EGC-442 HW# 2_ Solution Dr. Izadi

First Name:________________________ Last Name: ________________________

Problem 1 (10 Pt)
For the following C statement, what is the corresponding MIPS assembly code? Assume that the C variables f, g, and h, have already been placed in registers $s0, $s1, $s2, respectively. Use a minimal number of MIPS assembly instructions.

f = g + (h − 5);

```
addi $s0, $s2, -5
add $s0, $s0, $s1

[addi f,h,-5 (note, no subi) add f,f,g]
```

Problem 2 (10 Pt)
Write a single C statement that corresponds to the two MIPS assembly instructions below.

add f, g, h
add f, i, f

```
f = g + h + i
```

Problem 3 (10 Pt)
For the following C statement, write the corresponding MIPS assembly code. Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively.

B[8] = A[i − j];

```
sub $t0, $s3, $s4
sll $t0, $t0, 2
add $t0, $s6, $t0
lw $t1, 0($t0)
sw $t1, 32($s7)
```

Problem 4 (10 Pt)

Translate the following C code to MIPS. Assume that the variables, f,g. h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of A and B are in registers $s6 and $s7, respectively. Assume that the elements of the arrays A and B are 4-byte words:

B[8] = A[i] + A[j];

```
sll $t0, $s3, 2 # $t0 <-- 4*i
add $t0, $t0, $s6 # $t0 <-- Addr(A[i])
lw $t0, 0($t0) # $t0 <-- A[i]
sll $t1, $s4, 2 # $t1 <-- 4*j
add $t1, $t1, $s6 # $t1 <-- Addr(A[j])
lw $t1, 0($t1) # $t1 <-- A[j]
add $t0,$t0,$t1 # t1 <-- A[i] + A[j]
saw $t0,32($s7) # B[8] <-- A[i] + A[j]
```

Problem 5 (10 Pt)

For each MIPS instruction in Exercise 2.8, show the value of the opcode (op), source register (rs) and funct field, and destination register (rd) fields. For the I-type instructions, show the value of the immediate field, and for the R-type instructions, show the value of the second source register (rt).

	type	opcode	rs	rt	rd	immed
addi $t0, $s6, 4	I-type	8	22	8	--	4
add $t1, $s6, $0	I-type	0	22	0	9	--
sw $t1, 0($t0)	I-type	43	8	9	--	0
lw $t0, 0($t0))	I-type	35	8	8	--	0
add $s0, $t1, $t0	R-type	0	9	8	16	--

Problem 6 (10 Pt)
Assume that registers $s0 and $s1 hold the values 0x80000000 and 0xD0000000, respectively.

(a) What is the value of $t0 for the following assembly code?

```
add $t0, $s0, $s1
```

0x50000000

(b) Is the result in $t0 the desired result, or has there been overflow?

Overflow

(c) For the contents of registers $s0 and $s1 as specified above, what is the value of $t0 for the following assembly code?

```
sub $t0, $s0, $s1
```

0xB0000000

(d) Is the result in $t0 the desired result, or has there been overflow?

no overflow

(e) For the contents of registers $s0 and $s1 as specified above, what is the value of $t0 for the following assembly code?

```
add $t0, $s0, $s1
add $t0, $t0, $s0
```

0xD0000000

(f) Is the result in $t0 the desired result, or has there been overflow?

overflow