



**POLYTECHNIQUE
MONTRÉAL**

LOG1600

Architecture des micro-ordinateurs

Travail pratique #1: Périphériques et architecture

Groupe 6

Soumis par:

Dao, Ylena - 2207305

Trinh Huynh, Minh Tam Kevin - 2222954

13 février 2023

1.2. BARÈME

TP1		/4,00
Section 2		
Partie 1		/1,00
	Q1	/0,25
	Q2	/0,25
	Q3	/0,25
	Q4	/0,25
	Q5 (bonus)	/0,25
Partie 2		/1,00
	Q1 – bon fonctionnement	/0,25
	Q1 – extensibilité	/0,50
	Q1 – discussion	/0,25
Section 3		
Partie 1		/1,00
	Q1	/0,25
	Q2	/0,25
	Q3	/0,25
	Q4	/0,25
Partie 2		/1,00
	Q1 – bon fonctionnement	/0,50
	Q1 – extensibilité	/0,50

2.1. PARTIE 1

Q0/ Indiquez votre valeur de a dans votre rapport.

Notre valeur de a est de 5.

Q1/ Quelles sont les structures de contrôle (while, if/else, for...) que ce programme fait intervenir ?

Ce programme fait intervenir des sauts conditionnels comme le brz, qui peut être représenté avec un « if », et des sauts inconditionnels comme le br, qui peut être représenté par une boucle « while ».

Q2/ Écrivez une version en C du programme. Le code en C devrait pouvoir être exécuté sans erreur et donner le même résultat que le code Assembleur.

Voir le fichier intitulé TD1partie2.1.cpp

Q3/ Quelle valeur se retrouve à l'emplacement mémoire 0x0021 à la fin de l'exécution de ce programme ?

À la fin de l'exécution, la valeur de 0x00AA, qui est 170, se retrouve à l'emplacement mémoire 0x0021.

Q4/ Que fait ce programme ? Répondez à la question en décrivant le principe de fonctionnement du programme en évitant de référer à son contenu ligne par ligne.

Ce programme exécute une fonction mathématique, qui est :

$$valeur = \left(\sum_{i=0}^a i \right)^2 - \sum_{i=0}^a (i^2)$$

Q5/ Avec quelle valeur de a obtiendrait-on le résultat maximum en supposant qu'on une machine de 16 bits? (BONUS)

On obtiendrait le résultat maximum, qui est de 65535 ($2^{16} - 1$), avec une valeur de a de 22. Lorsque la valeur de a est de 23, cela donne un résultat 71852, ce qui dépasse le résultat maximum. Lorsque la valeur de a est de 22, cela donne un résultat de 60214, qui est plus petit que 65535.

2.2. PARTIE 2

Voici notre programme :

```

1  .text
2  ld nombreADiviser
3  st compteur
4
5  loop1:
6  ld nombreADiviser
7  sub nombreDiviseur
8  st nombreADiviser
9  brz estMult
10 ld compteur
11 brz estNonMult
12 ld compteur
13 sub incrementer
14 st compteur
15 br loop1
16
17 estNonMult:
18 stop
19
20 estMult:
21 ld incrementer
22 st resultat
23 stop
24
25 .data
26 nombreADiviser: 16
27 nombreDiviseur: 4
28 compteur: 0
29 incrementer: 1
30 resultat: 0

```

Voir le fichier intitulé TD1partie2.2.asm

Notre approche consiste à utiliser une boucle loop1 pour soustraire le nombreDiviseur du nombreADiviser jusqu'à tant que la différence soit 0. Si la différence est 0, cela signifie que le nombre est divisible par l'autre, et le programme va à l'étiquette estMult. Le résultat est alors stocké dans la variable resultat et le programme retourne 1. Si la différence n'est pas 0, cela signifie que le nombre n'est pas divisible par l'autre, et le programme s'arrête en allant à l'étiquette estNonMult et retourne 0.

Cependant, il y a quelques limites qui s'imposent sur le choix de ce processeur. En effet, ce processeur n'a pas d'instruction de division ou de comparaison « plus grand que », ce qui signifie que nous devons utiliser une approche de soustraction pour déterminer si un nombre est divisible par un autre. Si on avait l'instruction de division ou de comparaison, cela aurait évité d'avoir autant de lignes d'étapes, ce qui aurait été plus optimal.

3.1 PARTIE 1

Q0/ Indiquez votre valeur de a dans votre rapport.

Notre valeur de a est de 18.

Q1/ Quelles sont les structures de contrôle utilisées au sein de ce programme ?

Les structures de contrôle utilisées dans le programme sont des sauts conditionnels comme le `brz`, et des sauts inconditionnels comme le `br`.

Q2/ Quel est le contenu en mémoire à l'adresse 0x001F à la fin de l'exécution de ce programme ?

Le contenu est de 0x0000, qui est 0, à l'adresse 0x001F à la fin de l'exécution du programme.

Q3/ Que fait ce programme ? Répondez à la question en décrivant le principe de fonctionnement du programme en évitant de référer à son contenu ligne par ligne.

Lorsque la valeur de a est paire ou premier, le programme retourne 0. Sinon, si le a ne répond pas à ces deux critères, le programme retourne 1. Dans ce cas-ci, il est important de mentionner que les nombres 0, 1 et 2 ne sont pas pris en compte. En effet, la valeur de a ne pourrait jamais correspondre à ces valeurs, car la formule de a est : $a = 3 + |(\text{matricule } 1 + \text{matricule } 2) \% 21|$.

Q4/ Écrivez votre version en langage C du programme. Ce dernier devrait compiler et s'exécuter sans erreur, et ce, avec le même comportement et le même résultat que le code en assembleur fourni. Votre programme devrait inclure des commentaires, un échange avec l'utilisateur et une gestion intelligente de la mémoire qui devrait être libérée à la fin du programme (aucune fuite de mémoire). Enfin, votre programme doit être facilement extensible et modifiable.

Voir le fichier intitulé TD1partie3.1.cpp

3.2 PARTIE 2

Voici notre programme :

```
1  .text
2  ld terme
3  sub decrementer
4  st terme
5
6  boucle:
7  brz fin
8  ld n0
9  shl
10 add n1
11 st valeurProchaine
12 ld n1
13 st n0
14 ld valeurProchaine
15 st n1
16 ld terme
17 sub decrementer
18 st terme
19 br boucle
20
21 fin:
22 ld valeurProchaine
23 st valeurProchaine
24 stop
25
26 .data
27 terme: 9
28 n0: 0
29 n1: 1
30 valeurProchaine: 0
31 decrementer: 1
```

Voir le fichier intitulé TD1partie3.2.asm