Name:_____

Problem 1: (20 points) Short Answers (10 minutes)

1. **(5 points)**: What is the result in register $t0 after these lines of MIPS assembly code are executed?

```
andi $t0, $t0, 0x00000000
ori  $t0, $t0, 0xA1A2A4A8
andi $t0, $t0, 0x5E5D5B57
ori  $t0, $t0, 0xFFFF0000
```

2. **(10 points):** Write the MIPS code for the following array access:
   B = A[i], where *B* is mapped to register $s0, the base address of the array *A* is in register $s1 and the variable *i* is in register $s2.

   a) **(5 points)** First assume that A is a byte array

   b) **(5 points)** Now assume A is a word array

3. **(5 points)** Name the 5 components of a computer

Name:_____

Problem 2: (25 points) Simple Arithmetic (20 minutes)

Translate the following code into MIPS assembly:

Z = A*B * (C+D) + E*F + G - 40

Assume that variables A-G, Z are in registers $1-$8, respectively (e.g. C = $3). Also, assume that you cannot overwrite variables A-G since they will all be used later in the program.

For this question, you can use the "mul" pseudo-instruction. It has this form:

```
mul $1, $2, $3, where $1 = $2 * $3.
```

It does the same thing as:

```
mult $2, $3
mflo $1
```

You should not use mult and mflo.

a) Assume that you have sequential processor where add and mul take 1 and 3 cycles, respectively. Write the code such that it uses the minimum number of registers. How many additional registers (other than $1-$8) does your code require? How many cycles does your code need?

Name:_____

b) Assume that you have a different processor (call it VLIW) that can perform one ADD operation and one MULTIPLY operation during every cycle. Rewrite the code to take advantage of this and use the minimum number of cycles. The VLIW processor uses a faster implementation of MULTIPY, which only takes 1 cycle, the same as the cycle time of ADD. To make it easier to grade, please write mul and add that are executed in the same cycle on one line, for example:

mul $1, $2, $3 add $4, $2, $8

How many cycles does your code need? How many additional registers (other than $1-$8) does your code require?

Name:_____

Problem 3: (25 points) Understanding MIPS Programs (20 minutes)

```
     li $t0, 0
     li $v0, 0
moral:
     mul $t1, $t0, 4
     add $t1, $a0, $t1
     lw $t1, 0($t1)
     blt $t1, $0, orel
     mul $t2, $v0, 4
     add $t2, $a1, $t2
     sw $t1, 0($t2)
     add $v0, $v0, 1
orel:
     add $t0, $t0, 1
     blt $t0, $a2, moral
```

a) (20 points): Translate the assembly code above into a high-level language like C or Java. You should include a header that lists the types of any variables. Also, your code should be as concise as possible, without explicit pointers. We will not deduct points for syntax errors unless they are significant enough to alter the meaning of your code. You are not allowed to use go to statements; go to statements are harmful.

b) (5 points): Describe briefly, in English, what this function does.

Name:_____

Problem 4: (30 points) Compilation (30 minutes)

The following lines of code perform a finite impulse response (FIR) filter using input from array *z* constant coefficients found in integer array *h*.

```
int h[];
int z[];
int input;
int ntaps;
int ii;
int accum;

/* store input at the beginning of the delay line */
z[0] = input;

/* calc FIR and shift data */
accum = h[ntaps - 1] * z[ntaps - 1];
for (ii = ntaps - 2; ii >= 0; ii--) {
    accum += h[ii] * z[ii];
    z[ii + 1] = z[ii];
}
```

Assume the following for the assignment of the variables:

Variable	Register
the base address of h	$a0
base address of z	$a1
input	$a2
ntaps	$a3
ii	$s0
accum	$s1

Write the MIPS assembly code that correctly executes the code. You can not overwrite the given variables unless the code explicitly states to do so.

Name:_____