Factor y Hell

Factorial(N) in base B : The number of trailing zeros.

Factorial(**19**) in base **9**×10^**0**=9 can be written 725735500635080**000**, ending with **4** zeros.
Factorial(**43**) in base **2**×10^**1**=20 can be written 59HHHFECFCCEGH5G7I7A3A8G88F8CD8G**000000000**, ending with **9** zeros.
What about working with serious constraints and tricky cases ?
Factorial(**N**) will be a huge one, the base will be dummy too and have the special form : **B**×10^**E**.

Input

The input begins with the number T of test cases in a single line.
In each of the next T lines there are three integers : N, B, E.

Output

For each test case, print the number of zeros at the end of Factorial(N) written in base B×10^E.

Example

Input:
3
19 9 0
43 2 1
10000 100 10

Output:
4
9
208

Constraints

1 <= T < 2000
1 <= N < 10^1000
1 <= B < 10^9
0 <= E < 10^9

Informations

Don't worry about the 'special' base 1 (B=1 and E=0), it is absent from input.
About distribution : random input (N : log-uniform, B : uniform, E : uniform) in their range. Some tricky cases are added.
It is recommended to solve FACTBASE first, and find a way to solve FCTRL much faster than common solutions.
Time limit is ×12 my best Python3 time, or ×1.2 my "basic" one.