TP 2 - Mémoire et boucles

On utilise le simulateur QtSpim. Les programmes sont disponibles sur le site du cours.

1. Endianité

Charger le programme TP2Ex1.s. Afficher l'onglet Data. La colonne la plus à gauche affiche la première de 16 adresses. La zone la plus à droite affiche l'interprétation ASCII des 16 octets correspondants.

a. Exécuter le programme. Donner le contenu en hexadécimal des registres $t0 et $t1 après l'exécution.
b. Décoder l'instruction correspondant à la pseudo-instruction la $t0, X et expliquer son action.
c. QtSpim utilise l'endianité de la machine sur laquelle il s'exécute. D'après le contenu de $t2, quelle est l'endianité de votre machine ?
d. A quelle adresse se trouve le caractère 'Q' dans la zone *User data segment*?
e. Compléter le programme en utilisant **une** instruction de rangement pour que la zone la plus à droite affiche «ArchitectureL2S3 ».
f. Décommenter la déclaration de la variable U et observer la zone de données. Que peut-on en conclure à propos du programme d'assemblage ?
g. Décommenter l'instruction lw $t1, 1($t0) et exécuter le programme. Expliquer le comportement observé.

2. Boucle

Un pseudo-code non optimisé du tri bulle est donnée figure 1.

```
procedure bubbleSort( A : list of sortable items )
  n = length(A)
  repeat
    swapped = false
    for i = 1 to  n-1 inclusive do
       if A[i-1] > A[i] then
       swap( A[i-1], A[i] )
       swapped = true
      end if
    end for
  until not swapped
end procedure
```

Figure 1 : tri bulle

Le programme TP2Ex2.s contient un squelette de programme MIPS32 implémentant le corps de cette procédure.

a. Charger un programme ne contenant que les trois premières lignes. Expliquer le contenu de la zone *User data segment*.
b. Quel doit être le contenu de cette zone après le tri ?
c. Compléter le programme et vérifier qu'il donne un résultat correct.