In-Class Exercise 8

Submit your work to the moodle before the deadline

1. Write a function that updates an array of 1024 bytes repeatedly (i.e., read, add one in the value of every byte and write the modified values) using four parameters, ***int arrSize***, ***int stepSize***, ***int repCount*** and ***int loopSize***. Use the MIPS conventions to pass the parameters.

   The signature of this procedure in C would look like this:

```
void  myMemoryUpdate (int repCount, int loopSize, int arrSize, int stepSize){
    // an array type could be different  (e.g.,, char array[arrSize]; )

    for (int repIdx=0; repIdx < repCount ; repIdx ++){
        for (int index = 0; index < arrSize; index += stepSize) {
            // do read, add one in every byte and write
            //e.g.,  array[index] = array[index] + 1;    // for char type, stepSize =1
        }
    }

    // Another way
    for (int index = 0; index < arrSize/loopSize; index ++) {
        for (int repIdx=0; repIdx < repCount ; repIdx ++) {
            for (int loopIdx = 0; loopIdx < loopSize ; loopIdx += stepSize){

                // read, add one in every byte and write
            }
        }
    }
}
```

   Example of array declaration:

```
array:   .space   1024
```

   Example of Read Data of the array (for 1024 bytes) :

Address	value(+0)	value(+4)	value(+8)	value(+c)
0x1000 :	**0x00000000**	**0x00000000**	**0x00000000**	**0x00000000**

   Example of Write Data of the array (for 1024 bytes) :

Address	value(+0)	value(+4)	value(+8)	value(+c)
0x1000 :	**0x04040404**	**0x04040404**	**0x04040404**	**0x04040404**

2. (**Optional**) And then, try to improve cache performance with optimizing (actually better) your **assembly source code** and **cache organization parameters** in the **Data Cache Simulator** Tool. We assume that the memory performance metric is like below and **lower value is better**. (The Miss Penalty was not considered currently).

   Cache performance metric:
   Memory Access Count X Cache Miss Rate
   = Memory Access Count X (100 – Cache Hit Rate + 1(offset for 100% hit rate))

Note 1: How to activate the Cache and Memory related Tools

Run **Tools-->Data Cache Simulator.**
Enable the Runtime Log and then click "Connect to MIPS".
Run **Tools-->Memory Reference Visualization**
Click "Connect to MIPS".