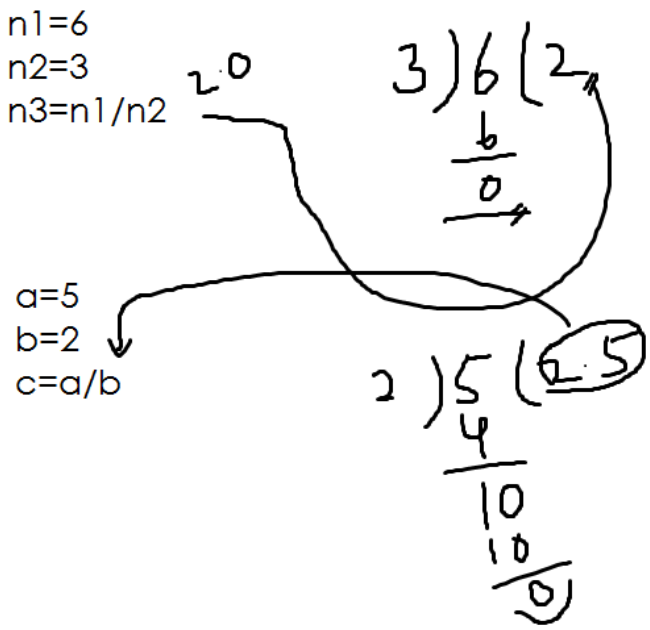
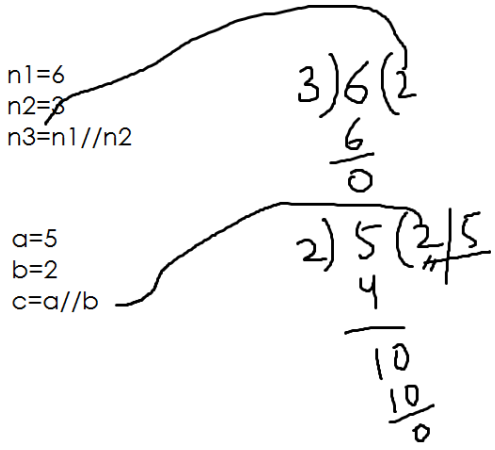
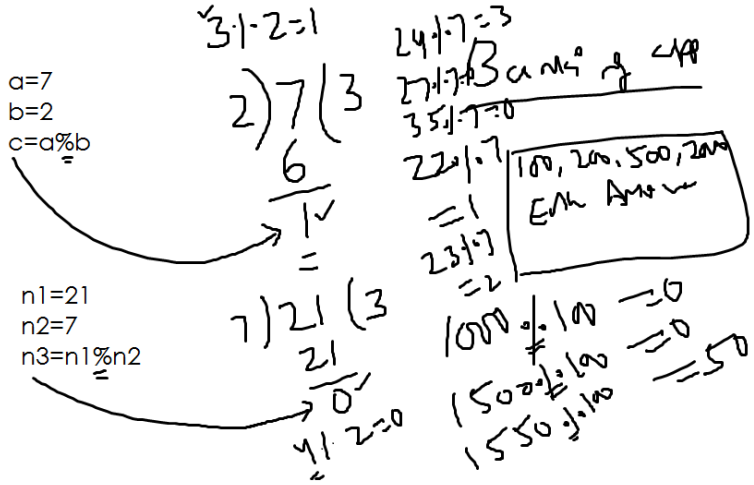
/	Float division
	This is binary operator it required 2 operands
	This operator divide two numbers and return result in float type.
	
	 
	
	Example:
	>>> n1=6
	>>> n2=3
	>>> n3=n1/n2
	>>> print(n1,n2,n3)
	6 3 2.0
	>>> a=5
	>>> b=2
	>>> c=a/b
	>>> print(a,b,c)
	5 2 2.5
	
	Example:
	# write a program to find simple interest
	# si=ptr/100
	
	p=float(input("Enter Amount"))
	t=int(input("Enter Time"))
	r=float(input("Enter Rate"))
	si=p*t*r/100

	print("Simple Interest is ",si)
	
	Output:
	Enter Amount5000
	Enter Time12
	Enter Rate1.2
	Simple Interest is 720.0
	
	Example:
	>>> x=5
	>>> y=0
	>>> z=x/y
	Traceback (most recent call last):
	  File "<pyshell#10>", line 1, in <module>
	  z=x/y
	ZeroDivisionError: division by zero
	>>>
	
	Note: cannot divide number with zero.

//	Floor division
	This division is also called integer division. This operator divide two numbers and get result integer
	
	 
	
	>>> n1=6
	>>> n2=3
	>>> n3=n1//n2
	>>> print(n1,n2,n3)
	6 3 2
	>>> a=5

	>>> b=2
	>>> c=a//b
	>>> print(a,b,c)
	5 2 2
	>>> x=5.0
	>>> y=2
	>>> z=x/y
	>>> print(x,y,z)
	5.0 2 2.5
	>>> z=x//y
	>>> print(x,y,z)
	5.0 2 2.0

%	Modular or Modulo
	This operator divide two numbers and return remainder
	 
	
	>>> a=23
	>>> b=7
	>>> c=a%b
	>>> print(a,b,c)
	23 7 2
	>>> n1=21
	>>> n2=7
	>>> n3=n1%n2
	>>> print(n1,n2,n3)
	21 7 0

**	Exponent Operator or Power of It is binary operator, which return power of number

	>>> a=5
	>>> b=2
	>>> c=a**b
	>>> print(a,b,c)
	5 2 25
	>>> x=5
	>>> y=-1
	>>> z=x**y
	>>> print(x,y,z)
	5 -1 0.2

Operator precedence

The following table summarizes the operator precedence in Python, from highest precedence (most binding) to lowest precedence (least binding). Operators in the same box have the same precedence. Unless the syntax is explicitly given, operators are binary. Operators in the same box group left to right (except for exponentiation, which groups from right to left).

Operator	Description
(expressions...), [expressions...], {key: value...}, {expressions...}	Binding or parenthesized expression, list display, dictionary display, set display
x[index], x[index:index], x(arguments...), x.attribute	Subscription, slicing, call, attribute reference
await x	Await expression
**	Exponentiation
+x, -x, ~x	Positive, negative, bitwise NOT
*, @, /, //, %	Multiplication, matrix multiplication, division, floor division, remainder
+, -	Addition and subtraction
<<, >>	Shifts
&	Bitwise AND
^	Bitwise XOR
|	Bitwise OR
in, not in, is, is not, <, <=, >, >=, !=, ==	Comparisons, including membership tests and identity tests

not x	Boolean NOT
and	Boolean AND
or	Boolean OR
if – else	Conditional expression
lambda	Lambda expression
:=	Assignment expression

Find output

```
expr1=2**1**8
print(expr1)
expr2=1+2+3+4
print(expr2)
expr3=1+2-3+4
print(expr3)
expr4=3*4/2
print(expr4)
expr5=2/3*4
print(round(expr5,2))
```

Output:

```
2
10
4
6.0
2.67
```

https://www.sanfoundry.com/python-questions-answers-precedence-associativity-1/