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.

Формат ввода

Первая строка содержит число *M* ($\leqslant 100$).
Вторая строка — *D* ($\leqslant 300$).
Третья строка — *N* ($\leqslant 300$).
Четвёртая строка — *L(1) L(2) … L(N)*. Все $L(i) \leqslant D$.
Все входные параметры — натуральные числа.

Формат вывода

Необходимо вывести единственное число — искомое количество людей.

Решение

Решим задачу методом динамического программирования. Условно упорядочим пассажиров, а также все автобусы и все единицы объема в каждом из них, чтобы четко понимать, какую подзадачу рассматривать в данный момент времени. Пусть $d_{i\ j\ k}$ – максимальное количество человек, которое можно перевезти, если рассматривать первые i человек при этом уже уехало j автобусов (с номерами от 0 до $j-1$), а в j автобусе заполнено k мест. Заметим, что если единица объема занята, то это не означает, что ее занимает человек. Смысл только в том, что если в автобусе вместимостью L занято k мест, то мы не можем добавить туда человека, занимающего больше чем $l-k$ мест. Это условие необходимо, чтобы учесть, что автобусы могут быть заполнены частично.

Переход будем осуществлять таким образом:

$$d_{i\ j\ k} = \max(d_{i-1\ j\ k-L[i]} + 1,\ d_{i-1\ j\ k},\ d_{i\ j\ k-1})$$

Т.е. текущее состояние определяется как максимум из:

- $d_{i-1\ j\ k-L[i]} + 1$ – часть перехода, соответствующая тому, что мы берем текущего пассажира ($L[i]$ – его объем). При этом $i > 0, k \geq L[i]$.
- $d_{i-1\ j\ k}$ – часть перехода, соответствующая тому, что пассажир не садится в автобус и уезжает на такси. При этом $i > 0$.
- $d_{i\ j\ k-1}$ – часть перехода, соответствующая тому, что мы резервируем текущую единицу объема. Эта часть перехода учитывает частично заполненный автобус. Например, в автобус вместимостью 4, можно посадить человека объемом 3, и при этом одно единица объема останется свободной и в случае, если нет людей объемом 3 мы больше никого в этот автобус посадить нельзя. При этом $k > 0$.

Отдельно рассмотрим случай, когда один из автобусов уезжает и прибывает новый. Этой ситуации соответствует $d_{i\ j\ 0}$, которое в свою очередь не отличается от состояния, в котором $j-1$ автобус заполнен полностью (при $j > 0$):

$$d_{i\ j\ 0} = d_{i\ j-1\ D}$$

Базовое состояние – $d_{0\ 0\ 0} = 0$.

Ответом будет $d_{n-1\ m-1\ D}$.

Очевидно, что алгоритм корректен т.е. он в любом случае найдет решение с каким-то количеством (возможно нулевым) перевезенных людей. Докажем, что он оптимален т.е. находит именно максимальное количество пассажиров.

Воспользуемся методом доказательства от противного. В текущее состояние $d_{i\ j\ k}$ есть максимум 3 варианта перехода ($d_{i-1\ j\ k-L[i]}$, $d_{i-1\ j\ k}$, $d_{i\ j\ k-1}$). Обозначим A – состояние с максимальным значением d (т.е. то, которое будет выбирать алгоритм). Предположим, что оно неоптимальное, то есть выбрав другое состояние (обозначим его B) получим в итоге большее количество пассажиров автобусов. Теперь попробуем заменить для нашего текущего состояния предыдущее B на A. Решение все еще корректно т.к. это возможное состояние. Ответ же мы ухудшить не могли т.к. $d_{i\ j\ k}$ мы увеличили, а в дальнейшем переходы не меняли. Т.е. конечный результат либо остался оптимален, либо улучшился. Получили противоречие. Так как доказательство проводилось для любого состояния $d_{i\ j\ k}$, то доказана оптимальность для всех состояний т.е. для алгоритма в целом.

Асимптотика алгоритма:

- $O(N * M * D)$ по времени;
- $O(N * M * D)$ по памяти.