

Суть решения: считать все товары, отсортировать по убыванию. Создать массив простых товаров. Для того, чтобы сумма была наименьшей, использовать акции нужно для самых дорогих простых товаров. Считаем обе акции для самых дорогих товаров, вычисляем меньшую и считаем разницу.

Ход решения:

- 1) Для начала открываем файл, считываем в переменную n кол-во товаров, в массив mas – стоимости каждого товара. Сортируем массив по убыванию.

```
with open('26_1.txt') as f:
    n=int(f.readline())
    mas=[int(x) for x in f]
    mas.sort(reverse=True)
```

- 2) Создаем функцию isprime, которая будет определять, является ли число, обозначающее цену товара простым.

```
def isprime(n):
    d=2
    while d*d<=n:
        if n%d==0:
            return False
        d+=1
    return True
```

- 3) Создаем: mas_pr (массив “простых” товаров), и суммы s1 и s2 (сумма покупки при первой и второй акциях соответственно)

```
mas_pr=[]
s1, s2 = 0, 0
```

- 4) Пробегаемся по массиву товаров, “простые” товары добавляем в массив простых товаров, обычные добавляем в сумму покупки.

```
for i in mas:
    if isprime(i):
        mas_pr.append(i)
    else:
        s1+=i
        s2+=i
```

- 5) Создаем переменные k1 и k2, в которые добавляем количество товаров, которые участвуют в первой и второй акциях соответственно.

```
k1=len(mas_pr)//4
k2=len(mas_pr)//2
```

6) Считаем сумму товаров (сначала проверяем, не равны ли k1 и k2 нулю, ибо в таком случае ни один товар не будет участвовать в акции):

- 1) Для первой акции k1 товаров идут бесплатно, поэтому мы их не считаем и делаем срез [k1::] и суммируем оставшиеся товары.
- 2) Для второй акции мы считаем первые k2 товаров за полцены (делаем срез [:k2:]), а остальные просто суммируем (делаем срез [k2::])

```
if k1!=0:
    for i in mas_pr[k1::]:
        s1+=i
if k2!=0:
    for i in mas_pr[:k2:]:
        s2+=i//2
    for i in mas_pr[k2::]:
        s2+=i
```

7) Считаем минимальную сумму из s1 и s2 и разность.

```
print(min(s1, s2), max(s1, s2)-min(s1, s2))
```

Ответ: 8362843 78948.