H. Equivalence classes

Difficulty: Hard **Time**: 2 s **Memory**: 1024 MB

by canin3

Wilbert is practicing for a medal at next year's ICPC World Finals. He knows ICPC problems are usually heavier on implementation and he is already incredible at math, so he should avoid pure math problems. Yet he is suddenly struck by a combinatorial curiosity...

He has a set of n elements $1, ..., n$ which are equivalent to each other at the beginning. Then, k operations occur. In each operation, a subset of the n elements is picked. Then the elements inside the set become different from the elements outside the set. Note that the selected set may be empty.

Since equivalence is bidirectional and transitive, it can be seen that after k operations, the elements can be uniquely partitioned into *equivalence classes*: maximal subsets in which each element is equivalent to every other, i.e. either both chosen or not chosen by each operation.

For each $1 \leq i \leq n$, how many sequences of operations are there that result in exactly i equivalence classes? Two sequences of operations are different if the subset chosen in the i-th operation is different for some i.

Input

The first line contains 2 integers n, k ($1 \leq n \leq 10^4$ and $1 \leq k \leq \min(10^{18}, 2^n)$), the number of elements and operations respectively.

Output

Output n lines.

On the i-th line, print one integer: the number of ways to end up with i equivalence classes, mod 998244353.

Sample 1

Input

```
2 2
```

Output

```
4
12
```

Explanation

When $i = 1$: for both sets, we may choose the empty set or all elements ($2 \cdot 2 = 4$ ways in total). At the end, there's only one equivalence class consisting of all elements.

Since there are $(2^n)^k = 2^4 = 16$ ways to perform all operations and there can only be 1 or 2 equivalence classes when $n = 2$, the number of ways to get $i = 2$ equivalence classes is $16 - 4 = 12$.

Sample 2

Input

```
5 5
```

Output

```
32
14880
744000
8630400
24165120
```