**Задача С. 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 ввода 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**

М – количество автобусов *bus*

D – вместимость автобуса *volume*

N – количество людей *people*

1. Создадим массив *PassVolume* для хранения объема каждого из пассажира.
2. Прочитаем входные данные *bus, volume, people,* а так же в цикле от 0 до *people,* что соответствует количеству людей, прочитаем значение объемов в массив *PassVolume.*
3. Созданы два массива *task* и *TrueOrFalse,* в которых хранятся ответ на подзадачу и посчитан ли ответ на эту подзадачу соответственно. Первый массив типа int, второй типа bool.

Весь алгоритм программы находится в функции *Function,* на вход этой функции подается три значения: *allVolume* (занятый объем в текущем автобусе), *numberBus* (номер автобуса) и *numberPeople* (номер текущего человека в очереди).

Первый запуск функции происходит со значениями 0, 0, 0.

В этой функции мы рассматриваем 3 ситуации. Первая ситуация, содержит в себе проверку есть ли еще автобусы и есть ли еще пассажиры в очереди, если это не так, то мы возвращаем ответ ноль, в обратном случае переходим ко второй ситуации.

Во второй ситуации мы проверяем был ли у нас ответ на эту задачу, то есть обращаемся к массиву *TrueOrFalse,* если значение true, тогда мы просто возвращаем ответ из массива *task,*  в обратном случае переходим к следующей ситуации.

Третий вариант,

* когда пассажир уходит из очереди, тогда все параметры сохраняются, а номер пассажира увеличивается на один и вызывается функция с новыми значениями.
* приезжает следующий автобус, если для этого пассажира нет места, тогда объем обнуляется, номер пассажира не меняется, номер автобуса увеличивается на один и вызывается функция с новыми значениями.
* Если место для пассажира есть, то мы его садим в этот автобус, при этом номер автобуса не меняется, объем увеличивается на объем пассажира, номер пассажира увеличивается на 1 и вызывается функция с новыми значениями.

В этих трех пунктах рассмотрены все варианты событий, функция рекурсивно возвращает максимальный из этих вариантов, заносит этот ответ в массив и ставит флажок в соответствующей клетки булевого массива.

В конечном итоге наш ответ будет записан в переменную Answer и выведен в файл.

Сложность определяется тем, что в ходе алгоритма, наша программа заполняет трехмерный массив ответов, при чем каждую клеточку заполнив лишь раз.

Сложность задачи равна О(D\*N\*M).