CSE 140 Lab/HW#2 – Due: see below

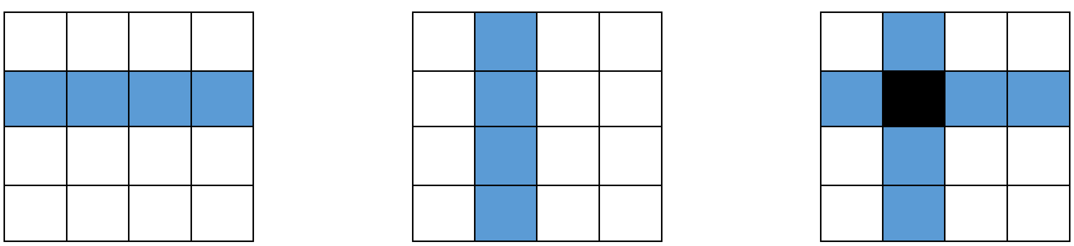
MIPS assembly (80pts)

1. Implement a matrix multiply code in MIPS assembly. The skeleton code is provided. The input data are also embedded to the skeleton code (do not modify). Fill your code between the following two comment lines in the skeleton code. Do not change any other code in the skeleton code. You can test and debug your code in MARS simulator.

```
# Your CODE HERE
   ← You can add new code here.
# End CODE
```

Matrix Multiplication: Matrix multiplication is a binary operation that takes a pair of matrices, and produces another matrix. The pseudo code and the graphical projection of the matrix multiplication of N×N square matrix is like below:

```
for (i = 0; i < N; i++)
    for (j = 0; j < N; j++)
        for (k = 0; k < N; k++)
            C[i][j] = C[i][j] + A[i][k] x B[k][j];
```



A[i][k], k = 0 to N   B[k][j], j = 0 to N   C[i][j], shown in black

Figure 1 Matrix accesses made for (i,j) = (1,1) (K iterating 0 to N-1).

Requirements and guidelines:

1. Test your program with the values of 'matrix_a' and 'matrix_b' provided. The output 'matrix_c' that is printed by the provided PRINT_MAT routine should match the matrix shown below.

   Result for test case:

3510	3588	3666	3744	3822	3900	3978	4056	4134	4212	4290	4368
13158	13380	13602	13824	14046	14268	14490	14712	14934	15156	15378	15600
22806	23172	23538	23904	24270	24636	25002	25368	25734	26100	26466	26832
32454	32964	33474	33984	34494	35004	35514	36024	36534	37044	37554	38064

42102	42756	43410	44064	44718	45372	46026	46680	47334	47988	48642	49296
51750	52548	53346	54144	54942	55740	56538	57336	58134	58932	59730	60528
61398	62340	63282	64224	65166	66108	67050	67992	68934	69876	70818	71760
71046	72132	73218	74304	75390	76476	77562	78648	79734	80820	81906	82992
80694	81924	83154	84384	85614	86844	88074	89304	90534	91764	92994	94224
90342	91716	93090	94464	95838	97212	98586	99960	101334	102708	104082	105456
99990	101508	103026	104544	106062	107580	109098	110616	112134	113652	115170	116688
109638	111300	112962	114624	116286	117948	119610	121272	122934	124596	126258	127920

2. You should be careful to calculate the address of an element of the 2D array. The address of an element (i, j) in a 2D array can be calculated as "base address of the matrix + (row_size × i + j) × element_size".
3. You may refer to MIPS Reference Data used for Lab 1 to find useful instructions. You can also use any pseudo-instructions you desire (i.e. "li – load immediate", "la – load address", and etc.).
4. This programming homework awards extra credit to the best five MIPS programmers. The five students who used the smallest number of instructions for their correctly working solution will earn 5% extra credit (5% of Lab 2 score). If there is any tie, the one who submitted the solution earlier gets the credit (the time recorded in CatCourse). You can check the instruction count of your code like below:

    1) Assemble your code.
    2) Click "Tools->Instruction Counter" in the MARS menu bar. Then, the Instruction Counter tool will appear.

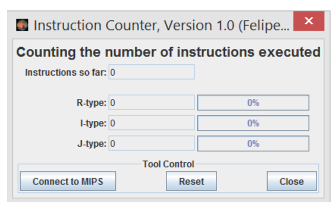


Figure 2 Instruction Counter tool

    3) Click "Connect to MIPS" button of the Instruction Counter tool so the tool can begin monitoring the MARS simulation.
    4) Run your code (you can still step through or set breakpoints if desired).

5. Name your source file as "cse140_your-first-name_your-last-name_mm.s" and submit it to CatCourse.

Performance (20pts)

2. Consider two processors P1 and P2 executing the same instruction set. And, their clock rate and CPI are like below:

	P1	P2
Clock rate	3 GHz	4 GHz
CPI	1.5	2

For the two processors, solve the following problems. Round off the calculated results to two decimal places if needed.

a. Which processor has the highest performance expressed in instructions per second?

b. Suppose that both processors took 100 seconds to run a program. We want to achieve 2x speedup in running this program on both processors by increasing clock frequency. But, we found that the frequency increase led to an increase of CPI by 10% and 20% for P1 and P2, respectively. So, we adjusted the clock frequency to accommodate the new CPI. What would be the final clock frequency of each of the two processors?

Submission Guideline

- Submit the matrix multiply code in an assembly file namely "cse140_your-first-name_your-last-name_mm.s", and your solution for the performance calculation problem in a separate MS Word or a pdf format to the CatCourse.
- Deadline: **11:59PM of one day before the next lab** (If this lab is assigned on 2/9, the deadline is 2/15 11:59PM)