

# IFT209 – Programmation système

## Université de Sherbrooke

### Laboratoire 1

Enseignant: Michael Blondin  
Date de remise: dimanche 26 janvier 2020 à 23h59  
À réaliser: en équipe de deux  
Modalités: remettre en ligne sur [Turnin](#)

Le but de ce laboratoire est d'écrire et d'exécuter votre premier programme en langage d'assemblage ARMv8, ainsi que de vous familiariser avec l'environnement de travail.

**Problème.** Nous disons qu'un triplet  $(a, b, c)$  d'entiers positifs est *ordonné* si  $a \leq b \leq c$ , et nous disons qu'il est un *triplet de Pythagore* si  $a^2 + b^2 = c^2$ . Par exemple:

(3, 4, 5) est un triplet de Pythagore ordonné;  
(1, 2, 3) est un triplet ordonné, mais pas de Pythagore;  
(4, 3, 5) est un triplet de Pythagore, mais non ordonné.

Vous devez écrire un programme, en langage d'assemblage de l'architecture ARMv8, qui:

- lit trois entiers positifs  $a, b, c$  de 64 bits;
- affiche la chaîne " : )" si  $(a, b, c)$  est un triplet de Pythagore ordonné;
- affiche la chaîne " : | " si  $(a, b, c)$  est un triplet de Pythagore non ordonné;
- affiche la chaîne " : ( " sinon.

**Tests.** Par exemple, dans un terminal, vous devriez obtenir:

```
3           1           4
4           2           3
5           3           5
: )        :(         : |
```

où les trois premières lignes sont les nombres entrés au clavier, et la **quatrième ligne** est la sortie affichée. Vous devez donc afficher *une seule* chaîne: " : )", " : | " ou " : ( " (avec saut de ligne).

#### Directives.

- Lisez les [instructions pour l'environnement de travail](#) afin d'assembler et d'exécuter votre programme;
- Votre programme doit être obtenu en complétant le code partiel de la page suivante;
- Votre programme doit être remis sur [Turnin](#) dans un seul fichier nommé `labo1.s`;
- Ne modifiez pas le point d'entrée ainsi que le format des entrées et sorties;
- Supposez que les valeurs en entrée sont valides; en particulier, elles seront numériques et positives;
- Si vous n'êtes pas certain-e de la façon d'organiser un programme en langage d'assemblage, lisez la brève section sur les [normes de programmation](#) à la fin du chapitre 3 des notes de cours;
- Consultez le [sommaire du jeu d'instructions](#) au besoin.

**Pointage.** Vous obtenez:

- 1,5 points si votre programme lit trois nombres;
- 1,5 points si votre programme affiche une sortie;
- 3 points si votre programme passe les trois tests ci-dessus;
- 2 points si votre programme affiche la bonne sortie sur d'autres triplets qui seront choisis à la correction;
- 2 points pour la lisibilité de votre code (indentation, commentaires et usage des registres).

**Code partiel.**

```
.global main

// Entrée: lit trois entiers positifs de 64 bits: a, b, c
// Sortie: ":)" si  $a \leq b \leq c$  et  $a^2 + b^2 = c^2$ 
// ...
// Usage des registres:
// x19 -- a
// ...
main:
    // Lire a
    adr x0, fmtEntree
    adr x1, nombre
    bl scanf // scanf(&fmtEntree, &nombre)
    ldr x19, nombre // a = nombre

    // Lire b
    /*
       code ici
    */

    // Lire c
    /*
       code ici
    */

    // Vérifier triplet (a, b, c) et afficher résultat
    /*
       code ici
    */

    mov x0, 0
    bl exit // quitter le programme

.section ".bss"
    .align 8
nombre: .skip 8

.section ".rodata"
fmtEntree: .asciz "%lu"
msgValide: .asciz ":)\n"
// ...
```