

Problem H. Expression Evaluation

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Expression evaluation is a classic problem. In this problem, you only need to evaluate an expression of length less than 10^5 . It may only contain non-negative integers (possibly with leading zeros), and also operations '+', '-', and '*'. Formally, it satisfies the following grammar:

```
<expression>:= <term>|<expression>+<term>|<expression>-<term>
  <term>:= <number>|<term>*<number>
  <number>:= <digit>|<number><digit>
  <digit>:= 0|1|2|3|4|5|6|7|8|9
```

For example, 013, 0213-2132*0213 are valid, but -2132 and 32113+-3213 are invalid.

The result of the evaluation may be too large or negative. Output the result modulo 2^{32} to avoid overflow since you will use a custom 32-bit machine to evaluate the expression.

The 32-bit machine contains 2^{10} units of memory, denoted as $r[0], r[1], \dots, r[2^{10} - 1]$. Each unit is a 32-bit unsigned integer, also working as an instruction. The machine's input is an expression described above, and the machine's output is the value of that expression modulo 2^{32} .

In each cycle, let $pc = r[0] \bmod 2^{10}$. The machine executes the instruction $r[pc]$. Consider the expansion $r[pc] = a2^{30} + b2^{20} + c2^{10} + d$ at the beginning of cycle, where a, b, c, d are non-negative integers less than 2^{10} :

- If $a = 0$, the machine outputs the value of $r[b]$ and stops.
- If $a = 1$, and then:
 - If there are no characters in input left, set $r[0]$ to $r[d]$;
 - Otherwise, set $r[b]$ to the ASCII code of the next character of input, and then set $r[0]$ to $r[c]$.
- If $a = 2$, set $r[b]$ to $(r[b] + r[c]) \bmod 2^{32}$, and then set $r[0]$ to $r[d]$.
- If $a = 3$, and then:
 - If $r[b] = 0$, set $r[0]$ to the value of $r[c]$;
 - Otherwise, set $r[0]$ to the value of $r[d]$.

Note that if $b = 0$ in some instructions, $r[0]$ may be set more than once. Its value is the value set last after the cycle.

You need to set the initial value for each unit of memory so that, for any expression possible within the constraints, the machine can stop after a finite amount of cycles and output the result of the expression modulo 2^{32} .

Since there is a time limit for testing, in this problem, the machine can execute at most 10^8 cycles.

Input

The only line contains a 32-bit unsigned integer: the seed for the expression generator.

You do not need to use it. It is just for the checker to generate the expressions.

Output

Output 2^{10} 32-bit unsigned integers in the only line: the initial values of the memory units.



Example

standard input	standard output
0	0 0 ... 0 (1024 zeros)

Note

The output in the example is wrong. It is given just to explain the output format.