

Задание выполняется с использованием прилагаемых файлов.

[Скачать архив с файлами.](#)

Набор данных состоит из троек натуральных чисел. Необходимо распределить все числа на три группы, при этом в каждую группу должно попасть ровно одно число из каждой исходной тройки. Сумма всех чисел в первой группе должна быть нечётной, во второй – чётной. Определите минимально возможную сумму всех чисел в третьей группе.

Входные данные

Первая строка входного файла содержит число N – общее количество троек в наборе. Каждая из следующих N строк содержит три натуральных числа, не превышающих 10 000.

Пример входного файла

```
3
1 2 3
5 12
8 12 4
6 9 7
```

Для указанных данных искомая сумма равна 11, она соответствует такому распределению чисел по группам: (2, 8, 7), (3, 12, 9), (1, 4, 6).

Вам даны два входных файла (A и B), каждый из которых имеет описанную выше структуру. В ответе укажите два числа: сначала значение искомой суммы для файла A, затем для файла B.

Число

Число



185

100918194



```
sum min=11
list min=1,4,6
```

```
sum extra=2+8+7+3+12+9=41=even+odd
list extra=2,8,7,3,12,9
```

```
file1 = open("/Users/ivangrablov/Documents/python/PyCharm/ege2/25-B.txt", "r")
list1 = file1.readlines()
```

```
for i in range(len(list1)):
    list1[i] = list1[i][:-1]
```

```
N = list1[0]
print(N)
```

```
list1 = list1[1:]
print(list1)
```

```
mas1 = []
for i in range(len(list1)):
    mas1.append(list1[i].split())
print(mas1)
```

```
mas_min = []
mas_other = []
for i in range(len(mas1)):
    for k in range(len(mas1[i])):
        mas1[i][k] = int(mas1[i][k])
    mas1[i].sort()
    mas_min.append(mas1[i][0])
    mas_other.append(mas1[i][1])
    mas_other.append(mas1[i][2])
```

```
print("")
print(mas_min)
print(mas_other)
```

```
sum1 = sum(mas_other)
print(sum1)
sum2 = sum(mas_min)
print(sum2)
```

```
if sum1 % 2 != 0:
    print('otvet = ', sum2)
else:
    num1 = 0
    num2 = 0
    min_f = 0
    for i in range(len(mas_min)):
        k = 2 * i
        if min_f == 0 and (mas_min[i] % 2 == 0 and mas_other[k] % 2 != 0 or mas_min[i] % 2 != 0 and mas_other[k] % 2 == 0):
            num1 = mas_min[i]
            num2 = mas_other[k]
            min_f = 1
        elif min_f == 1 and (mas_min[i] % 2 == 0 and mas_other[k] % 2 != 0 or mas_min[i] % 2 != 0 and mas_other[k] % 2 == 0):
            if abs(num1 - num2) > abs(mas_min[i] - mas_other[k]):
                num1 = mas_min[i]
                num2 = mas_other[k]
```

```
k = 2 * i + 1
if min_f == 0 and (
    mas_min[i] % 2 == 0 and mas_other[k] % 2 != 0 or mas_min[i] % 2 != 0 and mas_other[k] % 2 == 0):
    num1 = mas_min[i]
    num2 = mas_other[k]
    min_f = 1
elif min_f == 1 and (
    mas_min[i] % 2 == 0 and mas_other[k] % 2 != 0 or mas_min[i] % 2 != 0 and mas_other[k] % 2 == 0):
    if abs(num1 - num2) > abs(mas_min[i] - mas_other[k]):
        num1 = mas_min[i]
        num2 = mas_other[k]
    #print(abs(num1 - num2))
print('otvet = ', sum2 + abs(num1 - num2))
```

```
#print(num1, num2)
```