Problem P1 [10 points]: Assembly

Explain the notion of “mnemonic” and its role in assembly programming? Why do we need these?

Problem P2 [15, 10 points]: Assembly

(a) Translate the following code into MIPS assembly:

\[ A = B + C - D + 12 - E - F \]

Assume that variables A, B, C, D, E, F are in registers $s0, $s1, $s2, $s3, $s4, $s5, respectively. Also, assume that you cannot overwrite the variables B, C, D, E, F since they will all be used later in the program. Finally, assume that any ADD or SUBTRACT operation takes 1 cycle. Write the code such that it uses the minimal number of registers. How many additional registers (other than $s0-$s6) does your code require? How many cycles does your code need?

(b) Assume that the processor can perform two operations during every cycle where each operation can be either ADD or SUBTRACT. Rewrite the code to take advantage of this. How many cycles does your code need? How many additional registers (other than $s0-$s6) does your code require?

Problem P3 [25 points]: Assembly and Arrays

Translate the following code from Pseudo code into assembly


Assume that A and B are byte arrays. Both arrays are stored in memory. Register $s1 contains the initial address for array A; register $s2 contains the initial address for array B; and register $s3 contains the value of i.

Problem P4 [15 points]: Reverse Assembly

Translate the following assembly code into a high-level language of your choice like C or Java. For variable names, please use increasing letters, starting with A (ie. $s2 = C).

```
addi $s0, $0, 1
addi $s1, $0, 57
addi $s2, $0, 99
addi $s3, $0, 28
addi $s4, $0, 75
```
Problem P5 [25 points]: Reverse Assembly
Translate the following assembly code into a high-level language of your choice like C or Java. For variable names, please use increasing letters, starting with A (ie. $s2 = C).

Assume that B and C are byte arrays. Both arrays are stored in memory. Register $s1 contains the initial address for array B; register $s2 contains the initial address for array C; and register $s3 contains the value of i. The final result will reside in $s4

```
addi $s3, $0, 1
add $t0, $s2, $s3
lw $t1, 0($t0)
add $t2, $s1, $t1
lw $s4, 0($t2)
```

Problem P6 [10 points]: Assembly
For the following MIPS code, what will be the final value stored in $s0 after it has completed?

The initial values stored in $s0-$s7 are: 0, 23, 65, 78, 32, 12, 90, 55

```
addi $t0, $s1, 8
add $t1, $s3, $s5
sub $t1, $t1, $s6
add $s4, $t0, $t1
sub $t2, $t1, $t0
sub $s0, $s4, $t2
```