

## Информатика. Программирование. Задание 26. Современные задачи.

### Содержание

<b>1</b>	<b>Разбор задач с вебинара</b>	<b>2</b>
1.1	Пример 1 . . . . .	2
1.2	Пример 2 . . . . .	4
1.3	Пример 3 . . . . .	6

# 1 Разбор задач с вебинара

## 1.1 Пример 1

Входной файл содержит сведения о заявках на проведение занятий в конференц-зале. В каждой заявке указаны время начала и время окончания мероприятия (в минутах от начала суток). Если время начала одного мероприятия меньше времени окончания другого, то провести можно только одно из них. Если время окончания одного мероприятия совпадает с временем начала другого, то провести можно оба. Определите максимальное количество мероприятий, которое можно провести в конференц-зале и самое позднее время окончания последнего мероприятия.

**Входные данные** представлены в файле **26-128.txt** следующим образом. Первая строка входного файла содержит натуральное число  $N$  ( $1 \leq N \leq 1000$ ) – количество заявок на проведение мероприятий. Следующие  $N$  строк содержат пары чисел, обозначающих время начала и время окончания мероприятий. Каждое из чисел натуральное, не превосходящее 1440. Запишите в ответе два числа: максимальное количество мероприятий, которое можно провести в конференц-зале и самое позднее время окончания последнего мероприятия (в минутах от начала суток).

**Пример входного файла:**

5

10 150

100 110

131 170

131 180

120 130

При таких исходных данных можно провести максимум три мероприятия, например, по заявкам 2, 3 и 5. Конференц-зал освободится самое позднее на 180-й минуте, если состоятся мероприятия по заявкам 2, 4, 5. Ответ: 3 180

### Решение:

Если нам нужно провести наибольшее количество мероприятий, нужно смотреть на их время окончания. Для этого отсортируем данные по времени окончания мероприятий.

Это можно сделать в файле Excel.

Мы воспользуемся лямбда-функцией.

---

```
f = open('26-128.txt') # Открываем файл
n = int(f.readline()) # Считываем первое число - количество заявок
a = [list(map(int, i.split())) for i in f]
a.sort(key = lambda x: x[1]) # Сортировка по времени окончания мероприятий
hall = [a.pop(0)] # Список проводимых мероприятий
# В список записали первое мероприятие и удалили его из списка 'a'

for i in a:
    # Если время начала нового мероприятия >= времени окончания предыдущего
    # или в hall ничего нет
    if (not hall) or (i[0] >= hall[-1][1]):
        hall.append(i) # Добавляем новое мероприятие
for i in a :
    # Если время начала мероприятия больше времени окончания предпоследнего
    if i[0] >= 991:
        print(i) # Последнее число - самое позднее время окончания

print(len(hall)) # Количество мероприятий
```

---

Ответ: 16 1345

## 1.2 Пример 2

(Досрочный ЕГЭ-2023) Входной файл содержит заявки пассажиров, желающих сдать свой багаж в камеру хранения. в заявке указаны время сдачи багажа и время освобождения ячейки (в минутах от начала суток). Багаж одного пассажира размещается в одной свободной ячейке с минимальным номером. Ячейки пронумерованы начиная с единицы. Размещение багажа в ячейке или её освобождение происходит в течение 1 мин. Багаж можно поместить в только что освобождённую ячейку начиная со следующей минуты. Если в момент сдачи багажа свободных ячеек нет, то пассажир уходит. Определите, сколько пассажиров сможет сдать свой багаж в течение 24 ч и какой номер будет иметь ячейка, которую займут последней. Если таких ячеек несколько, укажите минимальный номер ячейки.

**Входные данные** представлены в файле **26-111.txt** следующим образом. В первой строке входного файла находится натуральное число  $K$ , не превышающее 1000, – количество ячеек в камере хранения. Во второй строке – натуральное число  $N$  ( $N \leq 1000$ ), обозначающее количество пассажиров, Каждая из следующих  $N$  строк содержит два натуральных числа, каждое из которых не превышает 1440: указанное в заявке время размещения багажа в ячейке и время освобождения ячейки (в минутах от начала суток). Запишите в ответе два числа: количество пассажиров, которые смогут воспользоваться камерой хранения, и номер последней занятой ячейки.

**Пример входного файла:**

2

5

30 60

40 1000

59 60

61 1000

1010 1440

При таких исходных данных положить вещи в камеру хранения смогут первый, второй, четвёртый и пятый пассажиры. Последний пассажир положит вещи в ячейку 1, так как ячейки 1 и 2 будут свободны. Ответ: 4 1

## Решение:

---

```
f = open('26-111.txt') # Открываем файл
k = int(f.readline()) # Количество ячеек
n = int(f.readline()) # Количество пассажиров
a = sorted([list(map(int, i.split())) for i in f])
cell = [[] for i in range(k)] # Список с камерами
с = 0 # Счетчик пассажиров, которые сдали багаж
for i in range(n): # Для каждого пассажира
    for j in cell: # Проверяем все ячейки
        # Если ячейка пустая или время сдачи > время забираяния последнего
        if (not j) or (a[i][0] > j[-1][1]):
            j.append(a[i]) # Добавляем багаж в ячейку
            с += 1 # Увеличиваем счетчик
    # Останавливаем цикл, чтоб не записывать 1 багаж в другие ячейки
    break

mx = 0 # Время сдачи последнего багажа
for i in cell: # Для всех ячеек
    mx = max(mx, i[-1][0]) # Проверяем время сдачи
print(mx) # Программа выведет 999

print(с) # Количество пассажиров

for i in range(k):
    if cell[i][-1][0] == 999: # Если время сдачи равно 'mx'
        print(i + 1) # Номер последней занятой ячейки
```

---

Ответ: 344 53

### 1.3 Пример 3

**(ЕГЭ-2023)** На производстве штучных изделий  $N$  деталей должны быть отшлифованы и окрашены. Для каждой детали известно время её шлифовки и время окрашивания. Детали пронумерованы начиная с единицы. Параллельная обработка деталей не предусмотрена. На ленте транспортёра имеется  $N$  мест для каждой из  $N$  деталей. На ленте транспортёра детали располагают по следующему алгоритму:

- все  $2N$  чисел, обозначающих время окрашивания и шлифовки для  $N$  деталей, упорядочивают по возрастанию;
- если минимальное число в этом упорядоченном списке – это время шлифовки конкретной детали, то деталь размещают на ленте транспортёра на первое свободное место от её начала;
- если минимальное число – это время окрашивания, то деталь размещают на первое свободное место от конца ленты транспортёра
- если число обозначает время окрашивания или шлифовки уже рассмотренной детали, то его не принимают во внимание.

Этот алгоритм применяется последовательно для размещения всех  $N$  деталей. Определите номер последней детали, для которой будет определено её место на ленте транспортёра, и количество деталей, которые будут отшлифованы до неё.

**Входные данные** представлены в файле **26-129.txt** следующим образом. Первая строка входного файла содержит натуральное число  $N$  ( $1 \leq N \leq 1000$ ) – количество деталей. Следующие  $N$  строк содержат пары чисел, обозначающих соответственно время шлифовки и время окрашивания конкретной детали (все числа натуральные, различные). Запишите в ответе два натуральных числа: сначала номер последней детали, для которой будет определено её место на ленте транспортёра, затем количество деталей, которые будут отшлифованы до неё.

### Решение:

---

```
f = open('26-129.txt')
n = int(f.readline())
a = []
for i in range(n):
    g, p = map(int, f.readline().split()) # Время шлифовки и время окраски
    if g < p: # Если время шлифовки меньше
        # Записываем время шлифовки, тип работы, номер детали
        a.append([g, 1, i + 1])
    else: # Если время окраски меньше
        # Записываем время окраски, тип работы, номер детали
        a.append([p, 2, i + 1])
a.sort()

l_start = [] # Начало конвейерной ленты
l_end = [] # Конец конвейерной ленты
for i in a:
    if i[1] == 1: # Если деталь шлифуют
        l_start.append(i) # Добавляем ее в начало ленты
    else: # Если деталь красят
        l_end.append(i) # Добавляем ее в конец ленты
l = l_start + l_end[::-1] # Конвейерная лента
c = 0 # Счетчик окрашенных деталей
print(a[-1]) # Последняя размещенная на ленте деталь с номером 895

for i in l:
    if i[1] == 1: # Если деталь шлифуют
        c += 1 # Увеличиваем счетчик
print(c) # c = 499 -> ДО последней детали будет отшлифовано 488
```

---

Ответ: 895 488