TD3+TP3
Archeticture des ordinateurs

Exercice 1

Écrire le programme qui lit de l'utilisateur une valeur. Le programme doit dire si la valeur est positif ou non.

- Écrire le programme en C.
- Le traduire dans le langage MIPS.

```
.data
msg1 : .asciiz  "Donnez S.V.P. une valeur : "
msg2 : .asciiz  "La valeur est positive."
msg3 : .asciiz  "La valeur est négative."

.text
       li  $v0 , 4          # pour afficher msg1
       la  $a0 , msg1       #
       syscall              #

       li  $v0 , 5          # pour lire un entier
       syscall              # l'entier lu se met autmatiquement sur $v0
       move  $t0 , $v0      # déplacer l'entier lu vers un registre de travail $t0

       bltz  $t0 , pos      # vérifier si ($t0<0) ?, puis branchement si vrai

       li  $v0 , 4          # si ($t0>=0)
       la  $a0 , msg2       # afficher msg2
       syscall

       j  end_if

pos :  li  $v0 , 4          # afficher msg3 [si ($t0<0)]
       la  $a0 , msg3
       syscall

end_if:

       li  $v0 , 10  # mieux ajouter cette partie qui indique fin d'exécution
       syscall
```

Exercice 2

Écrire le programme qui doit lire de l'utilisateur une valeur positif. Le programme répète la lecture jusqu'à ce que l'utilisateur donne une valeur positive. Le programme affiche ensuite cette valeur.

- Écrire le programme en C.
- Le traduire dans le langage MIPS.

```
.data
message1 : .asciiz  "Donnez S.V.P. une valeur: "
message2 : .asciiz  "La valeur positif est : "

.text
loop:   li  $v0 , 4
       la  $a0 , message1
       syscall
```

```
        li  $v0 , 5
        syscall

        move  $t0 , $v0

        bltz  $t0 , loop

        li  $v0 , 4
        la  $a0 , message2
        syscall

        li    $v0 , 1
        move  $a0 , $t0
        syscall

        li    $v0 , 10
        syscall
```

Exercice 3

Écrire le programme qui lit de l'utilisateur une valeur, et affiche ensuite les 5 premiers multiples de cette valeur.

- Écrire le programme en C.
- Le traduire dans le langage MIPS.

```
.data

msg1 : .asciiz  "Donnez S.V.P. une valeur : "
msg2 : .asciiz  "Les multiples sont : "
msg3 : .asciiz  "  "

.text
        li  $v0 , 4
        la  $a0 , msg1
        syscall

        li  $v0 , 5
        syscall

        move  $t9 , $v0

         li  $t0 , 1
         li  $t2 , 0
         li  $t1 , 5

         li  $v0 , 4
        la  $a0 , msg2
        syscall

loop :       bgt  $t0 , $t1 , lab

        add   $t2 , $t2 , $t9
        addi  $t0 , $t0 , 1

        li  $v0 , 1
        move  $a0 , $t2
        syscall

         li  $v0 , 4
        la  $a0 , msg3
        syscall

        j  loop

lab  :  li  $v0 , 10
        syscall
```