**Solution Exo 3**

1. On souhaite écrire, en assembleur MIPS R3000, un programme qui calcule et affiche xy .

.data

x: .word 5

y: .word 4

p: .word 1

.text

\_start :

la $15, x

lw $16, 0($15)

la $15, y

lw $17, 0($15)

la $15, p

lw $18, 0($15)

for : blez $17, Affichage

mult $18, $16

mflo $18  
 addi $17, $17, -1

j for

Affichage : add $4, $18, $0

addi $2, $0, 1

syscall

addi $2, $0, 10

syscall

1. Rangement en mémoire des résultats intermédiaires suivants : x0, x1, …, x(y-1), xy.

.data

x: .word 5

y: .word 4

p: .word 1

.text

\_start:

la $15, x

lw $16, 0($15)

la $15, y

lw $17, 0($15)

la $15, p

lw $18, 0($15)

for : blez $17, Affichage

mult $18, $16

mflo $18

**addi $15, $15, 4**

**sw $18, 0($15)**  
 addi $17, $17, -1

j for

Affichage : addi $4, $18, 0

addi $2, $0, 1

syscall

addi $2, $0, 10

syscall