$x+$
$2y =$
3
$2x+$
$1y$
x, y

```
def some_class(x, y)
  return x + y == 3
```

```
def some_computation(x, y)
  return 2 / x + 1 / y
```

```
def find_some_value()
  all_x, all_y = all_real_numbers()
  all_satisfied_instances = []
  for x in all_x and for y in all_y
    if (x + y == 3)
      all_satisfied_instances.append( 2 / x + 1 / y )
  return find_maximum_value_of(all_satisfied_instances)
```

find_some_value

```
def find_some_value()
  all_x, all_y = all_real_numbers()
  all_satisfied_instances = []
  for x in all_x and for y in all_y
    # ======== CHANGED ======== #
    if (1 / 3 * (x + y) == 1)
    # ======== CHANGED ======== #
      all_satisfied_instances.append( 2 / x + 1 / y )
  return find_maximum_value_of(all_satisfied_instances)
```

```
def find_some_value()
  all_x, all_y = all_real_numbers()
  all_satisfied_instances = []
  for x in all_x and for y in all_y
    if (1 / 3 * (x + y) == 1)
      # ======== CHANGED ======== #
      all_satisfied_instances.append( ( 2 / x + 1 / y ) * 1 )
      # ======== CHANGED ======== #
  return find_maximum_value_of(all_satisfied_instances)
```

```
def find_some_value()
  all_x, all_y = all_real_numbers()
  all_satisfied_instances = []
  for x in all_x and for y in all_y
    if (1 / 3 * (x + y) == 1)
      # ======== CHANGED ======== #
      all_satisfied_instances.append( ( 2 / x + 1 / y ) * 1 / 3 * (x + y) )
      # ======== CHANGED ======== #
  return find_maximum_value_of(all_satisfied_instances)
```

```
def find_some_value()
  all_x, all_y = all_real_numbers()
  all_satisfied_instances = []
  for x in all_x and for y in all_y
    if (1 / 3 * (x + y) == 1)
      # ======== CHANGED ======== #
      all_satisfied_instances.append( 1 + 1 / 3 * ( 2 * y / x + x / y ))
      # ======== CHANGED ======== #
  return find_maximum_value_of(all_satisfied_instances)
```

```
def find_some_value()
  all_x, all_y = all_real_numbers()
  all_satisfied_instances = []
  for x in all_x and for y in all_y
    if (1 / 3 * (x + y) == 1)
      # ======== CHANGED ======== #
      all_satisfied_instances.append( 2 * y / x + x / y )
  return 1 + 1 / 3 * find_maximum_value_of(all_satisfied_instances)
  # ======== CHANGED ======== #
```

```
def find_some_value()
  all_x, all_y = all_real_numbers()
  all_satisfied_instances = []
  for x in all_x and for y in all_y
    # ======== CHANGED ======== #
    # ======== CHANGED ======== #
    all_satisfied_instances.append( 2 * y / x + x / y )
  return 1 + 1 / 3 * find_maximum_value_of(all_satisfied_instances)
```

```
  def find_some_value()
    maimum_of_known_techniques = find_maximum_value_by_x_y('2 * y / x + x / y')
    return 1 + 1 / 3 * maimum_of_known_techniques
```

$\overrightarrow{AC}-$
$\overrightarrow{BC}$