**Q. Write a program to convert Decimal to Fraction.**

Input: A string which contains the number. If the decimal is recurring you write the first set of numbers after the decimal and an 'r' before that. For Example: 1.77777... is written as "1.r7" if no repeating it is written normally.

Output: A string which shows the fraction form of the decimal? Bonus point if the fraction is of lowest form.

Constraints:

Don't use: \*as\_integer\_ratio() function

\*Fraction Library

Test Case:

"1.3r"

The fraction is 4/3

"1.3"

The fraction is 13/10

**Python Program**

n = input()

def simplification(num, denom):  
 while denom % 1 == 0:  
 if num % 2 == 0 and denom % 2 == 0:  
 num /= 2  
 denom /= 2  
 elif num % 3 == 0 and denom % 3 == 0:  
 num /= 3  
 denom /= 3  
 elif num % 5 == 0 and denom % 5 == 0:  
 num /= 5  
 denom /= 5  
 else:  
 break  
 return num, denom  
  
  
if **'r'** in n:  
 pos1 = n.find(**"."**)  
 pos2 = n.find(**"r"**)  
 Denominator1 = 1  
 Denominator2 = 9  
 Numerator1 = float(n[:pos2])  
 Numerator2 = float(n[(pos2+1):])  
 for counter in range(len(n)-(pos2+2)):  
 Denominator2 = (Denominator2\*10) + 9  
 while Numerator1 % 1 != 0:  
 Numerator1 \*= 10  
 Denominator1 \*= 10  
 Denominator2 \*= 10  
 Numerator = (Numerator1\*Denominator2) + (Numerator2\*Denominator1)  
 Denominator = Denominator1\*Denominator2  
else:  
 Numerator = float(n)  
 Denominator = 1  
 while Numerator % 1 != 0:  
 Numerator \*= 10  
 Denominator \*= 10  
  
Numerator, Denominator = simplification(Numerator, Denominator)  
print(**"{}/{}"**.format(int(Numerator), int(Denominator)))