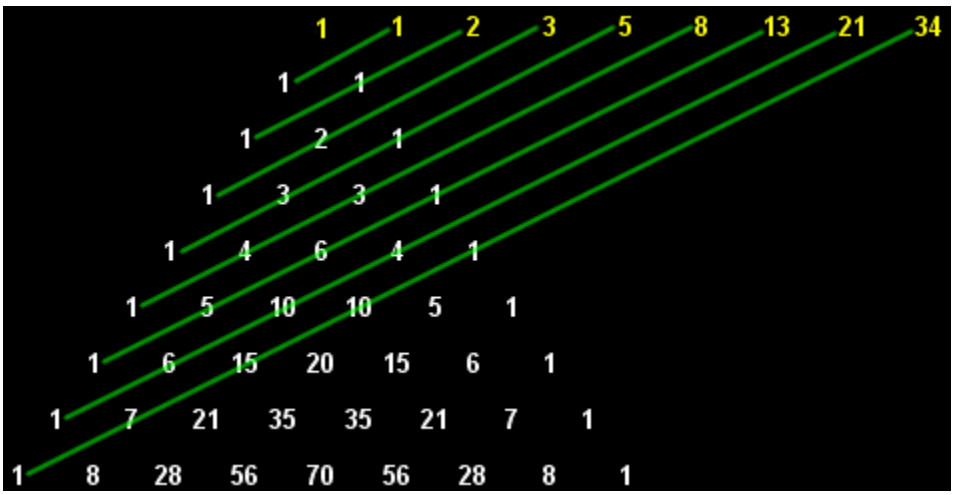
American Computer Science League

2021-2022 ● Contest 3: Fibonacci & Pascal ● Senior Division

PROBLEM: One of the many patterns that can be found in Pascal's Triangle is where the Fibonacci sequence 1, 1, 2, 3, 5, 8, 13, 21, 34, … can be found by adding the numbers on the diagonals as illustrated with the following diagram:



In this program, generate each row of Pascal's Triangle by placing a 1 in the first and last position. For the positions in the middle of the triangle, every number is the sum of the numbers immediately above it. The 0th row contains a single 1. For row N, there are N+1 items.

INPUT: There are 5 lines of data. Each line has a single integer representing one of the Fibonacci numbers in the following sequence: 1 1 2 3 5 8 13 21 34 55 89 .... We guarantee it will be less than 2^{63} since that is the largest integer that can be represented using 64 bits.

OUTPUT: For each line of data, find all of the numbers on that diagonal used to get that sum and any of the previous sums in the triangle. Find all of the integer(s) that occur only once and print how many there are. For example, in the triangle shown, if the input is 8, then the numbers found in this sum and any of the previous ones are {1, 4, 3} for 8, {1, 3, 1} for 5, {1, 2} for 3, {1, 1} for 2, and {1} for 1. The 2 and the 4 occur only once, so 2 is printed.

SAMPLE INPUT:

```
8
89
610
10946
317811
```

SAMPLE OUTPUT:

```
1. 2
2. 8
3. 16
4. 31
5. 58
```

American Computer Science League

2021-2022 ● Contest 3: Fibonacci & Pascal ● Senior Division

TEST DATA

TEST INPUT:

```
55
1597
832040
9227465
1836311903
```

TEST OUTPUT:

1. 6
2. 21
3. 67
4. 96
5. 171