Homework 1
CCC 2015 S1 Zero That Out
https://www.cemc.uwaterloo.ca/contests/computing/2015/stage%201/seniorEn.pdf

Problem Description
Your boss has asked you to add up a sequence of positive numbers to determine how much money your company made last year. Unfortunately, your boss reads out numbers incorrectly from time to time. Fortunately, your boss realizes when an incorrect number is read and says "zero", meaning "ignore the current last number." Unfortunately, your boss can make repeated mistakes, and says "zero" for each mistake. For example, your boss may say "One, three, five, four, zero, zero, seven, zero, zero, six", which means the total is 7 as explained in the following chart:

Boss statement(s)	Current numbers	Explanation
"One, three, five, four"	1, 3, 5, 4	Record the first four numbers.
"zero, zero"	1, 3	Ignore the last two numbers.
"seven"	1, 3, 7	Record the number 7 at the end of our list.
"zero, zero"	1	Ignore the last two numbers.
"six"	1, 6	We have read all numbers, and the total is 7.

At any point, your boss will have said at least as many positive numbers as "zero" statements. If all positive numbers have been ignored, the sum is zero. Write a program that reads the sequence of boss statements and computes the correct sum.

Input Specification
The first line of input contains the integer K ($1 \leq K \leq 100\,000$) which is the number of integers (including "zero") your boss will say. On each of the next K lines, there will either be one integer between 1 and 100 (inclusive), or the integer 0.

Output Specification
The output is one line, containing the integer which is the correct sum of the integers read, taking the "zero" statements into consideration. You can assume that the output will be an integer in the range 0 and 1 000 000 (inclusive).

Sample Input 1
4
3
0
4
0

Output for Sample Input 1
0

Sample Input 2

```
10
1
3
5
4
0
0
7
0
0
6
```

Output for Sample Input 2

```
7
```

Homework 2
CCC 2005 J3 Returning Home
https://cemc.math.uwaterloo.ca/contests/computing/2005/stage1/juniorEn.pdf

Problem Description
Jane's family has just moved to a new city and today is her first day of school. She has a list of instructions for walking from her home to the school. Each instruction describes a turn she must make. For example, the list

R
QUEEN
R
FOURTH
R
SCHOOL

means that she must turn right onto Queen Street, then turn right onto Fourth Street, then finally turn right into the school. Your task is to write a computer program which will create instructions for walking in the opposite direction: from her school to her home. The input and output for your program will be formatted like the samples below. You may assume that Jane's list contains at least two but at most five instructions, and you may assume that each line contains at most 10 characters, all of them capital letters. The last instruction will always be a turn into the "SCHOOL".

Sample Input 1
R
QUEEN
R
FOURTH
R
SCHOOL

Sample Output for Sample Input 1
Turn LEFT onto FOURTH street.
Turn LEFT onto QUEEN street.
Turn LEFT into your HOME.

Sample Input 2
L
MAIN
R
SCHOOL

Sample Output for Sample Input 2
Turn LEFT onto MAIN street.
Turn RIGHT into your HOME.

Homework3 Coins Problem (Recursive)

Given a value N, if we want to make change for N cents, and we have infinite supply of each of S = { S1, S2, .. , Sm} valued coins, how many ways can we make the change? The order of coins doesn't matter.

For example, for N = 4 and S = {1,2,3}, there are four solutions: {1,1,1,1},{1,1,2},{2,2},{1,3}. So output should be 4. For N = 10 and S = {2, 5, 3, 6}, there are five solutions: {2,2,2,2,2}, {2,2,3,3}, {2,2,6}, {2,3,5} and {5,5}. So the output should be 5.

Input 1

4

1 2 3

Output 1

4

Input 2
10
2 5 3 6

Output 2
5

Homework 4

Problem J5: π-day

Problem Description
You may know that March 14 is known as "π-day", since 3.14 (which is the third month and fourteenth day) is a good approximation of π.

Mathematicians celebrate this day by eating pie.

Suppose that you have n pieces of pie, and k people who are lined up for pieces of pie. All n pieces of pie will be given out. Each person will get at least one piece of pie, but mathematicians are a bit greedy at times. So, they always get at least as many of pieces of pie as the person in front of them.

For example, if you have 8 pieces of pie and 4 people in line, you could give out pieces of pie in the following five ways (with the first person in line being the first number in the list): $[1, 1, 1, 5]$, $[1, 1, 2, 4]$, $[1, 1, 3, 3]$, $[1, 2, 2, 3]$, $[2, 2, 2, 2]$.

Notice that if $k = n$, there is only one way to give out the pieces of pie: every person gets exactly one piece. Also, if $k = 1$, there is only one way to give out the pieces of pie: that single person gets all the pieces.

Write a program that determines the number of ways that the pieces of pie can be given out.

Input Specification
The first line of input is the integer number of pieces of pie, n $(1 \leq n \leq 250)$.
The second line of input is the integer k which is the number of people in line $(1 \leq k \leq n)$.

For at least 20% of the marks for this problem, $n \leq 9$. For at least 50% of the marks for this problem, $n \leq 70$. For at least 85% of the marks for this problem, $n \leq 120$.

Output Specification
The output will consist of a single integer which is the number of ways that the pieces of pie can be distributed. The output is guaranteed to be less that 2^{31}.

Sample Input 1
```
8
4
```

Output for Sample Input 1
```
5
```

Sample Input 2
```
6
2
```

Output for Sample Input 2
```
3
```