

TP4 : PROGRAMMES COMPLEXES (MULTIPLICATION, RECHERCHE, TRI)

Durée estimée : 4 heures

4.1 Objectifs détaillés

- Implémenter des algorithmes mathématiques en assembleur.
- Manipuler des structures de données (tableaux, listes).
- Comprendre les techniques de recherche et de tri.
- Optimiser le code en termes de taille et de vitesse.

4.2 Manipulations détaillées

Partie A : Multiplication 8 bits par 8 bits

1. Multiplication par additions successives :

- Multiplier \$25 (37 décimal) par \$0A (10 décimal).
- Algorithme : résultat = 0, tant que multiplicateur > 0, ajouter multiplicande et décrémenter multiplicateur.
- Implémenter avec une boucle.

2. Méthode optimisée (décalages et additions) :

- Utiliser l'algorithme de multiplication binaire (comme la multiplication manuelle).
- Pour chaque bit du multiplicateur (de droite à gauche) :
 - Si bit = 1, ajouter multiplicande décalé au résultat
 - Décaler multiplicande à gauche

Partie B : Division 8 bits

1. Division par soustractions successives :

- Diviser \$64 (100) par \$0A (10).
- Quotient = 0, tant que dividende >= diviseur, soustraire et incrémenter quotient.

- Gérer le reste.

2. Méthode optimisée :

- Implémenter la division par décalages (algorithme de la division binaire).

Partie C : Recherche dans une liste

1. Recherche séquentielle :

- Liste de 8 nombres à partir de \$50.
- Élément à chercher stocké en \$40.
- Si trouvé, stocker l'indice (position) en \$41, sinon mettre \$FF.
- Afficher le résultat en utilisant des LED (en TP6).

2. Recherche dichotomique (si la liste est triée) :

- Implémenter l'algorithme de recherche dichotomique.
- Comparer le nombre de pas avec la recherche séquentielle.

Partie D : Tri d'une liste

1. Tri à bulles (Bubble sort) :

- Liste de N nombres (N défini en début de programme).
- Implémenter l'algorithme :
 - ✓ pour $i = 0$ à $N-1$
 - ✓ pour $j = 0$ à $N-i-2$
 - ✓ si $\text{tableau}[j] > \text{tableau}[j+1]$
 - ✓ échanger $\text{tableau}[j]$ et $\text{tableau}[j+1]$
- Utiliser des boucles imbriquées et des branchements conditionnels.
- Afficher la liste triée en mémoire après exécution