

## TP1

### Structure du programme source (squelette.asm)

```
Pile segment ;déclaration d'un segment de Pile
    dw 256 dup(0); ici on réserve la taille de la pile (Pile)
Ends ; fin de la déclaration de la Pile
Data segment ; déclaration d'un segment de données qui va contenir les variables
    ; Ici on déclare les variables
Ends ; fin de la déclaration de données
Code segment ;déclaration du segment de Code qui va contenir le Code
    ; Ici on déclare les fonctions locales, globales et externes
Main: ;Point d'entrée du code
    Assume CS:Code, DS:Data, SS: Pile
    ;Assume permettant d'initialiser les registres de segments CS, DS et SS
    MOV AX, Data ; Initialisation du segment de données
    MOV DS, AX
    MOV AX, Pile ; Initialisation du segment de Pile
    MOV SS, AX

    _____
    ; À partir d'ici on peut placer nos lignes de code
    _____

Fin: ;Arrêt de programme
    MOV AH, 4Ch ; l'équivalent du return (0) en C
    INT 21h ; appel au MS-DOS
Ends ;fin de la déclaration du Code
End Main ; fin du point d'entrée du Code
```

Pensez lorsque vous écrirez un programme à bien partir de ce fichier squelette en le sauvegardant sous un autre nom avant de le modifier.

## Exercices

1. Créez un fichier TP1.asm identique au fichier squelette.asm qui permet d'afficher le message suivant :

**Bienvenue au monde de l'assembleur !.**

2. Écrire un programme qui permet de stocker le produit des éléments des deux tableaux de type byte dans un troisième tableau de type word.
3. Écrire un programme qui effectue le transfert des éléments d'un tableau vers la mémoire en utilisant l'adressage basé et indexé.
4. Écrire un programme qui permet d'inverser un tableau et le mettre dans un autre tableau en utilisant la pile.
5. Écrire un programme qui permet de déterminer la somme de deux tableaux.
6. Écrire un programme qui permet de déterminer le maximum d'un tableau.
7. Écrire un programme qui lit des caractères frappés au clavier et les affichent à l'écran tant que l'utilisateur n'appuie pas sur La touche <Spa> a le code ASCII 32= 20h.
8. Écrire un programme qui permet d'afficher le contenu de registre BL en binaire. L'idée est d'utiliser un masque BL AND 10000000b pour tester le bit de gauche, et de décaler BL à gauche pour parcourir ainsi tous les bits.
9. Écrire un programme qui affiche en binaire le code ASCII de chaque caractère saisi, en utilisant l'instruction de rotation qui convient.
10. Écrire un programme qui permet de lire un caractère au clavier et tester si son code ASCII est celui d'un chiffre. Si c'est le cas, afficher ce chiffre (et non son code ASCII), sinon afficher "N".
11. Écrire un programme qui permet de saisir une chaîne de caractères en minuscule et l'afficher en majuscule.