

# IFT 1227 – Architecture des ordinateurs

## Devoir 3

- À faire en groupe de **deux/trois** étudiants.
- Remise : Le 2 avril 2024 avant minuit.

### Programmation en assembleur MIPS

- Les boucles
- Les tableaux multidimensionnels
- Les fonctions et la décomposition fonctionnelle

### Tableaux multidimensionnels

Les tableaux multidimensionnels en assembleur sont modélisés par un grand tableau à une dimension. Travailler avec un tableau de **n x m** cases revient à travailler avec un tableau à une dimension. Supposons que **tab** est un tableau défini par l'adresse de son premier élément. Dans ce cas, la case à la position **[i] [j]** se trouve dans la mémoire à l'adresse :

**tab + (i \* m + j) \* taille d'un élément**

[0] [0]	[0] [1]	[0] [2]
[1] [0]	[1] [1]	[1] [2]

Lorsque **n = 2** et **m = 3**, les déplacements relatifs à l'adresse du début de tableau sont :

[0] [0] =  $0 \times 3 + 0 = 0$   
[0] [1] =  $0 \times 3 + 1 = 1$   
[0] [2] =  $0 \times 3 + 2 = 2$   
[1] [0] =  $1 \times 3 + 0 = 3$   
[1] [1] =  $1 \times 3 + 1 = 4$   
[1] [2] =  $1 \times 3 + 2 = 5$

0	1	2	3	4	5
---	---	---	---	---	---

### 1. Problématique

Une matrice **n x n** complétée avec les nombres **1, 2, 3, ... n<sup>2</sup>-1, n<sup>2</sup>** est un carré magique lorsqu'on trouve la même valeur un peu partout – c'est la somme des nombres:

- sur une ligne;

- sur une colonne;
- sur une diagonale;
- dans les coins du carré.

Par exemple, la matrice présentée plus bas est un carré magique. La valeur qu'on trouve partout est 34.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Écrire un programme qui permet de :

- a) Lire 16 valeurs de 1 à 16 saisies par l'utilisateur dans un ordre quelconque. L'ordre d'entrée des valeurs définit le positionnement des valeurs dans le carré.
- b) Vérifier si le carré avec les valeurs saisies représente un carré magique.

Pour implémenter ces fonctionnalités vous devez :

1. Demander à un utilisateur d'entrer 16 nombres de 1 à 16. L'ordre d'entrée correspondra à l'ordre des valeurs dans la matrice ligne par ligne : quatre nombres dans chaque ligne.
2. Permettre à l'utilisateur de saisir exactement 16 valeurs et pas plus.
3. S'assurer que la valeur de saisie courante ne soit pas entrée précédemment.
  - Pour implémenter cet aspect vous pourriez utiliser un tableau de vérification composé de 16 cases. Vu que chaque valeur de 1 à 16 doit être saisie une fois, vous pourriez au fur et à mesure mettre dans le tableau de vérification les valeurs saisies aux positions correspondantes. Si la position contient une valeur autre que zéro, cela veut dire que cette valeur a déjà été saisie et il faudra en demander une autre.
4. Imprimer la matrice en forme d'un carré.
5. Vérifier si la somme des éléments de chaque ligne, chaque colonne, dans les deux diagonales, dans les quatre coins représente la même valeur.

Implémenter au moins les fonctions suivantes :

- **creerMat(adresse, nbTs, nbCs)**  
 La fonction **creerMat** prend 3 paramètres : une adresse mémoire où le tableau représentant une matrice de deux dimensions sera stocké et sa taille en nombre des rangées et des colonnes et retourne l'adresse de l'espace mémoire disponible pour la prochaine allocation.  
 Il faut créer une matrice de 4 lignes et 4 colonnes dans la région mémoire **heap**, adresse 0x10040000, avec les valeurs saisies par l'utilisateur. Lors de la construction de la matrice, il faudra implémenter les fonctionnalités 1, 2 et 3 citées plus haut. Pour les tableaux auxiliaires, utilisez l'espace mémoire de **heap** disponible.
- **afficherMat(adrMatrice, nbRs, nbCs)**

La procédure **afficherMat** prend comme paramètres l'adresse de matrice (adresse de son premier élément) et sa taille : les nombres des rangées et des colonnes, et affiche la matrice spécifiée en plaçant les éléments en rangées et colonnes.

- **estMagique(adrMatrice, nbRs, nbCs)**  
Tester si une matrice représente le carré magique. La fonction doit prendre comme paramètres : l'adresse de matrice, le nombre des rangées et le nombre des colonnes et doit retourner soit la valeur magique trouvée ou -1 si la matrice ne représente le carré magique. Cette fonction doit faire les appels aux fonctions auxiliaires permettant de vérifier les conditions de détection des propriétés du carré magique.
- **main**  
La fonction **main** doit faire des appels à trois fonctions précédentes et afficher le résultat de la fonction **estMagique**.

Voici l'ébauche du programme:

```
#segment de la mémoire contenant les données globales
.data
...
msg:      .asciiz "... "

#segment de la mémoire contenant le code
.text
main:
...
    li    $v0,10    #terminer le programme
    syscall

#fonction creerMat - n'oubliez pas les commentaires!
creerMat:
...
#fonction afficherMat
afficherMat:
...
#fonctions auxiliaires de la fonction estMagique
...
#fonction estMagique
estMagique:
...
```

Les exemples des entrées et des sorties :

Entrée 1 :

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

La matrice n'est pas un carré magique!

Entrée 2 :

2  
6  
4  
17

Le nombre entré n'est pas valide. Entrez une valeur de 1 à 16.

3  
11  
8  
1  
2

La donnée est déjà entrée! Entrez une autre valeur de 1 à 16.

5  
6

La donnée est déjà entrée! Entrez une autre valeur de 1 à 16.

13  
14  
15  
16  
7  
8

La donnée est déjà entrée! Entrez une autre valeur de 1 à 16.

9	10	12	
2	6	4	3
11	8	1	5
13	14	15	16
7	9	10	12

La matrice n'est pas un carré magique!

Entrée 3 :

16

```

3
2
13
5
10
11
8
9
6
7
12
4
15
14
1
16      3      2      13
5      10     11      8
9      6      7      12
4      15     14      1
Carré magique !!! La valeur magique est:34

```

## Directives

Dans toutes les questions, il faut obligatoirement respecter toutes les **conventions MIPS** et utiliser les bonnes pratiques de la programmation.

L'information concernant les appels système se trouve dans le simulateur Mars, bouton Help => onglet MIPS => onglet Syscalls. Voici l'extrait de l'information concernant des appels système :

### Saisir une valeur à l'entrée standard et la faire stockée dans le registre \$t0 :

```

li    $v0, 5      # charger le numéro de service
syscall          # faire appel de ce service

```

```

# récupérer le résultat de $v0 dans $t0
add   $t0, $v0, $0

```

### Imprimer une valeur stockée dans le registre \$t0 :

```

li    $v0, 1      # service 1 imprime un entier

```

```

# charger la valeur à imprimer dans le registre $a0.

```

```

add   $a0, $t0, $zero    # Notre valeur se trouve dans $t0
syscall          # faire appel du service

```

### Imprimer une chaîne se trouvant à l'adresse msg

```

la    $a0, msg
li    $v0, 4      # Print string (Code 4 de syscall)
syscall          # faire appel du service

```

## Remise

Remettre le fichier source: **carreMagique.asm**

## Évaluation

Bon fonctionnement du programme :		
	Reproduire les 3 essais cités dans l'énoncé	30
	Les essais complémentaires	20
Programme adéquat (algorithme, déroulement du programme, conventions, structure etc.)		20
Interface utilisateur		5
Commentaires dans le programme		10
Contient en en-tête la description complète de ce que doit faire le programme		5
Contient les descriptifs: but, date, auteur(s), adresse(s) de courriel, code(s) permanent(s)		5
Le programme compile & le nom du programme source est <b>carreMagique.asm</b> et respect d'autres directives		5
Votre note:		/100