В городке Урюполе только один автобусный маршрут, соединяющий вокзал с главной местной достопримечательностью — продуктовым рынком, славящимся на всю округу большим ассортиментом и низкими ценами.

В Урюполь недавно пришел поезд из соседнего городка Крыжопинска, и на автобусной остановке возле вокзала образовалась очередь из N человек, желающих попасть на рынок.

В связи с этим, для развозки пассажиров к остановке собираются подать M автобусов вместимостью D каждый. Известно, что если пронумеровать людей от 1 до N в порядке очереди, то i-й из них при посадке в автобус займет L(i) единиц объема.

Однако автобус — не единственный транспорт в Урюполе: если человек устал ждать в очереди, он может выйти из очереди, сесть на такси и тут же уехать. При этом относительный порядок оставшихся в очереди людей не меняется.

Посадка в автобусы происходит следующим образом. Автобус подъезжает к остановке, открывает переднюю дверь, и в нее заходят люди в порядке очереди. Как только для очередного человека не хватает места, автобус закрывает дверь и уезжает, после чего к остановке подходит следующий автобус (если он есть).

Поскольку зарплата водителя автобуса зависит от количества перевезенных пассажиров, водители хотят знать, какое наибольшее суммарное количество людей из очереди они могут перевезти. Помогите им.

Первая строка содержит число M <= 100

Вторая строка — D <= 300.

Третья строка — N <= 300.

Четвёртая строка — L(1) L(2) … L(N). Все L(i) <= D.

Все входные параметры — натуральные числа.

Необходимо вывести единственное число — искомое количество людей.

**Решение.**

1. Создадим трехмерный массив dp на N \* M \* D элементов, где ячейка (i, j, k) будет хранить максимальное количество людей, среди первых i, которые смогут уехать на j автобусах, последний из которых будет заполнен не более, чем на k единиц объема.
2. Заполним этот массив значениями.  
   Для этого вначале заполним базу динамики. Нужно заполнить результат, только для случая, когда очередь будет состоять из одного человека. Для этого запустим 2 цикла, один из которых будет перебирать количество автобусов, а второй – занятость последнего. Очевидно, что dp будет равно 0 только тогда, когда еще ни один автобус не уехал, и этот первый автобус будет заполнен меньше, нежели будет заполнен, если туда посадить этого человека. А во всех остальных случаях результат динамики будет равен единице.  
   Далее нужно заполнить динамику полностью.  
   Для этого нужно запустить 3 цикла, которые будут перебирать все возможные значения префикса взятых людей (кроме случая, когда взят лишь один человек), количества заполненых автобусов и заполненность последнего.  
   Теперь, при заданных фиксированных значениях (i – префикс людей, j – количество приехавших автобусов, k – заполненность последнего автобуса) нужно найти результат динамики.  
   Результатом динамики будет лучшее значение, которое можно получить, исходя из возможных переходов.  
   В заданном состоянии автобусов мы можем либо взять человека и посадить его на последнее возможное место, либо не взять его (он покинет очередь).  
   Когда человек покидает очередь, состояние очереди изменится на (i – 1, j, k), а значит,   
   dp[i][j][k] может быть равно dp[i – 1][j][k]  
   Также человек может сесть в автобус. Но тут может возьникнуть проблемма в том, что он не поместится в последний автобус. В этом случае мы вынуждены будем его запихнуть в последний уехавший.  
   dp[i][j][k] может быть равно dp[i][j - 1][D – a[i]], где a[i] – размер i-го человека.  
   И, если человек влезает в этот автобус, то  
   dp[i][j][k] может быть равно dp[i][j][k – a[i]]  
   Из всех этих возможных переходов, мы выбераем лучший.
3. После заполнения массива, очевидно, наш ответ будет находится в ячейке dp[N][M][D].

Перефразировав сказанное, мы получаем ответ на задачу, использовав оптимальные ответы на подзадачах. Неочевидным является то, что таким образом мы получим оптимальный ответ на всей задаче.

Предположим, что существует не максимально возможный ответ на подзадачу, являющимся оптимальным.

Тогда пусть ответом на подзадачу будет A, а ответ на задачу – A+1 или А. Но более выгодно использовать ответ B < А. Тогда ответом на задачу будет либо B+1, либо B.

Таким образом ответ не только не улучшится, а еще и, возможно, ухудшится. Именно поэтому выбор максимального ответа на подзадачу является оптимальным.

Так как мы лишь используем два массива (один из которых задан), размерностью N и N \* M \* D, циклами проходясь по обоим полностью, то и оценка памяти и времени будет соответственная: O(N \* M \* D).