



**Mini Projet (contrôle N=°1)  
Assembleur MIPS R3000**

Ce Mini-Projet est décomposé en trois parties, dans chaque partie une tâche de recherche ou d'implémentation d'un code Assembleur MIPS R3000 est demandé. Les trois parties sont autonomes.

## Partie 1

Cherchez sur le web et répondez sur les questions suivantes :

1. Expliquer la méthode d'utilisation des fonctions en Assembleur MIPS, en donnant les différentes instructions utilisées.
2. Donner un exemple de code simple d'utilisation d'une fonction en Assembleur MIPS.
3. Expliquer la méthode d'utilisation de la Pile dans un programme assembleur MIPS.
4. Quand utilise-t-on la pile ?
5. Donner un exemple de code simple d'utilisation de la pile dans un programme Assembleur MIPS.

Remarque : Vous devez choisir des références de qualité : Livres, Cours universitaires, ou sites d'études reconnus.

## Partie 2

Soit le programme C de l'algorithme de tri de tableau suivant :

```
#include <stdio.h>
main()
{
    /* Déclarations */
    int A[50]; /*tableau donné */
    int N;    /*dimension    */
    int I;    /* indice élément tableau */
    int J;    /* indice élément tableau */
    int AIDE; /* pour la permutation */
    int PMAX; /* indique la position de l'élément */
              /* maximal à droite de A[I]    */

    /* Saisie des données */
    printf("Dimension du tableau (max.50) : ");
    scanf("%d", &N );
    for (J=0; J<N; J++)
    {
        printf("Elément %d : ", J);
        scanf("%d", &A[J]);
    }

    /* Affichage du tableau */
    printf("Tableau donné :\n");
    for (J=0; J<N; J++)
        printf("%d ", A[J]);
    printf("\n");

    /* Tri du tableau */
    for (I=0; I<N-1; I++)
    {
        /* Recherche du maximum à droite de A[I] */
        PMAX=I;
        for (J=I+1; J<N; J++)
            if (A[J]>A[PMAX]) PMAX=J;
        /* Echange de A[I] avec le maximum */
        AIDE=A[I];
        A[I]=A[PMAX];
        A[PMAX]=AIDE;
    }

    /* Edition du résultat */
    printf("Tableau trié :\n");
    for (J=0; J<N; J++)
        printf("%d ", A[J]);
    printf("\n");

    return 0;
}
```

1. Ecrire puis exécuter le code C précédant. N'hésiter pas à corriger le code s'il existe une erreur.
2. Traduire ce programme C en code assembleur MIPS.
3. Exécuter le code assembleur sur le simulateur QtSPIM.

## Partie 3

Soit le programme C de la résolution du problème de tour de Hanoi suivant :

```
#include<stdio.h>
#include<math.h>

void hanoi( int ndisque, char source, char intermediaire, char destination)
{
    if ( ndisque == 1)
        printf("Deplacer le disque de la tour ((%c)) a la tour ((%c)) \n", source, destination);
    else
    {
        hanoi ( ndisque - 1 , source , destination , intermediaire);
        hanoi( 1 , source , intermediaire , destination );
        hanoi ( ndisque - 1 ,intermediaire , source , destination );
    }
}

int main()
{
    int Nombre_Disque, Deplacement;
    printf("Entrer le nombre de disque a deplacer: \n");
    scanf("%d", &Nombre_Disque);
    Deplacement=pow(2,Nombre_Disque)-1;
    printf("Il faut %d deplacement pour deplacer les %d disques de la tour A vers la tour c en
    utilisant la tour B. \n",Deplacement,Nombre_Disque);
    hanoi(Nombre_Disque, 'A', 'B', 'C');
    printf("\n");
    return 0;
}
```

1. Ecrire puis exécuter du programme C précédant. N'hésiter pas a corrigé le code s'il existe une erreur.
2. Traduire ce programme C en code assembleur MIPS.
3. Exécuter le code assembleur sur le simulateur QtSPIM.

Votre rapport de TP doit être organisé comme suit :

- 1- Page de garde,
- 2- Sommaire,
- 3- Introduction,
- 4- Paritel : Explication de la méthode d'utilisation des fonctions et la pile avec MIPS.
  - a. Utilisation des fonctions en Assembleur MIPS.
  - b. Exemple de code Assembleur pour l'utilisation d'une fonction
  - c. Utilisation de la pile avec MIPS.
  - d. Exemple de code Assembleur pour l'utilisation de la pile.
- 5- Partie 2 : Traduction d'un programme C de tri de tableau vers un code Assembleur MIPS R3000.
  - a. Exécution du code C, vérification et correction du programme s'il existe une erreur.
  - b. Etapes de traduction du programme C vers le code Assembleur MIPS.
  - c. Exécution du code MIPS sur le simulateur QtSPIM.
- 6- Partie 3 : Traduction d'un programme C traitant le problème de tour de Hanoi en Assembleur MIPS.
  - a. Explication du problème de tour de Hanoi.
  - b. Explication de programme C fourni de la résolution de problème de tour de Hanoi.
  - c. Déroulement manuel du programme C fourni avec (Nombre Disque = 3).
  - d. Etapes de transformation du code C en code Assembleur MIPS.
  - e. Explication de la méthode d'utilisation de la pile.
  - f. Exemple d'exécution du code assembleur sur le simulateur QtSPIM .
- 7- Conclusion.
- 8- Bibliographie.

### Remarques d'ordre général:

- 1- Le travail doit être réalisé au maximum en trinômes, (Vous garder la même configuration que le TPI).
- 2- Chaque partie du travail est notée.
- 3- Toute copiée-collée d'une phrase ou code depuis internet ou depuis un autre trinôme, donnera la note « 0 » sur la partie concernée, (Par exemple, une copie collée sur l'explication du problème de tour de Hanoi sur internet engendra la note « 0 » dans cette partie). Il faut réaliser des résumés de ce que vous avez compris avec vos propres mots.
- 4- L'absence d'une partie du code ou rapport engendra la note « 0 » sur cette partie.
- 5- Dans le cas de copier-coller entre deux trinômes, alors les deux trinômes seront sanctionnés par la règle 3.
- 6- La taille du texte utilisée est 12 pt.