Com S 321 **Homework 5** **Due on Thursday, Oct. 5 by Midnight**

1. Write a MIPS program fragment for the following high-level language statement:

a = b + 100;

Assume that a is in register *$t0* and b is in register *$t1*.

2. Write a MIPS program fragment for the following high-level language statement:

g = h + A [14]; // Assume A is an int array with each element taking 4 bytes

Assume that g is in register *$t0*, h is in register *$t1*, and the base address of A is in register *$s1*.

3. Write a MIPS program fragment for the following high-level language statement:

A [275] = g + A [275]; // Assume A is an int array with each element taking 4 bytes

Assume that g is in register *$t0*, and the base address of A is in register *$s1*.

4. Write a MIPS program fragment for the following high-level language statement:

g = h + A [i]; // Assume A is an int array with each element taking 4 bytes

Assume that g is in register *$t0*, h is in register *$t1*, the base address of A is in register *$s1*, and i is in register *$s2*.

5. Write a MIPS program fragment for the following high-level language statement:

x [10] = x [11] + c; // Assume x is an int array with each element taking 4 bytes

Assume that c is in register *$t0*, and the base address of x is 4,000,000 in base 10. (Hint: You should use the Load Upper Immediate or *lui* instruction).

6.

(a) Write a MIPS program fragment for the following high-level language statement:

B [350] = g + B [350]; // Assume B is an int array with each element taking 4 bytes

Assume that g is in register *$s1*and that the base address of the array B is in register *$s2*.

Assume that registers *$s0* through *$s7* map to register numbers 16 through 23, and registers *$t0* through *$t7* map to register numbers 8 through 15. Use the instruction format information provided below to convert your instructions in Part (a) to machine language code. First express your machine language code using decimal numbers for the fields and then convert into Binary and eventually into hexadecimal representation for each instruction.

(b) Instructions for the program in Part (a) converted into Decimal

(c) Conversion of Part (b) into Binary

(d) Conversion of Part (c) into Hexadecimal

Instruction Format of relevant MIPS instructions

	6 bits	**5 bits**	**5 bits**	**5 bits**	**5 bits**	**6 bits**
add rd, rs, rt	0	rs	rt	rd	0	32

	6 bits	**5 bits**	**5 bits**	**16 bits**
lw rt, *d* (rs)	35	rs	rt	16-bit displacement *d*
sw rt, *d* (rs)	43	rs	rt	16-bit displacement *d*

Hosam Abdeltawab

H.w #5

1

1] a = b + 100; // a in $t0, b in $t1

```
addi $t0, $t1, 100
```

2] g = h + A[14]; // g in $t0, h in $t1, & A is in $S1

```
Lw $t2, 56($S1)
add $t0, $t1, $t2
```

3] A[275] = g + A[275]; // g in $t0, A in $S1

```
Lw $t1, 1100($S1)
Add $t0, $t0, $t1
SW $t0, 1100($1)
```

4] g = h + A[i]; // g in $t0, h in $t1, A in $S1, & i in $S2

```
Add $t2, $S2, $S2
Add $t2, $t2, $t2
Add $t2, $t2, $S1
Lw $t3, 0($t2)
Add $t0, $t1, $t3
```

5] X[10] = X[11] + C; // C in $t0, X base address is 4,000,000

$(4,000,000)_{10}$ = 0x 003D0900

```
LUi $t1, 0x003D
ORi $t1, $t1, 0x0900
Lw $t2, 44($t1)
Add $t3, $t2, $t0
SW $t3, 40($t1)
```

6 (a) B[350] = g + B[350]; // g in $s1, B in $s2

```
Lw   $t0, 1400($s2)
Add  $t1, $s1, $t0
Sw   $t1, 1400($s2)
```

$\$s_1 \rightarrow 17_{10}$

$\$s_2 \rightarrow 18_{10}$

$\$t_0 \rightarrow 8_{10}$

$\$t_1 \rightarrow 9_{10}$

~~(a)~~ (b)

Lw $t0, 1400($s2)	35	18	8	1400		
Add $t1, $s1, $t0	0	17	8	9	0	32
Sw $t1, 1400($s2)	43	18	9	1400		

~~(b)~~ (c)

Lw $t0, 1400($s2)	100011	10010	01000	0000010101111000		
Add $t1, $s1, $t0	000000	10001	01000	01001	00000	100000
Sw $t1, 1400($s2)	101011	10010	01001	0000010101111000		

(d)

③

8E480578
02284820
AE490578