

## TP 5 : MIPS

## Exercice 0 - rappels

1. Que fait le programme ci-dessous ?

```
.text

main :
    la $t0,X
    lw $t1,N
    sll $t1,$t1,2
    add $t1,$t0,$t1
    add $t2,$zero,$zero
    la $t5,N

LOOP :
    lw $t3,0($t0)
    add $t2,$t2,$t3
    addi $t0,$t0,4
    add $t5,$t5,$t2
    bne $t0,$t1,LOOP
    sw $t5,S

    jr $ra

.data
N : .word 10
X : .word 1,3,5,8,11,13,18,21,24,25
S : .word 0
```

Par la suite vous allez utiliser le logiciel `QtSpim` qui est un simulateur MIPS distribué sous licence BSD. Ce logiciel permet de visualiser l'état de l'espace utilisateur (instructions, mémoire, registres etc). Privilégiez l'exécution pas à pas (touche F10) pour voir ce qui se passe après chaque instruction pour vous familiariser.

## Exercice 1 - QtSpim

*pour le moment ça crash moins que TkGate*

1. Écrire et tester le programme assembleur qui fournit l'indice  $i$  du plus grand élément d'un tableau de 10 entiers.
2. Téléchargez les fichiers `helloworld.s` et `io.s` sur le site :

<http://perso.ens-lyon.fr/petru.valicov/Teaching.html>.

Exécutez-les et observez ce qui se passe.

3. Donner le code assembleur du programme ci-dessous. On supposera que le tableau est de taille 10. Testez votre code.

```
void triBulles( A : liste d'entiers ) {
  n = taille(A);
  repeat
    newn = 0;
    for i=1 to n-1 do
      if (A[i-1] > A[i] then
        swap(A[i-1],A[i]);
        newn = i;
      end if;
    end for;
    n = newn;
  until n = 0;
}
```

4. (**Bonus**) : Réalisez une version récursive de l'algorithme.

## Exercice 2 - Avant-goût des pipelines

*it tastes like copper*

Téléchargez le fichier `nope.s`. Exécutez le programme avec les options du simulateur par défaut. Ensuite réexécutez-le en cochant l'option *Enable Delayed Branches* dans l'onglet *MIPS* des paramètres (*Simulator -> Settings*). Conclure.