

Тренировочный контекст: разработка бэкенда

26 апр 2023, 20:51:56

старт: 23 апр 2023, 22:03:59

начало: 1 янв 2021, 02:00:00

В. Посадка в самолет

Ограничение времени	2 секунды
Ограничение памяти	512Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В самолете n рядов и по три кресла слева и справа в каждом ряду. Крайние кресла (А и F) находятся у окна, центральные (С и D) – у прохода. На регистрацию приходят группы из одного, двух или трех пассажиров. Они желают сидеть рядом, то есть на одном ряду и на одной стороне: левой или правой. Например, группа из двух пассажиров может сесть на кресла В и С, но не может сесть на кресла С и D, потому что они разделены проходом, а также не может сесть на кресла А и С, потому что тогда они окажутся не рядом. Кроме того, один из пассажиров каждой группы очень требовательный – он хочет сесть либо у окна, либо у прохода. Конечно же, каждая группа из пассажиров хочет занять места в ряду с как можно меньшим номером, ведь тогда они скорее выйдут из самолета после посадки. Для каждой группы пассажиров определите, есть ли места в самолете, подходящие для них.

Формат ввода

Первая строка содержит число n ($1 \leq n \leq 100$) – количество рядов в самолете. Далее в n строках вводится изначальная рассадка в самолете по рядам (от первого до n -го), где символами . (точка) обозначены свободные места, символами # (решетка) обозначены занятые места, а символами _ (нижнее подчеркивание) обозначен проход между креслами С и D каждого ряда.

Следующая строка содержит число m ($1 \leq m \leq 100$) – количество групп пассажиров. Далее в m строках содержатся описания групп пассажиров. Формат описания такой: $num\ side\ position$, где num – количество пассажиров (число 1, 2 или 3), $side$ – желаемая сторона самолета (строка left или right), $position$ – желаемое место требовательного пассажира (строка aisle или window).

Формат вывода

Если группа может сесть на места, удовлетворяющие ее требованиям, то выведите строку `Passengers can take seats:` и список их мест в формате `row letter`, упорядоченный по возрастанию буквы места. Затем выведите в n строках получившуюся рассадку в самолете, в формате, описанном выше, причем места, занятые текущей группой пассажиров, должны быть обозначены символом X. Если группа не может найти места, удовлетворяющие ее требованиям, то выведите строку `Cannot fulfill passengers requirements`.

Ответ сравнивается с правильным посимвольно, поэтому ваше решение не должно выводить никаких лишних символов, в том числе лишних переводов строк или пробельных символов в концах строк. В конце каждой строки (включая последнюю) должен быть выведен символ перевода строки.

Пример

Ввод

```
4
..._.#.
.##_...
.#.##_#
..._...
7
2 left aisle
3 right window
```

Вывод

```
Passengers can take seats: 1B 1C
.XX_.#.
.##_...
.#.##_#
..._...
Passengers can take seats: 2D 2E 2F
.##_#.
.##_XXX
```

Ввод

```

2 left window
3 left aisle
1 right window
2 right window
1 right window

```

Вывод

```

.#_.##
..._...
Passengers can take seats: 4A 4B
.##_.#.
.##_###
.#_.##
XX._...
Cannot fulfill passengers requirements
Passengers can take seats: 1F
.##_.#X
.##_###
.#_.##
##_._...
Passengers can take seats: 4E 4F
.##_.##
.##_###
.#_.##
##_._XX
Cannot fulfill passengers requirements

```

Язык Python 3.9 (PyPy 7.3.11)

Набрать здесь

Отправить файл

```

1 ap_rows = list()
2 qu_list = list()
3 with open('input.txt') as f:
4     r_amount = int(f.readline().strip('\n'))
5     for i in range(r_amount):
6         ap_rows.append(f.readline().strip('\n').split('_'))
7     qu_len = int(f.readline().strip('\n'))
8     i = 0
9     for i in range(qu_len):
10         qu_list.append(f.readline().strip('\n').split(' '))
11
12
13 def dump_update(side, row, seats):
14     if side == "left":
15         r = 0
16     else:
17         r = 1
18     for seat in seats:
19         l = list(ap_rows[row][r])
20         l[seat] = 'X'
21         ap_rows[row][r] = "".join(l)
22     for i in ap_rows:
23         print(i[0] + '_' + i[1])
24     for seat in seats:
25         l = list(ap_rows[row][r])
26         l[seat] = '#'
27         ap_rows[row][r] = "".join(l)
28
29
30 def create_taken(side, row, seats):
31     seats_letters = "ABCDEF"
32     offset = 0
33     if side == "right":
34         offset = 3
35     message = "Passengers can take seats:"
36     for seat in seats:
37         message += " " + str(row + 1) + seats_letters[offset + seat]
38

```

Отправить

Предыдущая

Следующая