

Кофи

СЗЕОН 6 – ВТОРИ РУНД – 50 точки



Иванчо много обича да колекционира еднакви кофи. През свободното си време той понякога пълни кофите си с вода. И днес е такъв ден. Иванчо вече е напълнил кофите си и сега се чуди дали нямаше как да събере същото количество вода в по-малко на брой кофи. Помогнете на Иванчо, като напишете програма *buckets* която да каже на Иванчо в колко най-малко кофи можеше да събере цялото количество вода, което има в момента във всички.

Вход

От първия ред на файла *buckets.in* се въвеждат 2 естествени числа N и V – съответно броят на кофите и вместимостта на една кофа.

Следват N естествени числа – $a_1, a_2, \dots, a_n$ като числото a_i е количеството вода в i -тата кофа.

Забележка: всички кофи имат еднаква вместимост – V

Изход

В изходния файл *buckets.out* отпечатайте минималният брой кофи, които са нужни, за да поберат цялото количество вода.

Ограничения

$$1 \leq N \leq 10000$$

$$1 \leq V \leq 100$$

$$0 \leq a_i \leq V$$

Ограничение за време: 0.7 сек

Ограничение за памет: 256 MB

Забележка: Четенето от файл и печатането във файл става като добавите оператор за това. Можете да използвате оператор `freopen` като включите библиотека `fstream` и добавите следните два реда в началото на `main` функцията си:

```
freopen ("basek.in", "r", stdin);
freopen ("basek.out", "w" , stdout);
```

Примерен тест

Вход (<i>buckets.in</i>)	Изход (<i>buckets.out</i>)
3 3 1 2 3	2

Кофи

СЗЕОН 6 – ВТОРИ РУНД – 50 точки



5 2 1 1 1 1 1	3
---------------------	---

Пояснения на примери

1. Имаме 3 кофи с вместимост 3. Можем, например, да прелеем водата от кофа 1 в кофа 2 и да стигнем до конфигурацията 0 3 3. Вече всички кофи са пълни, и имаме една празна кофа, която не ни е нужна. Следователно ни трябват само 2 кофи за цялата течност.

2. Имаме 5 кофи с вместимост 2. Можем, например, да прелеем кофа 3 в кофа 1 и кофа 4 в кофа 2. Получаваме конфигурацията 2 2 0 0 1. Т.е. ни трябват 3 кофи за да поберем цялата вода.