Orhi takes register r0, does the OR operand with %hi(X) (high part of x) and stores the result in register 2

Ori takes register 2, does the OR operand with %lo(x) (low part of x) and stores it into the same place

Movia takes the address of Y and stores that address to register 3

Movia takes the address of N and stores that address to register r4

Ldw takes the value of r4 as a pointer to memory and adds the offset 0 to it. It then loads a word (4 bytes) of data from that address in memory and stores that data in r4.

Add adds register 0 with itself and stores it into r5. Basically a times two operand to r0.

Label???

Ldw takes the value of r2 as a pointer to memory and adds the offset 0 to it. It then loads a word (4 bytes) of data from that address in memory and stores that data in r6.

**2.2**

0x993FF915 = 10011 00100 1111111111100100 010101

010101 = 0x15 = stw

I-type

Ra = 0x13 or Decimal 19

Rb = 0x04

IMM16 = 1111111111100100 = 0xFFE4

**stw r4, OFFSET>??(r19)**

**2.3**

divu r14,r5,r22

is R type due to 3 register bits = 0x3A = 111010

divu rC, rA, rB

OPX = divu = 0x24 = 100100

rc = 14 = 01110

ra = 5 = 00101

rb = 22 = 10110

**00101 10110 01110 00000100100 111010**