In-Class Exercise 7

Submit your work to the moodle before the deadline

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

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

   void myMemoryUpdate (**int arrSize, int stepSize**);

   Example of array declaration:

   ```
   array:   .space   1024
   ```

   Example of Read Data of the array :

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

   Example of Write Data of the array :

Address	value(+0)	value(+4)	value(+8)	value(+c)
0x1000 :	**0x01010101**	**0x01010101**	**0x01010101**	**0x01010101**

2. 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 (i.e., **number of blocks** and **cache block size**). The cache size must be same with the default (i.e., 128 bytes). We assume that the memory performance metric is like below and **lower value is better**. (The Miss Penalty was not considered because of the limitation of the tool).

   Cache performance metric:
   Memory Access Count X Cache Miss Rate
   = Memory Access Count X (100 – Cache 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".

Note 2: You may change **ONLY the number of blocks** and **the cache block size** in the Data Cache Simulator Tool.

Note 3: You **MUST** write your optimized (better) parameters next **.text** as comments like below (if there are no parameters, it will be graded with the default values, it is **each student's responsibility**):

```
        .text
#Number of blocks:
#Cache block size:
# YOUR METRIC SCORE:
# The reasons for my optimization (better):
#1) In Assembly code:
#2) In the configurations of cache parameters:
```

Note4: A benefit will be given to the top 3 students, who have the lowest numbers in the metric.