

Плентус

Сложность



Всего тестов: 10

Кол-во баллов: 10



Условия

В магазине продается неограниченное количество плентусных реек с длинами $1, 2, \dots, n$. Вы хотите купить некоторый набор реек, чтобы получить суммарную длину L .

Разрешается чтобы в вашем наборе были рейки имеющие одну и ту же длину. Какое минимальное количество реек нужно, чтобы получить длину L ?

Формат входных данных

Единственная строка входных данных содержит два целых числа n и L
($1 \leq n \leq 100000$, $1 \leq L < 110$)

Формат выходных данных

Выведите ровно одно целое число — минимальное количество реек, которое нужно чтобы получить длину L .

Примеры

Входные данные:

5 11

Выходные данные:

3

Коробки

Сложность



Всего тестов: 10

Кол-во баллов: 20



Условия

На складе лежат n коробок с яблоками (четное число).

Кладовщик хочет сформировать $n/2$ паллет. Каждая паллета должна состоять ровно из двух коробок. Две коробки могут устойчиво лежать на паллете только тогда, когда их вес одинаков.

В коробки можно докладывать яблоки, каждое из которых увеличивает вес коробки на единицу.

Кладовщик хочет знать, какое минимальное количество яблок необходимо доложить в коробки, чтобы сформировать ровно $n/2$ паллет (то есть каждая пара коробок должна лежать на паллете). Ваша задача — найти это количество.

Формат входных данных

Первая строка входных данных содержит одно целое число n ($2 \leq n \leq 100$) — количество коробок. Гарантируется, что n всегда является четным числом.

Вторая строка входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a[i] \leq 100$), где $a[i]$ равно весу i -й коробки.

Формат выходных данных

Выведите одно целое число — минимальное количество яблок, которое необходимо доложить в коробки, чтобы сформировать ровно $n/2$ паллет.

Примеры

Входные данные:

6
5 10 2 3 14 5

Выходные данные:

5

Дефектный XML

Сложность



Всего тестов: 10 Кол-во баллов: 50 ?

Условия

Вам дан испорченный XML-документ, в котором потеряна часть символов (например, пробелы, кавычки). При этом известно, что в для каждого тега документа верно следующее:

- если у него есть вложенные теги, то первый из них - не испорчен
- все вложенные на одном уровне теги всегда относятся к одному типу данных
- каждый тег имеет уникальный атрибут id

Кроме того, известно что корневой тег не испорчен.

Вам необходимо восстановить из этого документа полезные данные с сохранением их структуры. Полезными данными считается значение атрибута value каждого тега, у которого есть этот атрибут, и его удалось восстановить.

Формат входных данных

На вход (stdin) поступает XML файл.

Пример испорченного файла:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <group id="1" name="A">
    <item name="a1" id="2" value="a 1"/>
    <itemname=a2id=3value=a2/>
    <itemname=a3id=4value=a3/>
  </group>
  <groupid=5name=B>
    <group id="6" name="B1">
      <file id="7" value="b11"/>
      <fileid=8value=b12/>
    </group>
  </group>
</root>
```

Формат выходных данных

На выход нужно выдать восстановленные данные в формате

parent_id - id родительского узла в дереве тегов
id - идентификатор тега
value - полезные данные (может отсутствовать)

Пример вывода для приведенного выше файла:

```
0 1
1 2 a 1
1 3 a2
1 4 a3
0 5
5 6
6 7 b11
6 8 b12
```

Автоматическая торговля

Сложность



Всего тестов: 10

Кол-во баллов: 100 ?

Условия

Реализуйте управляющую логику для автомата по продаже товаров. Автомат должен выполнять следующие функции

- формирование "корзины" из выбранных покупателем товаров
- прием наличных денег - купюр и монет
- выдача оплаченного товара
- выдача сдачи
- отмена заказа

Автомат продает штучные товары N ($1 \leq N \leq 100$) видов, для каждого товара на панели автомата расположена кнопка "Купить".

При нажатии этой кнопки товар откладывается в "корзину". После выбора всех нужных товаров покупатель должен оплатить заказ и нажать кнопку "Выдать".

Оплата осуществляется наличными деньгами - монетами и купюрами M ($1 \leq M \leq 20$) допустимых номиналов. Допустим номинал денежных единиц в диапазоне ($1 \leq \text{nominal} \leq 100000$). Если внесена недостаточная для заказа сумма, и покупатель нажал кнопку "Выдать", автомат должен отказать в выдаче товара и проигнорировать это действие. При внесении достаточной для оплаты суммы автомат выдает товары и возвращает рассчитанную сдачу.

Дисплей автомата содержит четыре числовых значения:

- Стоимость – стоимость текущей корзины
- Внесено – сумма денег, внесенных покупателем с момента последней продажи или отмены заказа.
- Осталось внести – разница между стоимостью текущей корзины и внесенной суммой (если эта разница отрицательна - отображается 0)
- Сдача – разница между и внесенной суммой и стоимостью текущей корзины (если эта разница отрицательна - отображается 0)

Внесение денег и добавление товаров в корзину допустимо в любой последовательности (то есть возможно авансовое внесение денег).

Заказ может быть отменен покупателем путем нажатия кнопки «Отмена». В этом случае автомат должен вернуть покупателю внесенную сумму и очистить корзину.

Если в запасе автомата не хватает монет или купюр нужного номинала для того, чтобы выдать точную сумму сдачи – заказ должен быть отменен. То есть это равносильно нажатию кнопки «Отмена».

При этом ваш алгоритм должен стремиться получить максимальную выручку, стараясь при возможности избегать отмены заказов из-за нехватки монет/купюр для сдачи. В данной задаче считается, что автомат может «заглядывать в будущее» - то есть он заранее знает на каком шаге, какие заказы будут выполнены, и используя это знание, может спланировать оптимальное использование монет. Однако запрещено обманывать покупателя, отменяя заказ в том случае, если точная выдача сдачи возможна.

Если какой-то из товаров закончился, но покупатель пытается его заказать, автомат должен просто игнорировать это действие (то есть отобразить на дисплее то же самое, что было на предыдущем шаге).

Покупатели могут пытаться расплатиться фальшивыми монетами или купюрами. Фальшивыми признаются денежные единицы неизвестного номинала. Допустимые номиналы перечислены во входных данных автомата. Внесение фальшивой денежной единицы должно игнорироваться.

Формат входных данных

На первой строке - три числа через пробел:

`N M K`

`N` - количество видов товаров ($1 \leq N \leq 100$)

`M` - количество допустимых номиналов денежных единиц ($1 \leq M \leq 20$)

`K` - количество операций покупателя ($1 \leq K \leq 100000$)

На следующих **N** строках - информация о товарах в формате:

`id qty price`

`id` - код товара ($1 \leq \text{id} \leq N$)

`qty` - количество штук в остатке ($1 \leq \text{qty} \leq 1000$)

`price` - цена товара ($1 \leq \text{price} \leq 100000$)

На следующих **M** строках - информация о наличных деньгах в формате:

`nominal count`

`nominal` - номинал монеты / купюры ($1 \leq \text{nominal} \leq 100000$)

`count` - количество монет / купюр в автомате ($1 \leq \text{count} \leq 1000$)

На следующих **K** строках - последовательность действий покупателя в формате:

`op arg`

Виды операций **op**:

1 - нажата кнопка "Купить". Аргумент **arg** задает номер товара (от 1 до **N**)

2 - внесена монета или купюра. Аргумент **arg** задает номинал

3 - нажата кнопка "Выдать". Без аргумента.

4 - нажата кнопка "Отмена". Без аргумента.

Все числовые значения - неотрицательные целые.

Формат выходных данных

В качестве выхода должно быть выведено **K** строк (одна строка на каждую операцию).

Выходная строка должна содержать через пробел четыре суммы (соответствуют значениям на дисплее):

стоимость
внесено
осталось внести
сдача

Если операция проигнорирована (отсутствующий товар, выдача при неполной оплате, фальшивая купюра и т.д.) - вывод должен повторять предыдущую строку.

После операций отмены или успешной продажи все четыре значения должны быть равны 0.

Примеры

Входные данные:

5 5 8
1 5 100
2 10 25
3 0 30
4 1 100
5 3 120
1 100
2 50
5 20
10 10
100 1
1 2
1 4
2 100
3
2 10
2 10
2 10
3

Выходные данные:

25 0 25 0
125 0 125 0
125 100 25 0
125 100 25 0
125 110 15 0
125 120 5 0
125 130 0 5
0 0 0 0