**“Автобусы”**

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 https://contest.yandex.ru/testsys/tex/render/TChpKVxsZXFzbGFudHt9RA==.png.

Все входные параметры — натуральные числа.

Необходимо вывести единственное число — искомое количество людей.

**Решение.**

1. Задачу будем решать методом динамического программирования.
2. Создадим трехмерный массив mas. В ячейке mas[i][j][k] будем хранить ответ на подзадачу вида: «Какое максимальное количество людей в промежутке от 1 до j человека можно посадить, если раннее было использовано i автобусов, использовав k единиц объема в i+1 автобусе?»
3. Базой динамики будет mas[0][0][0] = 0.
4. Перебираем все значения j, k для каждого автобуса i. (0 <= i < M; 0 <= j <= N; 0 <= k <= D)
5. Когда происходит переход на новый автобус, то ответ для данной подзадачи можно записать так: mas[i][j][0] = mas[i-1][j][D] для всех j, так как в новый мы пока что посадить никого не можем. (Для i = 0, mas[i][j][0] = 0, так как еще никого в автобусы не садили)

Для остальных подзадач нужно рассмотреть 3 перехода:

* Из mas[i][j][k] в mas[i][j-1][k] – в данном случае j-го пассажира мы не садим в автобус, значит объем наполненности текущего автобуса останется прежним, поэтому делаем переход в состояние когда сажали (j-1)-го пассажира. Данный переход выполняется при j > 0.
* Из mas[i][j][k] в mas[i][j-1][k-L[j]] – в данном случае мы садим j-го пассажира в автобус, значит должны обратиться к состоянию, когда сажали (j-1) -го пассажира и когда в текущем автобусе был занят объем равный объему в данном состоянии без j-го пассажира тоесть k – L[j]. К ответу подзадачи (i, j-1, k-L[j]) нужно прибавить 1, так как мы посадали j-го пассажира в автобус. Данный переход выполняется при j > 0 и k >= L[j].
* Из mas[i][j][k] в mas[i][j][k-1] – в данном случае мы занимаем единицу объема пустотой. Это нужно для того чтобы исключить появление пустот в массиве ответов. Данный переход выполняется при k > 0.

Итого ответом для подзадачи (i,j,k) будет max(mas[i][j-1][k], mas[i][j-1][k-L[j]] + 1, mas[i][j][k-1]).

1. Ответ задачи будет храниться в ячейке mas[M-1][N][D].

Доказательство алгоритма состоит в том, что для каждого состояния (количество уже использованных автобусов, количество рассмотренных людей, количество занятых единиц объема данного автобуса) мы рассматриваем все возможные действия и выбираем максимальный результат среди них.

Время работы алгоритма будет равно времени заполнения массива с ответами на подзадачи, а каждую ячейку мы проходим не более одного раза. Поэтому сложность алгоритма будет равна О(М\*N\*D);

Объем памяти также будет определятся размером массива с ответами, тоесть О(M\*N\*D).