Codeforces Round #213 (Div. 1)

A. Matrix

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have a string of decimal digits s. Let's define $b_{ij} = s_i \cdot s_j$. Find in matrix b the number of such rectangles that the sum b_{ij} for all cells (i, j) that are the elements of the rectangle equals a in each rectangle.

A rectangle in a matrix is a group of four integers (x, y, z, t) **$(x \le y, z \le t)$**. The elements of the rectangle are all cells (i, j) such that $x \le i \le y, z \le j \le t$.

Input

The first line contains integer a ($0 \le a \le 10^9$), the second line contains a string of decimal integers s ($1 \le |s| \le 4000$).

Output

Print a single integer — the answer to a problem.

Please, do not write the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Examples

input
10 12345
output
6

input
16 439873893693495623498263984765
output
40

B. Free Market

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

John Doe has recently found a "Free Market" in his city — that is the place where you can exchange some of your possessions for other things for free.

John knows that his city has n items in total (each item is unique). You can bring any number of items to the market and exchange them for any other one. Note that each item is one of a kind and that means that you cannot exchange set $\{a, b\}$ for set $\{v, a\}$. However, you can always exchange set x for any set y, unless there is item p, such that p occurs in x and p occurs in y.

For each item, John knows its value c_i. John's sense of justice doesn't let him exchange a set of items x for a set of items y, if $s(x) + d < s(y)$ ($s(x)$ is the total price of items in the set x).

During one day John can exchange only one set of items for something else. Initially, he has no items. John wants to get a set of items with the maximum total price. Find the cost of such set and the minimum number of days John can get it in.

Input

The first line contains two space-separated integers n, d ($1 \le n \le 50$, $1 \le d \le 10^4$) — the number of items on the market and John's sense of justice value, correspondingly. The second line contains n space-separated integers c_i ($1 \le c_i \le 10^4$).

Output

Print two space-separated integers: the maximum possible price in the set of items John can get and the minimum number of days needed to get such set.

Examples

input
3 2 1 3 10
output
4 3

input
3 5 1 2 3
output
6 2

input
10 10000 10000 9999 1 10000 10000 10000 1 2 3 4
output
50010 6

Note

In the first sample John can act like this:

- Take the first item ($1 - 0 \le 2$).
- Exchange the first item for the second one ($3 - 1 \le 2$).
- Take the first item ($1 - 0 \le 2$).

C. Beautiful Set

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

We'll call a set of positive integers a beautiful if the following condition fulfills: for any prime p, if $\exists x \in a, x \equiv 0 \mod p$, then $|\{y \in a | y \equiv 0 \mod p\}| \geq \frac{|a|}{2}$. In other words, if one number from the set is divisible by prime p, then at least half of numbers from the set is divisible by p.

Your task is to find any beautiful set, where the number of elements is equal to k and each element doesn't exceed $2k^2$.

Input

The first line contains integer k ($10 \leq k \leq 5000$) that shows how many numbers the required beautiful set should have.

Output

In the first line print k space-separated integers that are a beautiful set. If there are multiple such sets, you are allowed to print any of them.

Examples

input
10
output
16 18 24 27 36 48 54 72 108 144

D. Ghd

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

John Doe offered his sister Jane Doe find the gcd of some set of numbers a.

Gcd is a positive integer g, such that all number from the set are evenly divisible by g and there isn't such g' ($g' > g$), that all numbers of the set are evenly divisible by g'.

Unfortunately Jane couldn't cope with the task and John offered her to find the ghd of the same subset of numbers.

Ghd is a positive integer g, such that at least half of numbers from the set are evenly divisible by g and there isn't such g' ($g' > g$) that at least half of the numbers from the set are evenly divisible by g'.

Jane coped with the task for two hours. Please try it, too.

Input

The first line contains an integer n ($1 \le n \le 10^6$) showing how many numbers are in set a. The second line contains space-separated integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^{12}$). Please note, that given set can contain **equal** numbers.

Please, do not write the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `%I64d` specifier.

Output

Print a single integer g — the Ghd of set a.

Examples

input
6 6 2 3 4 5 6
output
3

input
5 5 5 6 10 15
output
5

E. Empty Rectangles

time limit per test: 12 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You've got an $n \times m$ table (n rows and m columns), each cell of the table contains a "0" or a "1".

Your task is to calculate the number of rectangles with the sides that are parallel to the sides of the table and go along the cell borders, such that the number one occurs exactly k times in the rectangle.

Input

The first line contains three space-separated integers n, m and k ($1 \le n, m \le 2500$, $0 \le k \le 6$) — the sizes of the table and the required number of numbers one.

Next n lines each contains m characters "0" or "1". The i-th character of the j-th line corresponds to the character that is in the j-th row and the i-th column of the table.

Output

Print a single number — the number of rectangles that contain exactly k numbers one.

Please, do not write the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Examples

input
3 3 2 101 000 101
output
8

input
5 5 1 00000 00000 00100 00000 00000
output
81

input
5 5 6 01010 10101 01010 10101 01010
output
12

input
3 3 0 001 010 000
output
15

input
4 4 0 0000 0101 0000 0000
output

```
52
```