class Binaryconversion():

def \_\_init\_\_(self):

self.BitStream = None

self.Mantissa = None

self.Exponent = None

self.RealNumber = None

def InputBitStream(self):

ValidPattern = False

self.BitStream = input("Enter a 12 bit word: ")

while not ValidPattern:

while len(self.BitStream) != 12:

self.BitStream = input("That was not 12 bits. Enter again: ")

ValidPattern = True

for string in self.BitStream:

if string != "0" and string != "1":

ValidPattern = False

if not ValidPattern:

self.BitStream = input("That was not a binary word. Enter again. ")

def ExtractMantissa(self):

self.Mantissa = 0

for counter in range(1,8):

self.Mantissa = self.Mantissa + int(self.BitStream[counter]) \* 2\*\*(-counter)

def ExtractExponent(self):

pass

# Add code here

def ApplyExponent(self):

pass

# Add code here

def OutputRealNumber(self):

print("Exponent: ",self.Exponent)

print("Mantissa: ",self.Mantissa)

print("Real Number: ",self.RealNumber)

B = Binaryconversion()

B.InputBitStream()

B.ExtractExponent()

B.ExtractMantissa()

B.ApplyExponent()

B.OutputRealNumber()