

## Exercise 1

In mathematics, the quadratic equation  $ax^2+bx+c=0$  can be solved with the formula  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ .

Write a function `solve_quadratic`, that returns both solutions of a generic quadratic as a pair (2-tuple) when the coefficients are given as parameters. It should work like this:

```
print(solve_quadratic(1,-3,2))
```

```
(2.0,1.0)
```

```
print(solve_quadratic(1,2,1))
```

```
(-1.0,-1.0)
```

```
In [1]: # Defining the function to solve the quadratic formula
def solve_quadratic(a,b,c):
    import math
    discriminant = b ** 2 - (4*a*c)
    # if function sees if the user inputed values will result in real roots
    if discriminant >= 0:

        x1 = (-b + math.sqrt(discriminant))/(2*a)
        x2 = (-b - math.sqrt(discriminant))/(2*a)
        return(x1,x2)
    else:
        print("No Real Roots")
```

```
In [2]: solve_quadratic(1,-3,2)
```

```
Out[2]: (2.0, 1.0)
```

```
In [3]: solve_quadratic(1,2,1)
```

```
Out[3]: (-1.0, -1.0)
```

```
In [4]: solve_quadratic(1,-3,5)
```

```
No Real Roots
```

## Exercise 2

Get a space-separated list of integers from the user, create a tuple of those integers. Then compute and print the result of `hash(tuple)`.

```
In [5]: # Receive input from the user
print("Please enter a list of integers separated by a space: ")

# still need to understand map and split

inputs = list(map(int, input().split()))
print(tuple(inputs))
# convert input to tuple and hash / dont really understand hash
print(hash(tuple(inputs)))
```

Please enter a list of integers separated by a space:

1 2

(1, 2)

-3550055125485641917

### Exercise 3

Take the following two lists. Create a third list by picking a odd-index elements from the first list and even-index elements from the second.

listOne = [3, 6, 9, 12, 15, 18, 21]

listTwo = [4, 8, 12, 16, 20, 24, 28]

```
In [6]: # the two lists
list_one = [3, 6, 9, 12, 15, 18, 21]
list_two = [4, 8, 12, 16, 20, 24, 28]

# obtaining part 1 and part 2 of list 1 and 2
# slice the first list in increments of 2 starting from first odd index repeat for
part_one_list_3 = list_one[1::2]

part_two_list_3 = list_two[0::2]

print(part_one_list_3 + part_two_list_3)
```

[6, 12, 18, 4, 12, 20, 28]

### Exercise 4

Take the following list. Slice it into three equal chunks and reverse each list.

sampleList = [11, 45, 8, 23, 14, 12, 78, 45, 89]

In [7]:

```
sample_list = [11, 45, 8, 23, 14, 12, 78, 45, 89]

c1 = sample_list[0:3]
c2 = sample_list[3:6]
c3 = sample_list[6:]
print(c1[::-1])
print(c2[::-1])
print(c3[::-1])
```

```
[8, 45, 11]
[12, 14, 23]
[89, 45, 78]
```

### Exercise 5

Iterate through a given list and check if a given element already exists in a dictionary as a key's value. If not, delete it from the list.

```
rollNumber = [47, 64, 69, 37, 76, 83, 95, 97]
```

```
sampleDict = {'Zach':47, 'Emma':69, 'Kelly':76, 'Jason':97}
```

```
In [50]: roll_number = [47, 64, 69, 37, 76, 83, 95, 97]

sample_dict = {'Zach':47, 'Emma':69, 'Kelly':76, 'Jason':97}
values = list(sample_dict.values())
print(values)
delete_list = []
for i in roll_number:
    print(i)

    if i == values[0]:

        print("its here")
    elif i == values[1]:

        print("its here")
    elif i == values[2]:

        print("its here")
    elif i == values[3]:

        print("its here")
    else:
        delete_list.append(i)

for d in delete_list:
    roll_number.remove(d)

print(roll_number)
```

[47, 69, 76, 97]

47

its here

64

69

its here

37

76

its here

83

95

97

its here

[47, 69, 76, 97]