

Module : architecture des ordinateurs

Durée : 1h15

Consignes (Très importantes) :

- Le nom et les prénoms de l'étudiant doivent être écrits en lettres majuscules.
- L'usage du téléphone portable ou même le fait de le voir ou de le poser sur la table est strictement interdit.
- L'usage d'un effaceur ou correcteur est aussi interdit.

Exercice 1 (10 pts)

Ecrire un programme en assembleur MIPS R3000 qui exécute dans l'ordre l'algorithme suivant :

1. Déclarer les 4 messages qui sont sur la figure.
2. Afficher le 1^{er} message.
3. Lire la valeur entière entrée au clavier et la mettre dans le registre t1.
4. Afficher le 2^{em} message.
5. Lire la valeur entière entrée au clavier et la mettre dans le registre t2.
6. Afficher le 3^{em} message.
7. Lire la valeur entière entrée au clavier et la mettre dans le registre t3.
8. Afficher le 4^{em} message.
9. Si t3=1 alors aller à **traitement1**.
10. Se brancher inconditionnellement vers **saut1**.
11. **traitement1** :
12. Additionner t1 à t2 et mettre le résultat dans le registre t5.
13. Se brancher inconditionnellement vers **fin**.
14. **saut1** :
15. Si t3=2 alors aller à **traitement2**.
16. Se brancher inconditionnellement vers **saut2**.
17. **traitement2** :
18. Soustraire t2 de t1 et mettre le résultat dans le registre t5.
19. Se brancher inconditionnellement vers **fin**.
20. **saut2** :
21. Si t3=3 alors aller à **traitement3**.
22. Se brancher inconditionnellement vers **fin2**.
23. **traitement3** :
24. Multiplier t1 par t2 et mettre le résultat dans le registre t5.
25. Se brancher inconditionnellement vers **fin**.
26. **Fin** :
27. Afficher la valeur entière du registre a0.
28. **fin2** :

L'exécution du programme doit ressembler à ça :

```
Donnez le nombre a SVP:45
Donnez le nombre b SVP:15
Donnez le code operation SVP:1
Le résultat de l'opération est:60
```

```
Donnez le nombre a SVP:45
Donnez le nombre b SVP:15
Donnez le code operation SVP:8
Le résultat de l'opération est:
```

Exercice 2 (10 pts)

Ecrire un programme en assembleur MIPS R3000 qui exécute dans l'ordre l'algorithme suivant :

1. Déclarer un tableau de 400 octets.
2. Déclarer les 3 messages qui sont sur la figure.
3. Le registre t0 doit contenir l'adresse du tableau.
4. Le registre t1 est le registre compteur initialisé à 0.
5. Le registre t2 doit contenir le nombre d'éléments du tableau.
6. Le registre t3 doit contenir les valeurs des éléments du tableau.
7. Le registre t4 doit contenir la somme des éléments d'indices impairs du tableau (initialisé à 0).
8. Afficher le 1^{er} message et lire la valeur entière entre au clavier et mettre la valeur dans t2.

9. **Remplir_tab:**

Cette boucle permet de remplir un tableau avec des entiers de type **Word**.

Utilisez l'instruction **beq** (branch if equal) pour sortir de cette boucle.

10. **reset :**

Réinitialisez les registres t0 et t1

11. **calcule :**

Cette boucle permet de calculer la somme des valeurs des éléments d'indices impaires donc t1 est incrémenté de 2 et t0 est incrémenté de 8.

Utilisez l'instruction **bge** (branch if greater or equal) au début de la boucle pour sortir de cette boucle.

12. **afficher :**

Afficher le dernier message et afficher la valeur entière de t4.

L'exécution du programme doit ressembler à ça :

```
Donnez le nombre d'element de votre tableau SVP:5
Inserer l'element SVP:2
Inserer l'element SVP:6
Inserer l'element SVP:9
Inserer l'element SVP:4
Inserer l'element SVP:3
La somme des elements impairs est:14
```

Bonne chance