

Задача № 1

В стране Z для поступления в университет ученикам школ необходимо сдать экзамен “нервотрёпка” по 4 предметам. На каждую дисциплину отводится по 3 попытки. Результатом экзамена будет считаться сумма средних арифметических всех попыток по каждому предмету. Если же по какой-либо дисциплине средним арифметическим будет не целое число, то округляют не в пользу сдающего. Также известно, что при получении наивысшего балла хотя бы за одну попытку, ученик автоматически набирает 100 баллов по данному предмету и получает 100 зеров, - зеры – валюта страны Z. По одному предмету ученик может получить 100 зеров не более 1 раза. После получения результатов экзамена абитуриент выбирает вуз, куда собирается поступать. Если он не проходит по баллам на бюджет, то его зачисляют на платное обучение вне зависимости от согласия абитуриента, так как высшее образование в стране Z обязательно по закону. Если человеку не хватает денег, то он становится должником. В случае, когда несколько человек имеют одинаковые баллы и борются за бюджет, они все проходят. Необходимо в ответ указать количество бедолаг, которые погрязли в долгах и сумму наименьшего долга одного из абитуриентов. Гарантируется, что хотя бы 1 человек – должник.

Входные данные представлены в файле следующим образом. В первой строке входного файла записано число N ($1 \leq N \leq 20000$) – количество поступающих, число K ($1 \leq K \leq 100$) – количество вузов. Каждая из следующих N строк содержит ID абитуриента, ID вуза, который он выбрал и сумма денег на счету до получения выигрыша 100 балльниками. Далее в следующих $4 * N$ строках содержится ID абитуриента и полученные баллы за каждую отдельную попытку сдачи экзамена “нервотрёпка” по определённому предмету. В оставшихся K строках записаны ID вуза, количество бюджетных мест и стоимость контрактного обучения.

Пример входного файла:

```
4 2
101 11 30
102 12 250
103 12 500
104 11 0
101 50 55 60
102 50 55 70
103 80 83 93
104 99 99 99
104 98 98 98
103 98 98 98
102 100 98 98
101 56 54 55
101 84 87 87
102 72 67 30
103 94 96 92
104 100 100 100
101 30 68 54
102 70 81 76
103 79 97 97
104 100 100 100
12 1 300
11 1 210
```

При таких исходных данных:

Номер студента	Результат экзамена	Сумма на счету после поступления в вуз
101	246	-180
102	285	50
103	368	500
104	397	200

Ответ: 1 180

Решение:

```

with open('ex.txt') as f:
    n, k = map(int, f.readline().split())

    st = dict()
    vuz = dict()

    for _ in range(n):
        id_st, id_vuz, mon = map(int, f.readline().split())
        st[id_st] = [id_vuz, mon, 0]

    for _ in range(4 * n):
        id_st, *ex = map(int, f.readline().split())

        if 100 in ex:
            st[id_st][1] += 100
            st[id_st][2] += 100
        else:
            st[id_st][2] += (sum(ex) // 3)

    for _ in range(k):
        id_vuz, budz, price = map(int, f.readline().split())

        vuz[id_vuz] = [budz, price, []]

    for st_id, info in st.items():
        id_vuz, mon, ball = info
        vuz[id_vuz][2].append((ball, st_id))

    cnt_dolg = 0
    dolg = -100000000

    for id_vuz, info in vuz.items():
        budz, price, info_st = info

        info_st = sorted(info_st, reverse=True)

        for ball, st_id in info_st[budz:]:
            if info_st[budz - 1][0] == ball:
                continue

            babki = st[st_id][1] - price

            if babki < 0:
                cnt_dolg += 1
                dolg = max(dolg, babki)

    print(cnt_dolg, abs(dolg))

```

Пояснение:

Для хранения информации о студентах и вузах будем использовать словари. В начале добавим информацию о студентах, а именно: вуз, в который они хотят поступить, текущее кол-во денег и кол-во баллов (изначально 0). Потом обновим информацию о баллах, если у человека 100 баллов, то увеличим кол-во денег и баллов, если нет, то просто увеличим кол-во баллов. Далее заполним информацию про вузы. Кол-во бюджетных мест, цена обучения, а также баллы студентов, которые хотят поступить (изначально пустой массив). Потом заполним информацию о студентах, будем перебирать словарь и добавлять в id вуза информацию о студенте (кол-во баллов и идентификатор). Затем перебираем информацию о вузе (id и информацию о студентах, которые хотят поступить в него). Будем сортировать по баллам людей. Далее перебираем балл студента и его id из тех людей, которые не поступили (срез массива). Если у студента такой же балл, как и у человека, который поступил, то пропускаем его (по условию), иначе находим сколько у него осталось денег, если их меньше 0, то человек становится должником, увеличиваем счетчик и находим минимальный долг.

Задача № 2

В город “Дно” неожиданно приехала известная певица. Это событие не сильно оглашалось, поэтому о наличии билетов знали только те, кто проходил рядом с домом культуры. По итогу люди либо сразу покупали билеты, либо уходили домой, если им не хватало денег, и позже возвращались. Покупка билета занимает 4 минуты, поэтому у кассы образовалась очередь. Определите количество людей, которые простояли в очереди зря, так и не успев купить билеты, а также время закрытия кассы в секундах, если учитывать, что она перестаёт работать после продажи последнего билета.

Входные данные представлены в файле следующим образом. В первой строке входного файла записано число N ($1 \leq N \leq 20000$) – количество человек, увидевших объявление о концерте, число K ($1 \leq K \leq 100$) – количество билетов. Каждая из следующих N строк содержит 2 числа: время прихода в дом культуры в секундах от начала суток и время, которое было затрачено на путь в секундах. Если оба числа в строке совпадают, считать, что у покупателя были деньги с собой. Гарантируется, что одновременно прийти к кассе или в очередь не могут сразу 2 или более человек.

Пример входного файла:

```
6 2
1000 600
1500 1500
1100 450
1700 300
1200 360
1400 700
Ответ: 2 1980
```

Решение:

```
with open('ex1.txt') as f:
    n, k = map(int, f.readline().split())
    tm = []

    for _ in range(n):
        st, end = map(int, f.readline().split())

        if st == end:
            tm.append([st, st, end])
        else:
            tm.append([st + end, st + end, end])

    tm.sort()
    cnt = 1
    ans_cnt = 0
    fl = False
    ans = 0

    tm[0][0] += 240

    for i in range(1, len(tm)):
        cnt += 1
        tm[i][0] += 240 * (i + 1) - (tm[i][1] - tm[i - 1][1])

        if cnt == k:
            ans = tm[i][0]
            fl = True

        if fl:
            if tm[i][1] > ans:
                ans_cnt += 1

    print(ans_cnt, ans)
```

Пояснение:

Сначала читаем данные и добавляем в массив `tm` (timeline), если `st == end`, то есть по условию у человека были деньги, то просто добавляем время прихода, иначе учитываем время для дороги. Потом сортируем по времени прихода (то есть время прихода изначальное + дороги). Далее учитываем время покупки билета других людей + время дороги, если кол-во людей, которые купили билеты равняется кол-во билетов, то ответом будет время прихода + покупка билетов у текущего человека, если у других время прихода больше, то они стояли в очереди