
NYU Tandon School of Engineering

Spring 2021, ECE 6913

Homework Assignment 8

Instructor Contact: ajb20@nyu.edu

Instructor Office hours: Wednesday: 9AM - 10:30 AM or by appointment

Course Assistant: **Archit** ax368@nyu.edu; **Deep** dss569@nyu.edu

Course Assistant Office hours: Monday, Tuesday, Thursday & Friday 9 AM - 10:30 AM

[released Friday April 30th 2021] [due* **Saturday May 8th 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.

Instructions:

1. Please use the online 32-bit RISC V simulator: <https://www.kvakil.me/venus/> or <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
- Step 1: check if N is odd number. If not find next odd no less than N.
 - Step 2: find negative of odd numbers. compute a cube of odd numbers add them.

C program:

```
Int N; //input value

Int sum =0;

Int cube = 1;

if(N%2 == 0) //check if number is even
```

```
{N = N-1;}    //N-1 is odd number
```

```
While(N>=1){
```

```
    Sum = sum + cal_pow(N,3);
```

```
    N--;
```

```
}
```

```
int cal_pow(int in, int pow){
```

```
    cubeN =1;
```

```
    for(;pow>0;pow--){
```

```
        cubeN = cubeN * in;
```

```
    }
```

```
    Return cubeN;
```

```
}
```

Assembly Program: (optimized)

```
li t0, 0xFF    // input number N
```

```
li t2, 2        //to check for divisibility
```

```
li t4, 1
```

```
add t3,x0,x0    //sum = 0
```

```
andi t1,t0,0x1  //AND immediate with lowest bit to check if no is odd
```

```
beq t1,t4,Label1
```

```
addi t0,t0,-1    // N = N-1
```

Label1:

```
beq t0,x0,Label2
```

```
mul t5,t0,t0
```

```
mul t5,t5,t0
```

```
sub t6,x0,t5    // -(N)^3 = (-N)^3
```

```
add t3,t5,t6    //add cubes of N and -N
```

```
addi t0,-2      //N-2 for next odd number
```

Label2:

(2) Write a RISC V program using instructions in the RISC V ISA to calculate the factorial of any positive integer $N < 100$

- $13! = 6,227,020,800 > 32\text{-bit unsigned maximum value } (4,294,967,295)$
- If input value N is greater than or equal to 13 output of factorial overflows.
- If input is 0 $\rightarrow 0! = 1$

C program:

```
int n; //input number

int out=1;

if(n!= 0){
while(n>=1)
{
    out = out*n
    n = n-1;
}
}
else{
out = 1;
}
```

Assembly Program:

```
li t0,0x05 //load value N here N = 5 for example
addi t3,0x1
add t1,x0,1 //initialize t1 register to value 1
beq t0,x0,Label
Loop:
mul t1,t1,t0 // out = out*n
addi t0,t0,-1 // n = n-1
bge t0,t3, Loop
Label:
addi t1,x0,1 //set out = 1 for N =0
```

- (3) 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 $N < 100$

-

C program:

```
Int N; // input number

int sum = 0;

int i =2;

int M =2;

while(M<=N) {

    while(i <= M)

    {

        if(M%i == 0)

            i++;

    }

    if(M != i) {

        sum = sum + M;

        M = M+1;

    }
```

Assembly program:

```
//int N input number is in reg t0
Li t1,2 // i=2
Li t2,2 // M=2
Add t4,x0,x0 //sum =0
Loop1:
Bgt t0,t2, Label1
Loop2:
```

```

Bgt t1,t2, Label2
Rem t3,t2,t1
Beq t3,x0, Label2 //break if number is divisible
Addi t1,t1,1
J Loop2
Label2:
Beq t1,t2,Label1
Add t4,t4,t2
Addi t2,1
J Loop1
Label 1:

```

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

Sequence: $a \cdot r^n$ summation of N terms

C program:

```

int n = 5;    //no of terms
int a = 1;
int r = -3;
int result = 1;
for(int i = 0; i<n; i++)
{
    result = result + a* cal_pow(r,n);
}

int cal_pow(int in, int pow){
    cubeN =1;

    for(;pow>0;pow--){

        cubeN = cubeN * in;

    }

    Return cubeN;
}

```

```
}
```

Assembly program:

```
//load immediate pseudo-instruction
li t0,0x5 //n =5 no. of terms
li t1,0x1 // a =1
li t2,-3 // r =-3
addi t3,x0,1 //set sum = 1
add t4,x0,1 //initialize sum of power of r with 1
addi t0,t0,-1 //decrement iterator because value  $r^0$  already in reg
Loop:
beq t0,x0, Label
mul t4,t4,t2 //calculate  $r^n$ 
add t3,t3,t4 //summation( $r^n$ )
addi t0,t0,-1 //loop iterator decrement
j Loop
Label:
Mul t1,t1,t3 //optional can omit
```