CS1020E | Lab 11 | Exercise 2

Knapsack

Objectives

The objective of this exercise is to use **recursion** to solve combinatorial problem.

Problem Description

You are very lucky to win a free shopping voucher. You quickly grab the catalog of that shop to check what items are sold in that shop. Intuitively, you want to go back home with the best combination of items that you can get. Unfortunately, you do not know the price of each item; what you know is just the weight of all items.

You want to list down all possible sets of items that can be put into your limited sack, i.e. the total weight of a set of items should not exceed the capacity of your sack, otherwise your sack will tear down and you will get nothing.

You would only **get at most 50% of the marks** if **recursion** is not used, in a **correct** and **meaningful** way, in your program.

Add your code only to the parts of the file indicated. Do not modify any other part of the given code, and do not add new files.

Inputs

The first line of the input contains two integers, N ($1 <= N <= 25$), denoting the number of items in the shop and K ($1 <= K <= 100$), denoting the capacity of your sack. The next line contains N integers, denoting the weights of all N items. The weight of an item does not exceed 500.

Outputs

Output the number of different sets of items that can be put in your sack.

Sample Input 1

```
5 6
3 1 6 4 5
```

Sample Output 1

```
9
```

Explanation 1

There are 9 different sets of items with total weight not exceeding the capacity of your sack:

1. {} → Empty sack → Total weight = 0.
2. {3} → Total weight = 3.
3. {3, 1} → Total weight = 4.
4. {1} → Total weight = 1.
5. {1, 4} → Total weight = 5.
6. {1, 5} → Total weight = 6.
7. {6} → Total weight = 6.
8. {4} → Total weight = 4.
9. {5} → Total weight = 5.

Sample Input 2

```
3 1
4 3 7
```

Sample Output 2

```
1
```

Explanation 2

Empty sack is the only possible way.

Submission

You need to submit your completed **`knapsack.cpp`** to CodeCrunch (https://codecrunch.comp.nus.edu.sg/) before the specified deadline. We will take only your latest submission.

Late submissions will not be accepted. The submission system in CodeCrunch will automatically close at the deadline.