

IFT 1215 – Introduction aux systèmes informatiques

Devoir 3

- À réaliser en groupe de deux ou trois étudiants.
- Date de remise : le 24 novembre 2024, à 23 h 59 au plus tard.
- Une pénalité de 10 % par jour de retard sera appliquée, avec un maximum de 3 jours de retard.
- Le rapport doit être clair, présentable et concis.
- **IMPORTANT** : Pour les programmes en LMC, il est essentiel que le code soit clair et bien commenté. Assurez-vous que le programme fonctionne dans le simulateur.
- Soumettez un fichier .zip contenant les deux programmes sur StudiUM.

I. Little Man Computer

- a) Écrire un programme en langage d'assemblage LMC qui cherche le plus grand commun diviseur (PGCD) avec l'algorithme d'Euclide. Les nombres x et y doivent être saisis par un utilisateur.

Algorithme d'Euclide cherche PGCD de deux nombres x et y :

- Soustraire le plus petit de x et y de l'autre nombre et répéter jusqu'à ce qu'ils soient égaux, ce sera le PGCD de x et y.

Votre programme doit afficher PGCD de deux nombres saisis.

- b) Trouver le plus grand élément du tableau **TAB** de 6 éléments, initialisé par l'utilisateur, en implémentant le tri par sélection.

Algorithme

Pour un tableau de nnn éléments :

- Rechercher le plus petit élément du tableau et l'échanger avec l'élément d'indice 0 ;
- Rechercher le deuxième plus petit élément du tableau et l'échanger avec l'élément d'indice 1 ;
- Continuer ainsi jusqu'à ce que le tableau soit entièrement trié.

L'étiquette **TAB** doit être la dernière déclaration dans votre programme : c'est à partir de cette adresse que les éléments du tableau seront placés. La taille de 6 doit être déclarée comme une constante.

Considérer comme exemple le programme `selfModify.lmc` (Exemples du Simulateur LMC) pour comprendre comment manipuler des tableaux en langage d'assemblage LMC.

2. CPU et Mémoire

Fetch-Execute

Supposons que le programme suivant soit chargé en mémoire à partir de l'adresse 0.

```
op    LDA    a
      ADD    s
      STO    s
      LDA    i
      SUB    one
      STO    i
      BRZ    aff
      LDA    op
      ADD    one
      STO    op
      BR     op
aff   LDA    s
      OUT
      HLT
s     DAT    0
i     DAT    3
one   DAT    1
a     DAT    10
      DAT    -3
      DAT    4
```

Traduisez le code assembleur de ce programme en code machine dans la colonne appropriée. Simulez l'exécution de ce programme en complétant les colonnes du tableau ci-dessous, en indiquant le contenu des registres à la fin de l'exécution de chaque instruction.

	Adresse	Code Machine	ACCU	IR	PC	MAR	MDR
Fin instruction 1	00						
Fin instruction 2	01						
Fin instruction 3	02						
...	...						
Fin instruction 14	15						
	16						
...							

3. Barème de correction

- Numéro 1 (50 pts) : 1a – 20 pts; 1b – 30;

- Numéro 2 (45 pts) : Registres pour 14 instructions $\times 0.5 = 35$ pts + Code machine : 20 lignes de code $\times 0.5$ par ligne = 10 pts;
- Présentation de rapport : 5 pts – propreté, lisibilité, respect des directives etc.
- Total : 100 points

Bon travail!