

Homework 1

Problem 1

Add comments to the following code and describe in one sentence what it computes. Assume that a0 is used for the input and initially contains n, a positive integer. Assume that a0 is used for the output.

Code for Problem 1:

```
begin:  addi t0, x0, 0      // t0 = sum = 0
        addi t1, x0, 1      //
loop:   slt t2, a0, t1      //
        bne t2, x0, finish  //
        add t0, t0, t1      //
        addi t1, t1, 2      //
        j loop              //
finish: add a0, t0, x0      //
```

Problem 2

Write a loop that reverses order of the bits of an 8-bit number in s0 and stores the result in s1. RISC-V registers have 4 bytes, but in this problem, the higher 24 bits of s0 are always 0. For example, if the lower 8 bits of s0 are 00001101, the lower 8 bites s1 should be 10110000.

Problem 3

Assume we have an array in memory that contains `int* arr = {1,2,3,4,5,6,0}`. Let the values of arr be a multiple of 4 and stored in register s0. What do the snippets of RISC-V code do? Assume that all the instructions are run one after the other in the same context.

- a) `lw t0, 8(s0)`
- b) `slli t1, t0, 2`
`add t2, s0, t1`
`lw t3, 0(t2)`
`addi t3, t3, 1`
`sw t3, 0(t2)`
- c) `lw t0, 16(s0)`
`xori t0, t0, 0xFFF`
`addi t0, t0, 1`

Problem 4

Write a function `sumSquare` in RISC-V that, when given an integer n, returns the summation below. If n is not positive, then the function returns 0.

$$n^2 + (n - 1)^2 + (n - 2)^2 + \dots + 1^2$$

For this problem, you are given a RISC-V function called `square` that takes in an integer and returns its square. Implement `sumSquare` using `square` as a subroutine.