

## TP 3 : Conditionnelles Et Boucles

Prénom : \_\_\_\_\_ Nom : \_\_\_\_\_

### 1 Conditionnelles

- 1.1 Écrire la séquence d'instructions qui place dans un registre la valeur absolue du contenu de ce registre. En MIPS32 (compléter le fichier `mips32_abs.s`). En ARM avec un branchement (sans instruction conditionnelle). En ARM avec une instruction conditionnelle (sans branchement).

En code ASCII, les caractères 0 à 9 sont représentés sur 8 bits par 0x30 à 0x39. *A* et *B* sont chacun un caractère ASCII.

- 1.2 Pour chaque assembleur (MIPS32 et ARM) : écrire un programme qui écrit dans la variable *C* le maximum entre *A* et *B*. On écrira deux versions : une en utilisant les noms des variables (compléter le fichier `mips32_max_a_b_var.s`) et une en utilisant uniquement les calculs d'adresses en les repérant dans le simulateur (compléter `mips32_max_a_b.s`).

## 2 Boucles

```
for (i = 0; i < 8; ++i) | for (i = 0; i < 16; ++i) | for (i = 1; i < 7; ++i)
  s += X[i] + Y[i];    |   if (X[i] > 0) { s += X[i]; } |   Y[i-1] += X[i] + X[i-1];
```

X et Y sont des tableaux d'entiers (de 32 bits). On utilisera le chargement et l'écriture mémoire à partir des adresses.

**2.1 Écrire le code des trois boucles pour MIPS32 et ARM sans utiliser les noms de variable pour la première boucle (compléter les fichiers `mips32_loop_N.s`).**

2.2 Écrire le programme MIPS32 puis ARM qui écrit dans `min` la valeur minimale et dans `max` la valeur maximale du tableau `X` (en utilisant les noms de variables) (compléter `mips32_loop_min_max.s`).