

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
UNIVERSITÉ 8 MAI 1945 – GUELMA
FACULTÉ MATH ET INFORMATIQUE ET SCIENCE DE MATIÈRE
DÉPARTEMENT DE L'INFORMATIQUE



Complément Serie TP 04 module AO avec solution

Promo 2017-2018

Table des matières

1	Exercice 01	2
1.1	Solution exercice 01	2
2	Exercice 02	3
2.1	Solution exercice 02	3
3	Exercice 03	5
3.1	Solution exercice 03	5
4	Exercice 04	7
4.1	Solution exercice 04	7
5	Exercice 05	8
5.1	Solution exercice 05	8

1 Exercice 01

Ecrire un programme MIPS qui calcule les 10 premières factorielles en utilisant une fonction.

1.1 Solution exercice 01

```

1  .data
2  val_a:.asciiz "\n fact de  "
3  egal:.asciiz " = "
4  .text
5
6  main:
7      li $t1,10
8      li $t2,1    # i=1
9      li $t3,1    # pour affichage
10     li $t5,1    # pour resultat
11  for:
12     ble $t2,$t1,lab    # if i<= 10 ==> lab
13  lab2:
14     li $v0,10          # else fin boucle
15     syscall
16  lab:
17     jal factorial    # ==> appel fonction factorial
18     li $v0,4
19     la $a0,val_a    # affichage de fact de
20     syscall
21     move $a0,$t3
22     li $v0,1
23     syscall
24     li $v0,4
25     la $a0,egal    # affichag de =
26     syscall
27     move $a0,$t5
28     li $v0,1          # affichage de resultat
29     syscall
30     addi $t2,$t2,1    # i++
31     addi $t3,$t3,1
32  j for
33
34  factorial:
35     mul $t5,$t5,$t2
36     jr $31

```

Resultat d'exécution :

```

fact de 1 = 1
fact de 2 = 2
fact de 3 = 6
fact de 4 = 24
fact de 5 = 120
fact de 6 = 720
fact de 7 = 5040
fact de 8 = 40320
fact de 9 = 362880
fact de 10 = 3628800
-- program is finished running --

```

2 Exercice 02

Ecrire un programme MIPS qui permet de trier par ordre décroissant les éléments d'un tableau **A** de **n** entiers positifs dans un nouveau tableau **B** de même dimension.

n étant un entier vérifiant $5 < n < 25$.

On utilisera la démarche suivante :

1. chercher le maximum de **A**
2. placer ce maximum dans **B**
3. remplacer le maximum par **-1** dans **A**
4. refaire les étapes 1, 2 et 3 jusqu'à ce que le tableau **A** soit entièrement composé de **-1**

2.1 Solution exercice 02

```

1  .data
2  tab:word 0,0,0,0,0,0,0,0,0,0
3  tab1:word 0,0,0,0,0,0,0,0,0,0,
4  size:word 10
5  vide:.asciiz " / "
6  tab_tri:.asciiz " Tab après tri décroissant : "
7  valn:.asciiz " Entrer val: "
8  .text
9
10 main:
11     la $s1,tab
12     la $t1,tab
13     la $s2,tab1
14     lw $t2,size
15     li $t3,1 # i=1
16     li $t4,1 # h=1
17     li $s7,0
18     li $s6,-1
19 while1:
20     bgt $t3,$t2,while
21     li $v0,4
22     la $a0,valn
23     syscall
24     li $v0,5
25     syscall
26     move $s5,$v0
27     sw $s5,0($s1)
28     addi $s1,$s1,4
29     addi $t3,$t3,1
30 j while1
31 while:
32     beq $s7,$t2,do
33     la $t1,tab
34     li $t3,1 # i=1
35     li $t4,1 # h=1
36     lw $t7,0($t1)
37     addi $t1,$t1,4
38 for1:
39     beq $t3,$t2 lab1
40     lw $t8,0($t1)

```

```

41     bgt $t8,$t7,lab2
42     addi $t3,$t3,1
43     addi $t1,$t1,4
44     j for1
45     lab2:
46     move $t7,$t8
47     addi $t3,$t3,1
48     addi $t1,$t1,4
49     j for1
50     lab1:
51     for2:
52     lw $t9,-4($t1)
53     beq $t9,$t7,lab3
54     addi $t4,$t4,1
55     subi $t1,$t1,4
56     j for2
57     lab3:
58     sw $s6,-4($t1)
59     labe2:
60     sw $t7,0($s2)
61     addi $s2,$s2,4
62     addi $s7,$s7,1
63     j while
64     do:
65     li $v0,4
66     la $a0,tab_tri
67     syscall
68     la $s3,tab1
69     li $s4,1
70     while2:
71     bgt $s4,$t2,lab5
72     lw $a0,0($s3)
73     li $v0,1
74     syscall
75     li $v0,4
76     la $a0,vide
77     syscall
78     addi $s3,$s3,4
79     addi $s4,$s4,1
80     j while2
81     lab5:

```

Resultat d'execution :

```

Entrer val: 1
Entrer val: 0
Entrer val: -1
Entrer val: 55
Entrer val: 99
Entrer val: 33
Entrer val: 20
Entrer val: 9
Entrer val: 8
Entrer val: 11
Tab après tri décroissant : 99 / 55 / 33 / 20 / 11 / 9 / 8 / 1 / 0 / -1 /
- program is finished running (dropped off bottom) --

```

3 Exercice 03

Deux entiers naturels strictement positifs m et n sont dits nombres amis si et seulement si :
La somme des diviseurs de m sauf lui-même est égale à n , et la somme des diviseurs de n sauf lui-même est égale à m .

Exemple :

220 et 284 sont deux nombres amis, en effet :

$D_{284} = \{1, 2, 4, 71, 142, 284\}$

$D_{220} = \{1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110, 220\}$

D_{284} et D_{220} sont respectivement les ensembles de tous les diviseurs de 284 et de 220.

$284 = 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110$

$220 = 1 + 2 + 4 + 71 + 142$

Ecrire un programme MIPS qui permet de déterminer puis d'afficher si deux entiers naturels donnés m et n sont amis ou non.

N.B. La solution doit comporter au moins un sous-programme.

3.1 Solution exercice 03

```

1  .data
2  val_n: .asciiz " Entrez la première val: "
3  val_N: .asciiz " Entrez la deuxième val: "
4  amis: .asciiz " Sont des amies"
5  neSontPasAmis: .asciiz " Ne sont pas amies"
6  .text
7
8  main:
9      li $v0,4
10     la $a0,val_n
11     syscall
12     la $v0,5
13     syscall
14     move $t1,$v0
15     move $t3,$v0
16     jal somme_div
17     move $t9,$t7
18     li $v0,4
19     la $a0,val_N
20     syscall
21     la $v0,5
22     syscall
23     move $t1,$v0
24     jal somme_div
25     move $s1,$t7

```

```

26 bne $t1,$t9,do
27 beq $t3,$s1,dol
28 do:
29     li $v0,4
30     la $a0,neSontPasAmis
31     syscall
32     j fin
33 dol:
34     li $v0,4
35     la $a0,amis
36     syscall
37 fin:
38     li $v0,10
39     syscall
40 somme_div:
41     li $t7,0
42     li $t2,1 # j=1
43 while:
44     beq $t2,$t1,lab
45     div $t1,$t2
46     mfhi $t5
47     beq $t5,0,lab1
48     addi $t2,$t2,1
49     j while
50 lab1:
51     add $t7,$t7,$t2
52     addi $t2,$t2,1
53     j while
54 lab:
55     jr $31

```

Resultat d'execution :

```

Entrez la première val: 10
Entrez la deuxième val: 6
Ne sont pas amies
- program is finished running --

```

```

Entrez la première val: 220
Entrez la deuxième val: 284
Sont des amies
- program is finished running --

```

4 Exercice 04

Ecrire un programme MIPS intitulé **DIVISION** qui permet de :

1. Saisir deux entiers naturels **a** et **b** strictement positifs.
2. Calculer puis afficher le quotient **q** et le reste **r** de la division euclidienne de **a** par **b** en utilisant la démarche suivante :

Etape1 : Initialiser un compteur **c** à 0.

Etape2 : Si $a < b$ alors aller à l'étape6.

Etape3 : Remplacer **a** par **a-b**.

Etape4 : Incrémenter le compteur **c**.

Etape5 : Aller à l'étape2.

Etape6 : Afficher le résultat de la division Euclidienne sous la forme suivante :

Le reste de la division est a

Le quotient de la division est c

N.B. La solution doit comporter au moins deux sous programmes.

4.1 Solution exercice 04

```

1  .data
2  val_A:.asciiz " Valeur de A: "
3  val_B:.asciiz " Valeur de B: "
4  quo:.asciiz  " Quotient de la division est: "
5  rest:.asciiz " Reste de la division est: "
6  .text
7  main:
8      li $v0,4
9      la $a0,val_A
10     syscall
11     li $v0,5
12     syscall
13     move $t1,$v0
14     li $v0,4
15     la $a0,val_B
16     syscall
17     li $v0,5
18     syscall
19     move $t2,$v0
20     jal fonction
21     li $v0,4
22     la $a0,rest

```

```

23     syscall
24     move $a0,$t1
25     li $v0,1
26     syscall
27     li $v0,4
28     la $a0,quo
29     syscall
30     move $a0,$t3
31     li $v0,1
32     syscall
33 fin:
34     li $v0,10
35     syscall
36 fonction:
37     li $t3,0 # c=0
38 while:
39     blt $t1,$t2 ,lab
40     sub $t1,$t1,$t2
41     addi $t3,$t3,1
42 j while
43 lab:
44     jr $31

```

Resultat d'exécution :

```

Valeur de A: 10
Valeur de B: 3
Reste de la division est: 1 Quotient de la division est: 3
-- program is finished running --

```

5 Exercice 05

On considère deux tableaux T_p et T_n contenant respectivement n entiers positifs et n entiers négatifs (n étant un entier naturel compris entre 2 et 20).

On désire afficher les éléments positifs et pairs du tableau TSOMME résultant de la somme de T_n et T_p tel que :

TSOMME[i] = $T_p[i] + T_n[i]$.

Ecrire un programme MIPS intitulé **SOMME_PAIR** qui permet de :

- saisir l'entier n ,
- saisir les tableaux T_n et T_p ,
- remplir le tableau TSOMME,
- afficher les éléments positifs et pairs du tableau TSOMME.

N.B. La solution doit comporter au moins deux sous programmes.


```

36  for2:
37      beq $t3,$t1,lab1
38      li $v0,4
39      la $a0,val_n
40      syscall
41      li $v0,5
42      syscall
43      sw $v0,0($s2)
44      addi $t3,$t3,1
45      addi $s2,$s2,4
46  j for2
47  lab1:
48      jal T_somme
49      la $s6,tab_somme
50      li $v0,4
51      la $a0,res
52      syscall
53  for4:
54      beq $t4,0,lab4
55      lw $t9,0($s6)
56      jal paire_positif
57      addi $s6,$s6,4
58      subi $t4,$t4,1
59  j for4
60  lab4:
61      li $v0,10
62      syscall
63  T_somme:
64      li $t4,0
65      la $s4,tab_p
66      la $s5,tab_n
67  for3:
68      beq $t4,$t1,lab3
69      lw $t5,0($s4)
70      lw $t6,0($s5)

```

```

71      add $t7,$t5,$t6
72      sw $t7,0($s3)
73      addi $t4,$t4,1
74      addi $s4,$s4,4
75      addi $s5,$s5,4
76      addi $s3,$s3,4
77  j for3
78  lab3:
79      jr $31
80  paire_positif:
81      li $t8,2
82      div $t9,$t8
83      mfhi $t3
84      beq $t3,0,labe
85      jr $31
86  labe:
87      bgez $t9, label1
88      jr $31
89  label:
90      move $a0,$t9
91      li $v0,1
92      syscall
93      li $v0,4
94      la $a0,vide
95      syscall
96      jr $31

```

Resultat d'execution :

```
Enter N entre 2 et 20: 6
Enter val positif: 3
Enter val positif: 4
Enter val positif: 1
Enter val positif: 2
Enter val positif: 3
Enter val positif: 7
Enter val negatif: -1
Enter val negatif: -2
Enter val negatif: -2
Enter val negatif: -2
Enter val negatif: -1
Enter val negatif: -1
Nbr positif et pair: 2 2 0 2 6
-- program is finished running --
```