

Exercice 1 : (inverser un tableau)

Soit T un tableau initialisé de 10 entiers.

T (avant):

3	0	12	11	8	6	5	2	4	1
---	---	----	----	---	---	---	---	---	---

T (après):

1	4	2	5	6	8	11	12	0	3
---	---	---	---	---	---	----	----	---	---

Questions :

Écrire un programme en assembleur Mips, qui permet de :

- d'inverser un tableau
- Afficher le tableau résultant

Exercice 2 : (modification des occurrences d'un nombre)

Soit T un tableau initialisé de 10 entiers.

T :

2	4	2	5	6	8	11	2	0	3
---	---	---	---	---	---	----	---	---	---

Questions :

Écrire un programme en assembleur Mips, qui permet de :

- Permet de saisir un entier (cas : 2)
- Permet de saisir le nouveau entier (cas : 1)
- Modifier les occurrences du nombre saisi et afficher le tableau modifié

T :

1	4	1	5	6	8	11	1	0	3
---	---	---	---	---	---	----	---	---	---

Exo 2 : Let T be an array initialized with 10 integers.

Tasks:

Write a MIPS assembly program that:

1. Allows the user to input an integer (**case: 2**).
2. Allows the user to input a new integer (**case: 1**).
3. Modifies all occurrences of the entered number in the array and displays the modified array.

[Retour](#)

Exercice 3 : (tableaux pairs et Impairs)

Ecrire un programme en assembleur Mips, qui permet d'éclater un tableau T de 10 entiers supposés positifs en deux tableaux **T_pairs** et **T_impairs** contenant respectivement les nombres pairs et impairs de T. (**T_pairs** et **T_impairs** sont deux tableaux de cardinalité 10, initialisés à zéro au départ).

Exemple :

Au début :

T

T_pairs

T_impairs

5	7	1	6	0	15	9	4	2	18
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Après traitement :

T

T_pairs

T_impairs

5	7	1	6	0	15	9	4	2	18
6	0	4	2	18	0	0	0	0	0
5	7	1	15	9	0	0	0	0	0

Exercice 4 : (succession de deux valeurs)

Soit T un tableau initialisé de 10 entiers.

T

2	4	2	5	6	8	11	2	4	3
---	---	---	---	---	---	----	---	---	---

Question :

Écrire un programme en assembleur Mips, qui :

- Permet de saisir deux valeurs entières (cas : 2 et 4)
- Déterminer le nombre de succession de ces deux valeurs (V1=2 et V2=4) dans le tableau (réponse = 2 fois)

Exercice 5 : (insertion d'un nombre)

Soit $T(10)$ un tableau des réels ; on suppose que le tableau est déjà rempli et ordonné. On va :

- Demander un nombre à insérer
- Insérer le nombre dans la position adéquate (c à d, de telle façon à ce que le tableau reste ordonné)
- Affiche le tableau après insertion du nouveau nombre

Exemple :

T : 1 4 5 10 20 22 25 30 32

Donner x : 7

Tableau après insertion : 1 4 5 7 10 20 22 25 30 32

Exercice 6 : (recherche du PIC d'un tableau)

Soit T un tableau de N (N=10) entiers initialisés. Un élément est dit Pic s'il est supérieur à l'élément qui le précède et celui qui le succède dans le tableau. Ecrire un programme MIPS qui permet d'afficher tous les pics contenus dans le tableau T et leurs indices, ainsi que leur nombre.

Exemple :

N=10,

T=

0	3	-1	-2	7	5	10	8	11	34
---	---	----	----	---	---	----	---	----	----

Le programme doit afficher :

3 est un pic à l'indice 2,

7 est un pic à l'indice 5,

10 est un pic à l'indice 7.

Il y' a 3 pics dans ce tableau.

Exercice 7 : (comparaison de deux tableaux)

On veut comparer deux tableaux T1 et T2 et afficher 'Tableaux identiques' si on a le même contenu dans les deux tableaux et 'tableaux non identiques' sinon.

Exemple :

T1 : 1 2 3 4 5 6 7 8 9 10

T2 : 1 2 3 4 7 6 7 8 9 10

Résultat : Tableau non identiques

Exo 1 : Let T be an array initialized with 10 integers.

Tasks: Write a MIPS assembly program that:

- 1.Reverses an array.
- 2.Displays the resulting array.

Exo 2 : Let T be an array initialized with 10 integers.

Tasks:

Write a MIPS assembly program that:

- 1.Allows the user to input an integer (**case: 2**).
- 2.Allows the user to input a new integer (**case: 1**).
- 3.Modifies all occurrences of the entered number in the array and displays the modified array.

Exo 3: (Even and Odd Arrays)

Write a MIPS assembly program that splits an array T of 10 integers (assumed to be positive) into two arrays:

- T_pairs, containing the even numbers from T.
- T_impairs, containing the odd numbers from T.

Both T_pairs and T_impairs are arrays of size 10, initialized to zero at the beginning.

Exo 4: Let T be an array initialized with 10 integers.

Task:

Write a MIPS assembly program that:

- 1.Allows the user to input two integer values (**case: 2 and 4**).
- 2.Determines the number of occurrences of these two values in succession (e.g., **V1 = 2** followed by **V2 = 4**) in the array T.
(For example, the response would be **2 times**).

Exo 5 : Let T(10) be an array of real numbers. The array is assumed to be already filled and sorted.

The task is to:

- Ask for a number to insert.
- Insert the number in the appropriate position (i.e., in such a way that the array remains sorted).
- Display the array after inserting the new number.

Exo 6: (Finding the Peak of an Array)

Let T be an array of N integers (N = 10) that has been initialized.

An element is called a **Peak** if it is greater than the element that precedes it and the element that follows it in the array.

Write a MIPS assembly program that:

- Displays all the peaks in the array T along with their indices.
- Displays the total number of peaks found.

Exo 7: (Comparison of Two Arrays)

The goal is to compare two arrays, T1 and T2, and:

- Display "**Identical Arrays**" if both arrays have the same content.
- Display "**Non-Identical Arrays**" otherwise.