Question - 1
Palindrome Counter

A palindrome is a string that reads the same from the left and from the right. For example, *mom* and *tacocat* are palindromes, as are any single-character strings. Given a string, determine the number of its substrings that are palindromes.

Example
The string is *s = 'tacocat'*. Palindromic substrings are *['t', 'a', 'c', 'o', 'c', 'a', 't', 'coc', 'acoca', 'tacocat']*. There are *10* palindromic substrings.

Function Description
Complete the *countPalindromes* function in the editor.

countPalindromes has the following parameter:
   string s: the string to analyze

Returns:
   int: an integer that represents the number of palindromic substrings in the given string

Constraints
- $1 \le |s| \le 5 \times 10^3$
- *each character of s, s[i] $\in$ {'a'-'z'}.*

▼ **Input Format For Custom Testing**

The first line contains a string, *s*.

▼ **Sample Case 0**

Sample Input

```
STDIN      Function
-----      -----
aaa    →   s = 'aaa'
```

Sample Output

```
6
```

Explanation
There are *6* possible substrings of *s*: *{'a', 'a', 'a', 'aa', 'aa', 'aaa'}*. All of them are palindromes, so return *6*.

▼ **Sample Case 1**

Sample Input

? Help

```
STDIN       Function
-----       -----
abccba   →  s = 'abccba'
```

Sample Output

```
9
```

Explanation

There are *21* possible substrings of *s*, the following *9* of which are palindromes: *{'a', 'a', 'b', 'b', 'c', 'c', 'cc', 'bccb', 'abccba'}*.

▼ **Sample Case 2**

Sample Input

```
STDIN       Function
-----       -----
daata    →  s = 'daata'
```

Sample Output

```
7
```

Explanation

There are *15* possible substrings of *s*, the following *7* of which are palindromes: *{'a', 'a', 'a', 'aa', 'ata', 'd', 't'}*.

Question - 2
Lifting Weights

Ollie is new to the gym and is figuring out the maximum weights she can lift. The maximum capacity of the barbell is given as *maxCapacity*. Each barbell plate has a weight, given by *weights[i]*. Now Ollie has to select as many plates as she can but the total weight of the selected plates should not exceed *maxCapacity*. What is the maximum weight of plates Ollie can add to the barbell?

For example, given barbell plates of weights of *1, 3* and *5 lbs* and a barbell of maximum capacity *7 lbs* - the right plates to insert would be *1* and *5 lbs (1+5 = 6),* thus making the right answer *6.*

Function Description

Complete the *weightCapacity* function in the editor below. The function must return an integer denoting the maximum capacity of items that he can purchase.

weightCapacity has two parameters:
  weights: An array of *n* integers, where the value of each element *weights[i]* is the weight of each plate *i* (where $0 \le i < n$).
  maxCapacity: An integer, the capacity of the barbell.

Constraints
- $1 \le n \le 42$

- $1 \le maxCapacity \le 10^9$
- $1 \le weights[i] \le 10^9$

Locked stub code in the editor reads the following input from stdin and passes it to the function:

The first line contains an integer, *n*, denoting the number of elements in *weights*.
Each line *i* of the *n* subsequent lines contains an integer describing *weights[i]*.
The last line contains an integer, *maxCapacity*, denoting the maximum capacity of the barbell.

▼ Sample Case 0

Sample Input 0

```
STDIN      Function
-----      --------
3          → weights[] size n = 3
1          → weights[] = [ 1, 3, 5 ]
3
5
7          → maxCapacity = 7
```

Sample Output 0

```
6
```

Explanation 0
All the possible combination of items that Ollie can insert are:
{}, {1}, {3}, {5}, {1, 3}, {1, 5}, {3, 5}, and *{1, 3, 5}*.
Out of these combinations, the capacity that can be accommodated is *{1, 5}* making the total weight *6.*

▼ Sample Case 1

Sample Input 1

```
STDIN      Function
-----      --------
4          → weights[] size n = 4
4          → weights[] = [ 4, 8, 5, 9 ]
8
5
9
20         → maxCapacity = 20
```

Sample Output 1

```
18
```

Explanation
All the possible combination of items that Ollie can insert are:
{}, {4}, {8}, {5}, {9}, {4, 8}, {4, 5}, {4, 9}, {8, 5}, {8, 9}, {5, 9}, {4, 8, 5}, {4, 8, 9}, {4, 5, 9}, {8, 5, 9}, {4, 8, 5, 9}.
Out of these combinations, the capacity that can be accommodated is *{4, 5, 9}* making the total weight *18.*