Note : ALL NUMBERS ARE UNSIGNED.

add R1 R2 R3
00000_**00**_001_010_011
// 2 unused bits

sub R1 R2 R3
00001_**00**_001_010_011
// 2 unused bits

mov R1 $Imm (mov R1 $12) //// 12 is a decimal value
00010_**0**_001_0001100
// 1 unused bit

mov R1 R2
00011_**00000**_001_010
// 5 unused bits

ld R1 var_name2 //// while creation of binary code var_name2 will be replaced by 7 bit binary address
00100_**0**_001_0001100
// 1 unused bits

st R2 var_name1 //// while creation of binary code var_name1 will be replaced by 7 bit binary address
00101_**0**_010_0001111
// 1 unused bits

mul R1 R2 R3
00110_**00**_001_010_011
// 2 unused bits

div R3 R4
00111_**00000**_011_100
// 5 unused bits

rs R1 $Imm (rs reg1 $12) // 12 is in decimal
01000_**0**_001_0001100
// 1 unused bit

ls R1 $Imm (ls reg1 $13) // 13 is in decimal
01001_**0**_001_0001101
// 1 unused bit

xor R1 R2 R3
01010_**00**_001_010_011
// 2 unused bits

or R1 R2 R3
01011_**00**_001_010_011
// 2 unused bits

and R1 R2 R3
01100_**00**_001_010_011
// 2 unused bits

not R1 R2
01101_**00000**_001_010
// 5 unused bits

cmp R1 R2
01110_**00000**_001_010
// 5 unused bits

jmp label1 //// while creation of binary code label1 will be replaced by 7 bit binary address
01111_**0000**_0001101
// 4 unused bits

jlt label2 //// while creation of binary code label2 will be replaced by 7 bit binary address
11100_**0000**_0001000
// 4 unused bits

jgt label3 //// while creation of binary code label3 will be replaced by 7 bit binary address
11101_**0000**_0001100
// 4 unused bits

je label4 //// while creation of binary code label4 will be replaced by 7 bit binary address
11111_**0000**_0001110
// 4 unused bits

hlt
11010_**00000000000**
// 11 unused bits

addf R1 R2 R3
10000_**00**_001_010_011
// 2 unused bits

subf R1 R2 R3
10001_**00**_001_010_011
// 2 unused bits

//// our floating point representation have no sign bit , 3 exponent bits, 5 mantissa bits ////
//// exponent_in_float = exponent_in_binary + bias
////(1.5)10 = (1.1)2 = 1.1 X 2^0
////1.1 X 2^0 in binary = (0 + 3) in exponent 10000 in mantissa
////representation = 111_10000
/////-------------- bias formula for floating point = bias_in_binary = 2^(k-1) -1 where k: no of bits in exponent

movf reg1 $1.5   //// 1.5 is in decimal representation
10010_001_111_10000
// 0 unused bit

addr_label: add R1 R2 R3   //// here when the addr_label will be used then it will be replaced
                                                    //// by the 7-bit address (where this add
instruction is stored)
addr_label2: hlt

----------------- one sample assembly program---------

```
var var1
var var2
var var3
ld R1 var1
ld R2 var2
st R3 var3
jmp hlt_label
add R1 R2 R3
hlt_label: hlt
```

------ memory content explanation---- //// remember we are following von-numen architecture so only
//// single memory both for instruction and data

addr instruction_code
0000000: 00100_**0**_001_0000110
0000001: 00100_**0**_010_0000111
0000010: 00101_**0**_011_0001000
0000011: 01111_0000_0000101
0000100: 00000_**00**_001_010_011
0000101: 11010_00000000000

---- locations allocated to variables in the order of their declaration (var1,var2,var3)
0000110:
0000111:
0001000:
------ memory content explanation ends ----

------ format of generated binary code ----

```
0010000010000110
0010000100000111
0010100110001000
0111100000000101
0000000001010011
1101000000000000
```

------ format of generated binary code ends ----

----------------- sample assembly program ends-----------------
----

------- corected program from the shared project pdf -------------
------ assembly program----
```
var X
mov R1 $10
mov R2 $100
mul R3 R2 R1
st R3 X
hlt
```
------- assembly program ends -----

------- explanation of binary from the assembler--
0000000: 00010_**0**_001_0001010
0000001: 00010_**0**_010_1100100

0000010: 00110_**00**_011_010_001
0000011: 00101_**0**_011_0000101
0000100: 11010_**00000000000**
------- explanation of binary from the assembler ends--

---------- binary genrated form assembler------
0001000010001010
0001000101100100
0011000011010001
0010100110000101
1101000000000000
----------- binary genrated form assembler ends---
------- corected program from the shared project pdf ends -------------
----