

Name: _____ Email: _____

Problem 1 (18%). Assemble the following machine instructions into **binary**, use spaces to separate fields, and **registers** in their symbolic form (\$ra NOT \$31). Assume absolute jump addresses.

Field 1	Fields 2 and etc	instruction
		jal 0x400
		li \$t0, 0xdefa
		lhu \$at, 26(\$sp)
		b 0x400
		not \$a2
0x12740002		

Problem 2 (7%).

Assume each part is **independent**. Assume absolute jump & branch addresses (no pc relative).

Fill in only registers that changed!

What is the value of the register or memory contents **after** the execution of the instruction.

Assume pc = 2020; \$s3=12; \$s4=6; \$ra=250; memory[8]=0xfedcba98; memory[12]=0x76543210;

instruction	pc	\$ra	\$s3	\$s4	memory[8]	memory[12]
jal 400						
sll \$s4, \$s3, 2						
sh \$s4, 8(\$s4)						
move \$ra, \$s4						
slt \$s3, \$s4, \$ra						
bgt \$s3, \$s4, 40						
ori \$s3, \$s4, 0x0001						

Problem 3. (25%) Translate the following C code into MIPS. Please comment your code. Assume signed unless defined otherwise. **x** is \$s0; **y** is \$s1; **s** is \$s2; **t** is \$s3; **r** is \$s4; **p** is \$s5; **d** is \$s6.

No pseudo-assembler instructions allowed. Points will be taken off for assembler syntax errors.
register unsigned int x, y; register int s, **t; struct { int a[3]; short b; char c; } r, *p; char d;

(a) `x += (x - y) + (s - 5);`

(b) `**t = d;`

(c) `x = (x >= 3)? y*3 : 0x40 + 3;`

(d) `for(x=y; x < y+2; x+=8) { y *= 3; }`

(d) `r.c = r.b + r.a[x];`

(e) `p->c = p->b + p->a[x];`

Problem 4. (25%) Translate the following code and add **comments**

No pseudo-assembler instructions allowed. Points will be taken off for assembler syntax errors.

```
char *strcpy(char *s, char *t) {
    register int i;

    i=0;
    while((s[i] = t[i]) != '\0') i++; /* note: "=" NOT "= =" */

    return s;
}
```

(a) Write the prolog

(b) Write the body

(c) Write the epilog

Problem 5. (10%) Translate the following global variables and assign the location counter beginning at 4000

(a) `short x=0x1914; short *s=&x; short **p = &s;`

Location counter	Assembler definitions
4000	

(b) `struct keyword {
char **argv;
int (*daytab)[13];
int *(montab[13]);
void (*strcpy)();
struct keyword *next;
} fp;`

Location counter	Assembler definitions
4000	

Problem 6. (15%) Given the following instruction sequence in the table below.

Assume the (alu and slt instructions are 5 clocks); (loads 10 clocks); (stores 20 clocks); (jumps 2 clocks); (branches 4 fall through/8 for branch);

- How many different timing paths? _____
- Show the **best** case timing path through the code showing annotations and total.
- Show the **worst** case timing path through the code showing annotations and total.
- What values will make this code execute the worst case?

instruction	best case	worst case
<code>slti \$t0,\$a0,3</code>		
<code>slt \$t1,\$a0,\$t0</code>		
<code>beq \$t1,\$0,L1</code>		
<code>addi \$t2,\$zero,5</code>		
<code>L1: beq \$a2,\$t1,L2</code>		
<code>addi \$a1,\$a1,3</code>		
<code>L2: addi \$s1,\$zero,10</code>		
Total Time		