Store:

1 bit for sign

15 bits for exponent

112 bits for significand (Not save the first set bit)

ATTENTION:

To be same with pdf case ,we output a long_double with all bits reseted as 0 X 2^-16382 rather than 0 X 2^16383

Work:

1.Implement three functions.

2.Add many functions to .

3.Add two functions to debug

1.Implement three functions.

1.1 FP_mul

a.We get the signs of op1 and op2,set result's sign.

b.We get the exponent bits of op1 and op2, we set result's exponent.

c.We create a characters array of size 225(Hold all the bit of the result),and may add one to exponent of result. And get the significand.

1.2 long_double_print_bitseq

Just output the long_double according to its bit number using the functions added.

1.3 long_double_print_normalized

Just output the long_double and process the format meanwhile.

2.Added Functions

2.1 GetNthBit

Get the Nth bit of pointer and return it.

2.2 SetNthBit

Set the Nth bit of pointer to 1

2.3 ResetNthBit

Set the Nth bit of pointer to 0

2.4 GetNValue

Get N Value as long

2.5 GetSign

Get the sign of a long_double pointer

2.6 BiggerM

Determine if M1 is Bigger than M2

2.7 MoveN

Move the char to left by N

2.8 SetN

Set exponential value

2.9 SetM

Set tail value of long_double

2.10 Minis

Get the abs number of M1 – M2 in R

3.Add two functions to debug

Two functions that can output the double in binary sequences and normalized format.