

Задание 1.

Исходные данные:

Пользователь вводит данные о количестве предприятий, их наименования и прибыль за 4 квартала (т.е. 4 отдельных числа) для каждого предприятия.. Программа должна определить среднюю прибыль (за год для всех предприятий) и вывести наименования предприятий, чья прибыль выше среднего и отдельно вывести наименования предприятий, чья прибыль ниже среднего.

Решение:

```
a = dict(baza='name', kvartal='number')
b = []
g = 0
def sr(a, b):
    try:
        a = input("a = ")
        d = int(input("d = "))
        e = int(input("e = "))
        x = int(input("x = "))
    except ValueError and ZeroDivisionError:
        return
    f = sum(range(e, x))
    g = f / d
    h = g / 2
    k = g / 1.5
    l = g / 2.5
    w = g * 2
    s = g, h, k, l, w
    return s

if g >= 0:
    print(f"{a}, Прибыль больше среднего")
else:
    g <= 0
    print(f"{a}, Прибыль меньше среднего")

print(f"{g >= 0, a}, {g <= 0, a}", sr(a, b))
```

Задание 2.

Исходные данные:

Написать программу сложения и умножения двух шестнадцатеричных чисел. При этом каждое число представляется как коллекция, элементы которой это цифры числа.

Например, пользователь ввёл A2 и C4F. Сохранить их как ['A', '2'] и ['C', '4', 'F'] соответственно. Сумма чисел из примера: ['C', 'F', '1'], произведение - ['7', 'C', '9', 'F', 'E'].

Решение:

```
import collections
from collections import deque
a = hex(55)
b = hex(75)
def my_sum_hex(a, b):
    try:
        hex_number = {0: 0, 1: 1,
2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9, 'A': 10, 'B': 11, 'C': 12, 'D': 13, 'E': 14, 'F': 15}
        hex_number = {0: 0, 1: 1,
```

```

2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9, 'A': 10, 'B': 11, 'C': 12, 'D': 13, 'E': 14, 'F': 15}
    result = deque()
    transfer = 0
    if len(b)>len(a):
        a, b = deque(b), deque(a)
    else:
        a, b = deque(a), deque(b)
    transfer = 0
    if result==16:
        result.appendleft(hex_number[result])
    transfer = 0
    if transfer:
        result.appendleft(1)
except zerodivisionerror:
    return
my_sum_hex = [a + b]
return my_sum_hex

```

```

a = hex(55)
b = hex(75)
mult_hex = deque()
def mult_hex(a, b):
    try:
        hex_number = {0: 0, 1: 1,
2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9, 'A': 10, 'B': 11, 'C': 12, 'D': 13, 'E': 14, 'F': 15}
        hex_number = {0: 0, 1: 1,
2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9, 'A': 10, 'B': 11, 'C': 12, 'D': 13, 'E': 14, 'F': 15}
        result = deque()
        abak = deque([deque() for _ in range(len(b))])
        for i in range(len(b)):
            n = hex_number[b.pop()]
            for k in range(len(a) - 1, - 1, - 1):
                abak[i].appendleft(n * a[k])
            for _ in range(i):
                abak[i].append(0)
            transfer = 0
            for _ in range(len(abak[-1])):
                result = transfer
            for i in range(len(abak)):
                if abak[i]:
                    abak.appendleft(result)
                else:
                    abak.append([i]%16)
            transfer = 1
            if transfer:
                abak.appendleft(hex_number[transfer])
except zerodivisionerror:
    return
mult_hex = a * b
return mult_hex

```

```

a = list(input("Введите первое шестнадцатиричное число: ").upper())

```

```
b = list(input("Введите второе шестнадцатичное число: ").upper())  
print(a, b)  
print(my_sum_hex(a, b))  
print(mult_hex(a, b))
```