In [ ]:
```
Python for data analysis Lab 2
Nurshanov Dias
IT3-2208
```

In [17]:
```
def multiply_tuples(tup1: tuple, tup2: tuple):
    if len(tup1) < 2 or len(tup2) < 2:
        print('Both tuples must contain 2 elements.')
        return None
    numerator = tup1[0] * tup2[0]
    denominator = tup1[1] * tup2[1]
    return f'{numerator}/{denominator}'


def divide_tuples(tup1: tuple, tup2: tuple):
    if len(tup1) < 2 or len(tup2) < 2:
        print('Both tuples must contain 2 elements.')
        return None
    numerator = tup1[0] * tup2[1]
    denominator = tup1[1] * tup2[0]
    return f'{numerator}/{denominator}'

def get_smallest_fraction(fractions):
    return min(fractions, key=lambda frac: frac[0] / frac[1])
```

In [18]:
```
#Exercise 1.1)

fraction1 = input('Enter the first fraction like (1/3, 52/1, and so on): ')
fraction2 = input('Enter the second fraction like (1/3, 52/1, and so on): ')

fraction1 = tuple(map(int, fraction1.split('/')))
fraction2 = tuple(map(int, fraction2.split('/')))

#Ex 1.1
print('1.1 Exercise')
print(f'Multiplication of the fractions: {multiply_tuples(fraction1, fraction2)}')
```

```
1.1 Exercise
Multiplication of the fractions: 2/30
```

In [19]:
```
#Exercise 1.2)
print('1.2 Exercise')
print(f'Division of the fractions: {divide_tuples(fraction1, fraction2)}')
```

```
1.2 Exercise
Division of the fractions: 10/6
```

In [20]:
```
#Exercise 1.3)
print('1.3 Exercise')
fractions = []
while True:
    fraction_input = input("Enter a fraction >>> ")

    if fraction_input.lower() == "stop":
        break

    try:
        num, den = map(int, fraction_input.split('/'))
        fractions.append((num, den))
    except ValueError:
        print("Please enter a valid fraction in the format numerator/denominator")

if fractions:
    smallest_fraction = get_smallest_fraction(fractions)
    print(f"Smallest fraction: {smallest_fraction[0]}/{smallest_fraction[1]}")
else:
    print("No fractions were entered.")
```

```
1.3 Exercise
Smallest fraction: 1/5
```

In [22]:
```
import numpy as np


nums_all = []
print('2 Exercise')
while True:
    value = input("Input a number >>> ")

    if value.lower() == "stop":
        break

    try:
```

```
        num = int(value)
        nums_all.append(num)
    except ValueError:
        print("Please enter a valid integer or 'stop' to finish.")

nums_all_np = np.array(nums_all)
nums_even_np = nums_all_np[nums_all_np % 2 == 0]
nums_odd_np = nums_all_np[nums_all_np % 2 != 0]

sum_all = np.sum(nums_all_np)
sum_even = np.sum(nums_even_np)
sum_odd = np.sum(nums_odd_np)

average_all = np.mean(nums_all_np) if nums_all_np.size > 0 else 0
average_even = np.mean(nums_even_np) if nums_even_np.size > 0 else 0
average_odd = np.mean(nums_odd_np) if nums_odd_np.size > 0 else 0

print(f"All numbers: {nums_all_np.tolist()}")
print(f"Average of all numbers: {average_all}")
print(f"Sum of all numbers: {sum_all}",end='\n\n')

print(f"Even numbers: {nums_even_np.tolist()}")
print(f"Average of even numbers: {average_even}")
print(f"Sum of even numbers: {sum_even}",end='\n\n')

print(f"Odd numbers: {nums_odd_np.tolist()}")
print(f"Average of odd numbers: {average_odd}")
print(f"Sum of odd numbers: {sum_odd}",end='\n\n')
```

```
2 Exercise
Please enter a valid integer or 'stop' to finish.
All numbers: [1, 2, 3, -100]
Average of all numbers: -23.5
Sum of all numbers: -94

Even numbers: [2, -100]
Average of even numbers: -49.0
Sum of even numbers: -98

Odd numbers: [1, 3]
Average of odd numbers: 2.0
Sum of odd numbers: 4
```

In [23]:

```
import bisect

print("3 Exercise")

sorted_list = []

while True:
    value = input("Input a number >>> ")

    if value.lower() == "stop":
        break

    try:
        num = float(value)
        # Use bisect.insort to insert the number in the correct sorted position
        bisect.insort(sorted_list, num)
    except ValueError:
        print("Please enter a valid number or 'stop' to finish.")
print()
print("Sorted List:", sorted_list)
```

```
3 Exercise
Sorted List: [-99.999999, -1.0, 1.0, 1.2323, 1090.0]
```

In [24]:

```
list = [12, -4, 7, 0, 23, -5, 3, 9, -15, 4]

print('Exercise 3 option 1 task 1')
positive_items_count = len([i for i in list if i > 0])
print(f'Number of positive items in list: {positive_items_count}')

print()

print('Exercise 3 option 1 task 2')
largest_item = max(my_list)
largest_item_index = my_list.index(largest_item)
print(f'The largest item in the list is: {largest_item}')
print(f'The index of the largest item is: {largest_item_index}')
print()

print('Exercise 3 option 1 task 3')
```

```
odd_list = [i for i in my_list if i % 2 != 0]
print('The smallest odd item in the list is: ', end='')
if odd_list:
    print(min(odd_list))
else:
    print(0)
```

```
Exercise 3 option 1 task 1
Number of positive items in list: 6

Exercise 3 option 1 task 2
The largest item in the list is: 23
The index of the largest item is: 4

Exercise 3 option 1 task 3
The smallest odd item in the list is: -15
```

In [25]:
```
list = [12, -4, 7, 0, 23, -5, 3, 9, -15, 4]

print('Exercise 3 option 1 task 1')
positive_items_count = len([i for i in list if i > 0])
print(f'Number of positive items in list: {positive_items_count}')

print()

print('Exercise 3 option 1 task 2')
largest_item = max(list)
largest_item_index = list.index(largest_item)
print(f'The largest item in the list is: {largest_item}')
print(f'The index of the largest item is: {largest_item_index}')
print()

print('Exercise 3 option 1 task 3')
odd_list = [i for i in list if i % 2 != 0]
print('The smallest odd item in the list is: ', end='')
if odd_list:
    print(min(odd_list))
else:
    print(0)
```

```
Exercise 3 option 1 task 1
Number of positive items in list: 6

Exercise 3 option 1 task 2
The largest item in the list is: 23
The index of the largest item is: 4

Exercise 3 option 1 task 3
The smallest odd item in the list is: -15
```

In [26]:
```
list = [3, 4, 1, 7, 5, 6, -2, 9, 10, -8, 0, 2]

print('Exercise 3 option 2 task 1')
ge_prev_item_list = []
for i in range(1, len(list)):
    if list[i] > list[i - 1]:
        ge_prev_item_list.append(list[i])
print(f'List items that are larger than the previous item: {ge_prev_item_list}')

print()
# Exercise 3 option 2 task 2
print('Exercise 3 option 2 task 2')
ge_neighbors_item_list = []
for i in range(1, len(list) - 1):
    if list[i] > list[i - 1] and list[i] > list[i + 1]:
        ge_neighbors_item_list.append(list[i])
print(f'Number of elements in the list that are greater than both of their neighbors: {len(ge_neighbors_item_li
print(f'These elements are: {ge_neighbors_item_list}')

print()
# Exercise 3 option 2 task 3
print('Exercise 3 option 2 task 3')
positive_list = [i for i in list if i > 0]
smallest_positive = min(positive_list)
print(f'The smallest positive number: {smallest_positive}')
```

```
Exercise 3 option 2 task 1
List items that are larger than the previous item: [4, 7, 6, 9, 10, 0, 2]

Exercise 3 option 2 task 2
Number of elements in the list that are greater than both of their neighbors: 4
These elements are: [4, 7, 6, 10]

Exercise 3 option 2 task 3
The smallest positive number: 1
```