Programming Exercises

For these programming exercises, use only those instructions that have been discussed so far in these notes:

add	div	mflo	slt, slti
addi	divu	mult	sltu, sltiu
addiu	j	multu	sra
addu	lb	nor	srl
and	lbu	or	sub
andi	lh	ori	subu
beq	lhu	sb	sw
bgez	lui	sh	xor
bltz	lw	sll	xori
bne	mfhi		

In the *Settings* menu of SPIM set Bare Machine ON, Allow Pseudo Instructions OFF, Load Trap File OFF, Delayed Branches ON, Delayed Loads ON, Mapped IO OFF, Quiet OFF.

Run the programs by setting the value of the PC to 0x400000 and then single stepping (pushing F10) or by multiple stepping (push F11 and enter a number of steps). Observing the results in the SPIM window.

A carefully thought-out and debugged flow chart will cut in half the time it takes to do one of these exercises. Plus you will learn good coding techniques and gain insight into programming languages and language translation.

1. Exercise - Array Maximum and Minimum

Declare an array of integers, something like:

```
        .data
size:   .word 8
array:  .word 23, -12, 45, -32, 52, -72, 8, 13
```

Write a program that determines the minimum and the maximum element in the array. Assume that the array has at least one element (in which case, that element will be both the minimum and maximum.) Leave the results in registers.

2. Exercise — Ascending Numbers

Declare an array of integers, something like:

```
        .data
size:   .word 10
array:  .word 2, 4, 7, 12, 34, 36, 42, 8, 57, 78
```

Write a program that determines if the numbers form an increasing sequence where each integer is greater than the one to its left. If so, it sets a register to 1, otherwise it sets the register to 0.

Of course, write the program to work with an array of any size, including 0. Arrays of size 0 and size 1 are considered to be ascending sequences. The array can contain elements that are positive, negative, or zero. Test the program on several sets of data.

3. Exercise — Paired Data

In this program data comes in pairs, say height and weight:

```
        .data
pairs:  .word 5             # number of pairs
        .word 60, 90        # first pair: height, weight
        .word 65, 105
        .word 72, 256
        .word 68, 270
        .word 62, 115
```

Write a program that computes the average height and average weight. Leave the results in two registers.

For these programming exercises, use only those instructions that have been discussed so far in these notes:

add	div	mflo	slt, slti
addi	divu	mult	sltu, sltiu
addiu	j	multu	sra
addu	lb	nor	srl
and	lbu	or	sub
andi	lh	ori	subu
beq	lhu	sb	sw
bgez	lui	sh	xor
bltz	lw	sll	xori
bne	mfhi		

In the *Settings* menu of SPIM set Bare Machine ON, Allow Pseudo Instructions OFF, Load Trap File OFF, Delayed Branches ON, Delayed Loads ON, Mapped IO OFF, Quiet OFF.

Run the programs by setting the value of the PC to 0x400000 and then single stepping (pushing F10) or by multiple stepping (push F11 and enter a number of steps). Observing the results in the SPIM window.

4. Exercise — Median of Three

Write a program that computes the median of three values in memory. After it has been found, store the median in memory.

```
    .data
A:  .word 23
B:  .word 98
C:  .word 17
```

The median of three integers is greater than or equal to one integer and less than or equal to the other. With the above three integers the median is "23". Assume that the data changes from run to run. Here is some more possible data:

```
    .data
```

A: .word 9
B: .word 98
C: .word 9

With the new data, the median is "9".