

# T.D. 11

## La pile et les sous-programmes

### Exercice 1

Soit le programme et les valeurs initiales ci-dessous :

```

001000 48E7C080    MOVEM.L D0/D1/A0,-(A7)

001004 7004        MOVEQ.L #4,D0
001006 7205        MOVEQ.L #5,D1
001008 41F84000    LEA      $4000,A0
00100C 6100000A    BSR      TAB
001010 2002        MOVE.L  D0,D2

001012 4CDF0103    MOVEM.L (A7)+,D0/D1/A0
001016 4E75        RTS

```

Valeurs initiales :     D0 = \$12345678    A0 = \$11112222

                             D1 = \$ABCDEF00    A7 = \$FFFFFFFC

                             \$FFFFFF8    02 24 32 AF 00 00 20 00

1. Quelles instructions du programme vont modifier la pile ?
2. Pour chacune d'entre elles, indiquez le contenu de la pile.
3. À quelle adresse se trouve la prochaine instruction à exécuter après le RTS ?

### Exercice 2

Soit la fonction ci-dessous rédigée en langage C :

```

long GetSol(short a, short b, short c)
{
    long delta;

    delta = b*b - 4*a*c;

    if (delta < 0)
        return 0;

    if (delta == 0)
        return 1;

    return 2;
}

```

1. Donnez l'équivalent en assembleur de l'instruction C suivante, sachant que les arguments sont passés à la fonction par la pile (le premier argument sera empilé en dernier) : `GetSol(5, 2, 1);`
2. La valeur de retour se trouvant dans le registre **D0**, donnez l'équivalent en assembleur de la fonction **GetSol()**. Vous commencerez par placer les arguments **a**, **b** et **c** dans les registres **D1**, **D2** et **D3**.