ISIMA 1 Assembleur **TP 2** : Programmes simples
Compte rendu *(titre, fichiers .LST, jeu d'essai copies d'écran et conclusion)*
Préparer ces deux problèmes avant de venir en TP !!!

1- Écrire un programme qui enlève les espaces d'une chaîne de caractères terminée par le caractère de code ASCII 0.

L'opération sera effectuée sur la chaîne originale à l'aide de deux registres pointeurs : un pour la lecture en mémoire et l'autre pour l'écriture en mémoire lorsque le caractère examiné est différent de ' ' (espace).
La chaîne sera implantée directement en mémoire.

2- Trier une liste de mots de 32 bits non signés

Algorithme de tri par sélection

```
Pour I de 1 à N-1 faire
        Pour J de I+1 à N faire
                Si L(I) > L(J) alors échanger L(I) et L(J)
                FinSi
        FinPour
FinPour
```

Avec séparation des fonctions compteur/indice :

```
I:=1;
Pour Cpt de N-1 à 1 pas -1 faire
        S:=Cpt; J:=I;
        Pour Cpt de  S à 1 pas -1 faire
                J:=J+1;
                Si L(I) > L(J) alors échanger L(I) et L(J)
                FinSi
        FinPour;
        I:=I+1; Cpt:=S
FinPour
```

Écrire un programme qui trie une liste **L** de **N** mots de 32 bits non signés, prédéfinie dans la partie « données ».
Bien séparer les fonctions compteur et index des indices de boucle.
Une petite liste sera directement implantée en mémoire (4 ou 5 mots).
Le jeu d'essai montrera la liste en mémoire avant et après le tri à l'aide de copies d'écran.

La copie d'écran s'effectue simplement avec la touche [Imp écr].
Quand l'écran est en mode graphique l'image est dans le presse-papier et elle peut être copiée dans une fenêtre du programme Paint (ou IrfanView).
Dans la fenêtre **Memory**, la sélection d'une partie puis clic droit et **Copy** a pour effet de copier le texte de la zone numérique hexadécimale sans les adresses ni les caractères à droite.

Tri à bulles en ordre croissant d'une liste de N nombres 32 bits signés
Solution simulation

```
        NAME    main
        PUBLIC  __iar_program_start
; --- Constantes ---

        SECTION .intvec : CODE (2)
        CODE32
__iar_program_start
        B       main

        SECTION .text : CODE (2)
        CODE32
main    NOP
; --- Code ---
Iter:
        MOV     r0, #0          ; indic = faux
        LDR     r1, =L          ; adr liste
        LDR     r12, N          ; nbre d'iter.
        SUB     r12, r12, #1    ; moins 1
Boucle:
        LDR     r2, [r1]        ; Li
        LDR     r3, [r1, #4]    ; Li+1
        CMP     r2, r3
        BLE     Raf             ; Rien à faire
        STR     r3, [r1]        ; permute
        STR     r2, [r1, #4]
        MOV     r0, #1          ; indic = vrai
Raf:
        ADD     r1, r1, #4      ; adr Li suivant

        SUBS    r12, r12, #1    ; compteur - 1
        BNE Boucle              ; remonte si =/= 0

        CMP     r0, #1          ; compare
        BEQ     Iter            ; remonte si vrai
        ; That's all folk !
        B .                     ; ici: branche ici

; --- Données variables ---
        DATA
N       DC32    4
L       DC32    -1, 2, -5, 1
        END
```