

In Nowhere town, people use “coin and slot” as their money. There are 2 types of coins called size1 and size2. Size2 coin is twice the thickness of size1. People stack their coins in slots and use them as money. There are many size of slots. The slot of size n is able to stack up n size1 coins. Only filled-up slot is considered as legal money. The value of each filled-up slot is the distinct ways its can stack coins inside. For example, for size-1 slot, there is only one way to stack a single size1 coin inside. For size-2 slot, there are 2 ways to stack two size1 coins or one size2 coin inside. And for size-5 slot, there are 8 ways to stack coins inside, which can be illustrated as follow: 1 1 1 1 1, 1 1 1 2, 1 1 2 1, 1 2 1 1, 2 1 1 1, 1 2 2, 2 1 2, 2 2 1. So the value of filled up slot with size-1, size-2 and size-5 are 1, 2 and 8 (monetary) units respectively (regardless of the type of coins or the ways they are stacked).

Mr. Thinktwice is an owner of a grocery store in this town. He noticed that customers are likely to go to the shop that can return the (money) change in the form that suits their customer. And from his little survey, he found that most customers would like to get their amount of change in the form according to these 2 simple constraints.

1. The number of slots is minimum.
2. The size of each slot must be different from each other by at least 2. —This means that customers does not want any slots with the same size and it will be easier for them to distinguish these differences if the sizes are not too close.

So Mr. Thinktwice ask you to write a program that can give him a series of slot sizes for a given amount of change according to the previous constraints. Moreover, the series must be sorted in descending order. For more specific, any amount of change can be written in this formula.

$$\begin{aligned}
 X &= \sum_{i=1}^n T(s_i) \\
 &= T(s_1) + \dots + T(s_i) + \dots + T(s_n), \quad s_1 \gg s_2 \gg \dots \gg s_i \gg \dots \gg s_n > 0
 \end{aligned}$$

where

X is an amount of change.

n is the total number of slot.

s_i is the size of i -th slot.

T is a function mapping from a slot size to a number of distinct ways of stacking coins.

and when $j \gg k \equiv j \geq k + 2$

For example :

$$\begin{aligned}
 10 &= T(5) + T(2) \\
 &= 8 + 2 \\
 1,000,000 &= T(29) + T(25) + T(23) + T(11) + T(9) \\
 &= 832,040 + 121,393 + 46,368 + 144 + 55
 \end{aligned}$$

Input

Input is a standard input which contains a set of integer. Each line of the input is an amount of change which represents by a positive integer less than or equal to 5,000,000,000,000,000,000 or 5×10^{18} . The input is terminated when the EOF (End-Of-File) is reached.

Output

For each amount of change, generate 4 lines of output data. The first line is the amount of change itself. The second line is a series of slot sizes (in descending order) separated by spaces. (The maximum slot size is less than or equal to 90.) The third line is a series of corresponding slot values. The fourth line is a blank line.

Sample Input

```
1
10
1000000
```

Sample Output

```
1
1
1
1

10
5 2
8 2

1000000
29 25 23 11 9
832040 121393 46368 144 55
```