# Problem C. Covering Words with Carrying

| | |
|---|---|
| Input file: | `cover.in` |
| Output file: | `cover.out` |
| Time limit: | 2 seconds |
| Memory limit: | 512 mebibytes |

Consider a word $p$ of length $m$ and a word $s$ of length $n$, $n \geq m$. The word $p$ is said to *cover* $s$ *with carrying* if there is integer $k$ such that $p^k$ is prefix of $ss$ and length of $p^k$ is greater or equal to the length of $s$. Here $p^k$ means the concatenation of $k$ copies of $p$.

You can imagine that the word $s$ is written on a circle and you cover it from the beginning by concatenating copies of $p$, but if the next copy is exceeding the end of $s$, it must continue to cover its beginning.

For example, the word "`01001001`" is covered with carrying by the word "`010`", but the word "`1011011`" is not covered with carrying by a word "`101`", neither is the word "`10110101`".

One word can be covered with carrying by several other words. For example, a word "`11111`" is covered with carrying by any word of length from 1 to 5 that consists of '`1`'-s only. Let us denote as $cc(s)$ the number of words that cover $s$ with carrying. For example, $cc($"`11111`"$) = 5$ and $cc($"`01001001`"$) = 2$.

Consider all words of length $n$ that consist of characters '`0`' and '`1`'. Find the sum of values $cc(s)$ for all such words. Print it modulo 998 244 353.

For example, if $n = 3$, $cc($"`000`"$) + cc($"`001`"$) + cc($"`010`"$) + cc($"`011`"$) + cc($"`100`"$) + cc($"`101`"$) + cc($"`110`"$) + + cc($"`111`"$) = 3 + 1 + 1 + 1 + 1 + 1 + 1 + 3 = 12$.

## Input

The input file contains several test cases. Each test case consists of an integer $n$ on a line ($1 \leq n \leq 1000$). The last test case is followed by a line that contains zero. It must not be processed.

## Output

For each test case output one integer: the sum of $cc(s)$ for all binary words of length $n$, modulo 998 244 353.

## Examples

| cover.in | cover.out |
|---|---|
| 3<br>0 | 12 |