Date

Aitzaz Tahir Ch
19p0012

BS(CS)-B
Numerical Computing

Task 1:

Real Number = 100.875

Step 1

Converting Decimal Binary (100)

2	100
2	50 – 0
2	25 – 0
2	12 – 1
2	6 – 0
2	3 – 0
	1 – 1

$(100)_{10} \rightarrow (1100100)_2$

(0.875)

$0.875 \times 2 = 1.75$
$0.75 \times 2 = 1.5$
$0.5 \times 2 = 1$

$(0.875)_{10} = (.111)$

$(100.875)_{10} = (1100100.111)_2$

Date

Step 2

Finding exponent

1100100.111

Exponent = 6

Biased exponent = 6 + 127

$= (133)_{10}$

2	133
2	66 – 1
2	33 – 0
2	16 – 1
2	8 – 0
2	4 – 0
2	2 – 0
	1 – 0

$(133)_{10} = (10000101)_2$

Step 3 Finding Mantissa

Mantissa = 10010011100000000000000

Final

0	10000101	10010011100000000000000

In [4]:

```
# Python program to convert a real value to IEEE 754 standardized format.

# Function to convert a fraction to binary form.
def binaryOfFraction(fraction):
#Declaring an empty string to store binary bits.
        binary = str()
        # Iterating through fraction until it becomes Zero.
        while (fraction):
                #Multiplying fraction by 2.
                fraction *= 2
                # Storing Integer Part of fraction in int_part.
                if (fraction >= 1):
                        int_part = 1
                        fraction -= 1
                else:
                        int_part = 0
                # Adding int_part to binary after every iteration.
                binary += str(int_part)
        #Returning
        return binary
```

In [5]:

```
# Function to get sign bit, exp bits and mantissa bits from given real no.
def floatingPoint(real_no):
        sign_bit = 0
        # Sign bit will set to 1 for negative no.
        if(real_no < 0):
                sign_bit = 1
        #Converting given no. to absolute value as we have already set the sign bit.
        real_no = abs(real_no)
        #Converting Integer Part of Real no to Binary
        int_str = bin(int(real_no))[2 : ]
        #Function call to convert fraction part of real no to Binary.
        fraction_str = binaryOfFraction(real_no - int(real_no))
        #Getting the index where bit was high for the first time in binary representation of Integer part of real no.
        ind = int_str.index('1')
        # The Exponent is the no. by which we have right shifted the decimal and it is given below.
        # Also adding 127.
        exp_str = bin((len(int_str) - ind - 1) + 127)[2 : ]
        # getting mantissa
        mant_str = int_str[ind + 1 : ] + fraction_str

        #Adding Zeroes in LSB of mantissa string so as to make it's length of 23 bits.
        mant_str = mant_str + ('0' * (23 - len(mant_str)))

        #Returning the sign, Exp and Mantissa Bit strings.
        return sign_bit, exp_str, mant_str
```

In [6]:

```
#Driver Code
if __name__ == "__main__":

        #Function call to get Sign, Exponent and Mantissa Bit Strings.
        sign_bit, exp_str, mant_str = floatingPoint(100.875)

        #Final IEEE-754 standardized format representation.
        ieee_32 = str(sign_bit) + '|' + exp_str + '|' + mant_str

        # Printing the ieee 32 representation.
        print("IEEE 754 representation of 100.875 is :")
        print(ieee_32)
```

```
IEEE 754 representation of 100.875 is :
0|10000101|10010011100000000000000
```

In [ ]: