#Week 1

#1

```
def q1_sum(list_el):

    total = 0

    for el in list_el:

        try:
            if el % 2 == 0:
                total += el

        except TypeError:

            for x in el:
                if x % 2 == 0:
                    total += x

    return total
```

#2

```
def move_vow(line):
    vow = 'aeiouAEIOU'
    final_vow = ''
    final_cos = ''
    for x in line:
        if x in vow:
            final_vow += x
        else:
            final_cos += x
    final = final_vow + final_cos
  return final
```

#3

```
guests = {    'Randy': 'Germany',    'Karla': 'France',    'Wendy': 'Japan',    'Norman': 'England',    'Sam': 'Argentina'}

def greetings(name):
    if name in guests:
        return f"Hi! I'm {name} and I'm from {guests[name]}."
    else:
        return f"Hi, I'm {name}."
```

#4

```
class Memories:

    def __init__(self, name, age=0, salary=0):

       self.name = name
       self.age = age
       self.salary = salary

    def remember(self, info):

        match info:

            case "name":
                return self.name
            case "salary":
                return self.salary
            case "age":
                return self.age

        return False
```

#5

```
class Test:

    def __init__(self, subject, cor_answ=[], per=''):

        self.subject = subject
        self.cor_answ = cor_answ
        self.per = per

class Student:

    def __init__(self, name):
        self.name = name

    def take_test(self, other, stud_answ=[]):

        cor_st_answ = 0

        for i, x in enumerate(stud_answ):

            if x == other.cor_answ[i]:
                cor_st_answ += 1

        final_grade = (cor_st_answ * 100) / len(other.cor_answ)
        passing_graade = other.per.split("%")
        passing_grade = int(passing_graade[0])

        if stud_answ == other.cor_answ or final_grade >= passing_grade:

            print(f'{self.name} passed the Chemistry test with the score {final_grade}%')

        else:

            print(f'{self.name} failed the Maths test!')
```

#6

```
def histogram(line, type):

    final = [''] * len(line)

    for i, x in enumerate(line):

        final[i] = (type * x)

        print(final[i])
```

#7

```
def  filter_star(the_dict, nr_stars):

    new_dict = {}

    for x in the_dict:

        if len(the_dict[x]) == nr_stars:

            new_dict[x] = '*' * nr_stars

    if len(new_dict) == 0:
        return 'No result found!'

    else:
        return new_dict
```