

Fiche de TP N° 3 Architecture des ordinateurs (AO)

Exercice 1

Revenir sur l'exercice 6 de la fiche de TP2 permettant d'écrire le code MIPS pour calculer les N premiers éléments d'une suite de Fibonacci comme indiqué dans la fenêtre ci-dessous.

1. Modifier le programme déjà fait, en remplaçant la partie du code permettant d'afficher un séparateur entre chaque paire de termes par une procédure « **AfficheSeparateur** » permettant d'afficher le séparateur « - »
2. Mettre à jour le programme en remplaçant la procédure « **AfficheSeparateur** » par une procédure « **AfficheChaine** » permettant d'afficher une chaîne de caractères qui doit être passée en paramètre avant d'appeler cette procédure.
3. Ajouter, à la dernière version du programme, une procédure « **LireEntier** » permettant de lire un entier et faire appel à cette procédure aux endroits adéquats.
4. Ajouter, à la dernière version du programme, une procédure « **AfficheEntier** » permettant d'afficher un entier et faire appel à cette procédure aux endroits adéquats.

Exercice 2

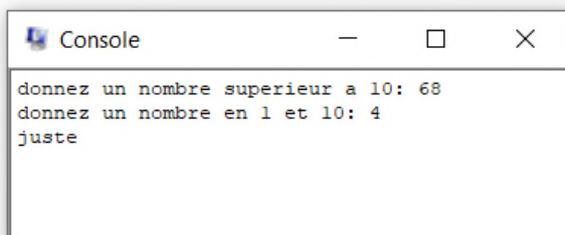
Écrire en assembleur MIPS le code permettant de :

1. Définir la fonction **int impair(int X)** retournant **1** si **X** est **impair** et **0** sinon.
2. Afficher un message à l'utilisateur, lui demandant de donner un nombre entier **X**.
3. Appeler la fonction **impair(X)**
4. Afficher un **1** si **X** est **impair** et **0** si **X** est **pair** en appelant une fonction **print_int**.

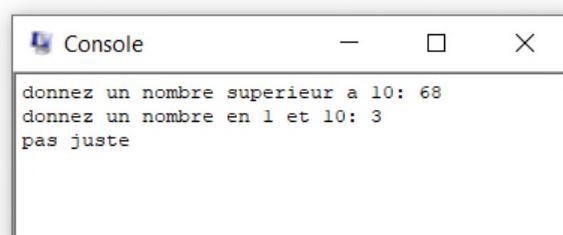
Exercice 3

Pour deux nombres saisis au clavier : Le premier supérieur à 10 et le deuxième entre 1 et 10.

Ecrivez un code mips pour appeler la **fonction diviseur** qui vérifie si le nombre 2 est un diviseur du nombre 1 et afficher le message « juste » sinon « pas juste » comme indiqué dans la fenêtre ci-dessous.



```
Console
donnez un nombre superieur a 10: 68
donnez un nombre en 1 et 10: 4
juste
```



```
Console
donnez un nombre superieur a 10: 68
donnez un nombre en 1 et 10: 3
pas juste
```

Exercice 4

Que fait ce programme ? et que stockent les registres suivants : \$t0, \$t1, \$t2 et \$t3

<pre> .data tab : .space 500 msg1 : .asciiz "Donnez la dimonsion de votre tableau : " msg2 : .asciiz "Inserer S.V.P. une valeur : " espace : .asciiz " " .text main: la \$t0 , tab addi \$t1 , \$zero , 0 li \$v0 , 4 la \$a0 , msg1 syscall li \$v0 , 5 syscall move \$t2,\$v0 remplire: bge \$t1 , \$t2 , suite li \$v0 , 4 la \$a0 , msg2 syscall li \$v0 , 5 syscall move \$t3,\$v0 </pre>	<pre> sw \$t3 , (\$t0) addi \$t0 , \$t0 , 4 addi \$t1 , \$t1 , 1 b remplire suite: la \$t0 , tab addi \$t1 , \$zero , 0 afficher: bge \$t1 , \$t2 , fin lw \$t3, (\$t0) li \$v0, 1 move \$a0, \$t3 syscall li \$v0, 4 la \$a0, espace syscall addi \$t0 , \$t0 , 4 addi \$t1 , \$t1 , 1 b afficher fin: li \$v0,10 syscall </pre>
---	--

Exercice 5

Ajouter au programme précédent le calcul de la somme des éléments du tableau.

Exercice 6

Ecrire un programme qui recherche le plus petit élément dans tableau saisi par l'utilisateur.