TRAP et les exceptions du 68000

Une exception est un événement pouvant interrompre la séquence de traitement du processeur, comme une interruption, une opération illégale, un problème matériel, etc.

Les TRAPS sont un des types de vecteurs d'exceptions définis par le 68000. Les TRAPS peuvent être considérées comme les interruptions logicielles du 68000. Les vecteurs TRAPS sont au nombre de 16 (numéros de 32 à 47 dans la table des vecteurs Chaque vecteur est un mot long en mémoire (32 bits) et ils occupent les emplacements mémoires (adresses) 128₁₀ à 191₁₀.

Une TRAP s'écrit simplement: TRAP #n n c'est le n0 du < vecteur > Une fois l'environnement sauvegardé, le 68000 calcule l'adresse du vecteur TRAP demandé.

Operation: 1 → S-Bit of SR

SSP – 4 → SSP; PC → (SSP); SSP – 2 → SSP;

SR → (SSP); Vector Address → PC

```

ORG      $1000
CR       EQU     $0D
LF       EQU     $0A

START
LOOP:    LEA      MESSAGE,A1
    MOVE.B    #14,D0
    TRAP      #15
    ;le code pour faire des entrées
    MOVE.B    #4,D0
    TRAP      #15
    CMP.L     #0,D1
    BEQ       DONE
    BRA       LOOP
DONE:    MOVE.B    #9,D0
    TRAP      #15
MESSAGE DC.B    'Faites entrer les
données entrer 0 pour exit',0 ,cr
    
```