**NYU Tandon School of Engineering**

**Fall 2021, ECE 6913**

**Homework Assignment 7**

*Instructor: Azeez Bhavnagarwala,* email: ajb20@nyu.edu

*Course Assistant Office Hour Schedule (Room 808, 370 Jay St: 9AM – 11AM)*

*Mondays & Tuesdays:* Haotian (Kenny) Zheng **hz2687@nyu.edu &** Shan Hao **sh6206@nyu.edu**,

*Wednesdays:* Karan Parikh **kap9580@nyu.edu**

*Thursday:* Sahil Chitnis **ssc9983@nyu.edu**

*Fridays:*Kewal Jani **kj2062@nyu.edu**

*Saturday:* Zhiming Fan **zf2035@nyu.edu**

[released Monday November 21st 2021] [due\* **Wednesday December 1st 2021, *before* 11:55 PM**]

You *are allowed* to discuss HW assignments only with other colleagues taking the class. You are *not allowed* to share your solutions with other colleagues in the class. Please feel free to reach out to the Instructor during office hours or by appointment if you need any help with the HW.

Please enter your responses in this Word document after you download it from NYU Classes. *Please use the NYU Classes portal to upload your completed HW. Please do not upload images of handwritten sheets or PDFs of scanned sheets of handwritten solutions. Please be sure to type-in your solutions into Word or Google Docs and upload machine readable documents only*

1. Please use the online 32-bit RISC V simulator: [**https://www.kvakil.me/venus/**](https://www.kvakil.me/venus/) or [**https://www.cs.cornell.edu/courses/cs3410/2019sp/riscv/interpreter/**](https://www.cs.cornell.edu/courses/cs3410/2019sp/riscv/interpreter/)
2. Please write the RISC V code, run it online to test/debug, demonstrate it works, include your code in the PDF you upload – as text not as an image
3. Your code is graded for (1) validity (it works) (2) size (fewer lines, higher grades) (3) discussion explaining choices you made and why
4. You cannot use/copy parts of or all of anyone else’s code

Please enter your responses in this Word document after you download it from NYU Classes. *Please use the NYU Classes portal to upload your completed HW.*

1. Write a RISC V program using instructions in the RISC V ISA to calculate the sum of the cubes of all odd numbers between -N and +N where N is an integer < 100

Assume, between the range of -100 to 100, the starting number N1 is stored in x21, and the ending number N2 is stored in x22.

Loop is used to iterate through all numbers from N1 to N2. x2 stores the final sum. x3 is a constant, initialized to 2. x4 is the remainder of x1 % 2. beq x4, x0, END2 is an if statement verifying if the remainder is not 0, in which case x1 will be then cubed (result stored in x5) and added to the final sum.

addi x3, x0, 2

LOOP: bgt x21, x22, END1

rem x4, x21, x3

beq x4, x0, END2

mul x5, x21, x21

mul x5, x5, x21

add x2, x2, x5

END2: addi x21, x21, 1

jal x0, LOOP

END1:

1. Write a RISC V program using instructions in the RISC V ISA to calculate the factorial of any positive integer N < 100

Assume the starting number N is stored in x21. Loop through the numbers in range and multiply them together, the result of which is stored in x2.

addi x1, x0, 1

addi x2, x0, 1

LOOP: bgt x1, x21, END

Mul x2, x2, x1

addi x1, x1, 1

jal x0, LOOP

END:

1. Write a RISC V program using instructions in the RISC V ISA to calculate the sum of all prime numbers less than a given integer N where N < 100

Assume the starting number N is stored in x21. Loop through the numbers in range and multiply them together, the result of which is stored in x2.

Loop1 loops through all numbers in range. Loop2 loops from 2 to (currently selected number from Loop1 divided by 2) in order to test how many divisors there are other than 1 and the currently selected number itself. END3 is an if statement where if the remainder of the test is 0, number of divisors x6 + 1, and breaks out of the loop (beq x5, x0, END2) and goes to END2, and then goes to END4 because x6 (number of divisors) is greater than 0, and then moves on to test the next number. If the said if statement earlier returns false (remainder != 0), x3 (initially 2) is added 1 so that next remainder can be tested, unless x3 already reaches x4 (midway between 2 and the currently selected number). END4 is an if statement where if x6 (number of divisors) is 0, the number then gets added to the final sum x2.

beq x5, x0, END2 is a break line (Break if divisor is found. This line makes the overall code longer, but reduces overall iterations looped)

addi x1, x0, 2

LOOP1: bgt x1, x21, END1

addi x3, x0, 2

srai x4, x1, 1

add x6, x0, x0

LOOP2: bgt x3, x4, END2

rem x5, x1, x3

bne x5, x0, END3

addi x6, x6, 1

beq x5, x0, END2

END3: addi x3, x3, 1

jal x0, LOOP2

END2: bne x6, x0, END4

add x2, x2, x1

END4: addi x1, x1, 1

jal x0, LOOP1

END1:

1. Write a RISC V program that calculates the sum of N terms in a geometric series where a = 1 and r = -3

Assume the starting number N is stored in x21, a stored in x22, and r stored in x23.

The formula = a \* (Math.pow(r, 5) - 1) / (r - 1); The final result is stored in x2.

addi x1, x0, 2

add x2, x23, x0

LOOP: bgt x1, x21, END

mul x2, x2, x23

addi x1, x1, 1

jal x0, LOOP

END: addi x2, x2, -1

addi x3, x23, -1

div x2, x2, x3

mul x3, x3, x22