Exercice I (Manipulation des instructions de traitement de données):

(a) Somme des N premier entiers

- Ecrire un programme qui calcule, dans r0, la somme des N premiers entiers (N étant une constante définie dans le programme à l'aide de EQU).

(b) Calcul d'une multiplication

- Ecrire un programme qui calcule la multiplication de r0 par r1 et range le résultat dans r2, sans utiliser l'instruction de multiplication. Etudier les deux cas où les 2 nombres seraient positifs ou non.

Pour ceci, réalisez la multiplication par des additions successives. Faites attention aux signes (négatif ou positif) des opérandes et du résultat. Pensez à comparer vos opérandes à zéro pour connaitre leurs signes.

(c) Calcul d'une division

- Ecrire un programme qui calcule la division de r0 (dividende) par r1 (diviseur) et range le résultat dans r2 (quotient) et r3 (reste). Etudier les deux cas où les 2 nombres seraient positifs ou non.

Pour ceci, réalisez la division par des soustractions successives. Faites attention aux signes (négatif ou positif) des opérandes et du résultat. Pensez à comparer vos opérandes à zéro pour connaitre leurs signes.
Pour rappel, le nombre qui est divisé s'appelle le dividende ; le nombre qui divise s'appelle le diviseur et le résultat de l'opération s'appelle le quotient. Du coup :
Le dividende = Le quotient X diviseur + le reste. Le reste doit être inférieur au diviseur (en valeur absolue).

Exercice II (Manipulation des tableaux de bytes dans la mémoire):

Que fait le programme ci-dessous ? Quelle est la taille, en bytes (octets), de chacune des cases des deux tableaux « srcstr » et « dststr » ? Expliquez pourquoi on incrémente de un '1' r0 et r1 dans ce programme.

Ecrire ce programme dans un nouveau projet, le compiler et l'exécuter. Faire une exécution pas à pas et regarder le contenu des cases mémoires. Afficher le contenu en « Ascii ».

```
    AREA    StrCopy, CODE, READONLY
    ENTRY

start
    LDR     r1, =srcstr
    LDR     r0, =dststr

    BL      strcopy

stop
    MOV     r0, #0x18
    LDR     r1, =0x20026
    SWI     0x123456


strcopy
    LDRB    r2, [r1],#1         ; incrémentation de 1 de r1
    STRB    r2, [r0],#1         ; incrémentation de 1 de r0
```

```
        CMP     r2, #0
        BNE     strcopy
        MOV     pc,lr


        AREA    Strings, DATA, READWRITE
srcstr  DCB "First string - source",0
dststr  DCB "Second string - destination",0
```

Exercice II (Manipulation des mots de 32 bits « 4 bytes » dans la mémoire):

Ecrire un programme (dans un nouveau projet) qui recopie le contenu d'un tableau source dans un tableau destination en inversant l'ordre des éléments. Le premier élément du tableau source « src » sera placé à la dernière case du tableau destination « dst », le deuxième élément du tableau « src » sera placé à l'avant dernière case du tableau « dst », etc.

Faites une exécution pas à pas afin de voir ce qui se passe au niveau de la mémoire (aux adresses de src et dst).

```
src     DCD     1,2,3,4,5,6,7,8,1,2,3,4,5,6,7,8,1,2,3,4
dst     DCD     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```