

Задание 1.

Исходные данные:

Отсортируйте по убыванию методом пузырька одномерный целочисленный массив, заданный случайными числами на промежутке $[-100; 100)$. Выведите на экран исходный и отсортированный массивы. Сортировка должна быть реализована в виде функции. По возможности доработайте алгоритм (сделайте его умнее).

Решение:

```
import random
from random import randint
randint(-100, 100)
n = randint(-100, 100)
mas = [randint(-100, 100) for i in range(n)]
for i in range(n):
    print(mas[i], sep=" ")
print(' ')
for i in range(n - 1):
    for k in range(n - 2, i - 1, -1):
        if mas[k] < mas[k+1]:
            mas[k], mas[k+1] = mas[k+1], mas[k]
for i in range(n):
    print(mas[i], sep=" ")
print(' ')
```

Задание 2.

Исходные данные:

Отсортируйте по возрастанию методом слияния одномерный вещественный массив, заданный случайными числами на промежутке $[0; 50)$. Выведите на экран исходный и отсортированный массивы.

Решение:

```
import random
from random import randint
randint(1, 50)
list = [randint(1, 50)]
list = [randint(1, 50) for i in list]
def merge_sort(list, start, end):
    merge_sort(list, 1, len(list))
    if end - start > 1:
        mid = (start + end) // 2
        merge_sort(list, start, mid)
        merge_sort(list, mid, end)
        merge_list_1(list, start, mid, end)

def merge_list_1(list, start, mid, end):
    left = list[start:mid]
    right = list[mid:end]
    d = start
    i = 0
    k = 0
    while (start + i < mid and mid + k < end):
        if (left[i] <= right[k]):
            list[d] = left[i]
            i = i + 1
```

```

    else:
        list[d] = right[k]
        k = k + 1
    d = d + 1
if start + i < mid:
    while d < end:
        list[d] = left[i]
        i = i + 1
        d = d + 1
else:
    while d < end:
        list[d] = right[k]
        k = k + 1
        d = d + 1

print("sorted list: ", end="")
print(list)

```

Задание 3.

Исходные данные:

Массив размером $2m + 1$, где m – натуральное число, заполнен случайным образом. Найдите в массиве медиану. Медианой называется элемент ряда, делящий его на две равные части: в одной находятся элементы, которые не меньше медианы, в другой – не больше медианы.

Задачу можно решить без сортировки исходного массива. Но если это слишком сложно, то используйте метод сортировки, который не рассматривался на уроках.

Решение:

```

import random
from random import randint
randint(1, 151)
list = [randint(1, 151) for i in range(randint(1, 151))]
def list_sort(list):
    for i in range(randint(1, 151)):
        n = len(list)
        if n % 2 == 0:
            median1 = list[n // 2]
            median2 = list[n // 2 - 1]
            median = (median1 + median2) / 2
        else:
            median = list[n // 2]
    s = sorted(list)
    return s[n // 2 + 1]

print(list_sort(list))

```