На фондовом рынке запланированы первичные размещения облигаций с номиналом 1000  
условных единиц, по которым каждый день выплачивается купон размером 1 уе.  
Погашение номинала облигации (то есть выплата 1000 условных единиц) происходит в  
конце срока.  
  
Каждая облигация на рынке характеризуется названием (некая строка) и ценой, цена  
выражается в виде процентов от номинала, то есть цена 98.5 соответствует цене  
98,5% \* 1000 = 985 условных единиц.  
  
У некоего инвестора есть информация о том, какие предложения по облигациям будут  
размещаться на рынке в ближайшие N дней. По каждому дню он знает, какие лоты  
будут размещены на бирже: название облигации, цену и количество в штуках. Каждый  
день на рынке может быть от 0 до M лотов. Инвестор располагает суммой денежных  
средств в количестве S.  
  
Необходимо определить какие лоты в какие дни нужно купить, чтобы получить  
максимальный доход с учетом следующих условий:  
  
1. Инвестор может только покупать облигации. Купленные облигации не продаются.  
2. Инвестор может купить только весь лот целиком при наличии доступных  
денежных средств.  
3. Выплаченные купоны по купленным облигациям не реинвестируются, то есть не  
Выходные данные  
Мегатрейдер  
увеличивают сумму доступных денежных средств.  
4. Все купленные облигации будут погашены в день N+30.  
5. Доход рассчитывается на день N+30, то есть после погашения облигаций.  
  
  
Входные данные:  
  
На первой строке будут даны числа N, M и S. Далее будет идти k строк вида “<день>  
<название облигации в виде строки без пробелов> <цена> <количество>”. Ввод будет  
завершен пустой строкой.  
  
2 2 8000  
1 alfa-05 100.2 2  
2 alfa-05 101.5 5  
2 gazprom-17 100.0 2  
  
Выходные данные:  
  
В первой строке необходимо указать сумму дохода, полученного трейдером на день  
N+30. В последующих строках привести купленные лоты в таком же формате,  
который используется во входных данных. Последняя строка должна быть пустой.  
  
135  
2 alfa-05 101.5 5  
2 gazprom-17 100.0 2  
  
  
Дополнительно необходимо указать:  
1. оценку необходимой памяти для его выполнения (можно экспериментально измерить)  
2. ограничения на размер входных параметров, при которых алгоритм будет  
выполняться разумное время (до 5 секунд, например)  
3. (если есть) использованные сторонние пакеты для оптимизированной версии (requirements.txt).  
  
Опционально: сравнить неоптимизированную (чистый Python) и оптимизированную реализацию.