Square-free division (hard version)

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is the hard version of the problem. The only difference is that in this version $0 \leq k \leq 20$.

There is an array $a_1, a_2, \ldots, a_n$ of n positive integers. You should divide it into a minimal number of continuous segments, such that in each segment there are no two numbers (on different positions), whose product is a perfect square.

Moreover, it is allowed to do at most k such operations before the division: choose a number in the array and change its value to any positive integer.

What is the minimum number of continuous segments you should use if you will make changes optimally?

Input

The first line contains a single integer t $(1 \leq t \leq 1000)$ — the number of test cases.

The first line of each test case contains two integers n, k ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq k \leq 20$).

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^7$).

It's guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case print a single integer — the answer to the problem.

Example

input

```
3
5 2
18 6 2 4 1
11 4
6 2 2 8 9 1 3 6 3 9 7
1 0
1
```

output

```
1
2
1
```

Note

In the first test case it is possible to change the array this way: $[\underline{3}, 6, 2, 4, \underline{5}]$ (changed elements are underlined). After that the array does not need to be divided, so the answer is 1.

In the second test case it is possible to change the array this way: $[6, 2, \underline{3}, 8, 9, \underline{5}, 3, 6, \underline{10}, \underline{11}, 7]$. After that such division is optimal:

- $[6, 2, 3]$
- $[8, 9, 5, 3, 6, 10, 11, 7]$

2021-03-17 19:05:48