Educational Codeforces Round 70 (Rated for Div. 2)

A. You Are Given Two Binary Strings...

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

You are given two binary strings x and y, which are binary representations of some two integers (let's denote these integers as $f(x)$ and $f(y)$). You can choose any integer $k \geq 0$, calculate the expression $s_k = f(x) + f(y) \cdot 2^k$ and write the binary representation of s_k in **reverse order** (let's denote it as rev_k). For example, let $x = 1010$ and $y = 11$; you've chosen $k = 1$ and, since $2^1 = 10_2$, so $s_k = 1010_2 + 11_2 \cdot 10_2 = 10000_2$ and $rev_k = 00001$.

For given x and y, you need to choose such k that rev_k is **lexicographically minimal** (read notes if you don't know what does "lexicographically" means).

It's guaranteed that, with given constraints, k exists and is finite.

Input

The first line contains a single integer T ($1 \leq T \leq 100$) — the number of queries.

Next $2T$ lines contain a description of queries: two lines per query. The first line contains one binary string x, consisting of no more than 10^5 characters. Each character is either `0` or `1`.

The second line contains one binary string y, consisting of no more than 10^5 characters. Each character is either `0` or `1`.

It's guaranteed, that $1 \leq f(y) \leq f(x)$ (where $f(x)$ is the integer represented by x, and $f(y)$ is the integer represented by y), both representations don't have any leading zeroes, the total length of x over all queries doesn't exceed 10^5, and the total length of y over all queries doesn't exceed 10^5.

Output

Print T integers (one per query). For each query print such k that rev_k is lexicographically minimal.

Example

input
4 1010 11 10001 110 1 1 1010101010101 11110000
output
1 3 0 0

Note

The first query was described in the legend.

In the second query, it's optimal to choose $k = 3$. The $2^3 = 1000_2$ so $s_3 = 10001_2 + 110_2 \cdot 1000_2 = 10001 + 110000 = 1000001$ and $rev_3 = 1000001$. For example, if $k = 0$, then $s_0 = 10111$ and $rev_0 = 11101$, but $rev_3 = 1000001$ is lexicographically smaller than $rev_0 = 11101$.

In the third query $s_0 = 10$ and $rev_0 = 01$. For example, $s_2 = 101$ and $rev_2 = 101$. And 01 is lexicographically smaller than 101.

The quote from Wikipedia: "To determine which of two strings of characters comes when arranging in *lexicographical order*, their first letters are compared. If they differ, then the string whose first letter comes earlier in the alphabet comes before the other string. If the first letters are the same, then the second letters are compared, and so on. If a position is reached where one string has no more letters to compare while the other does, then the first (shorter) string is deemed to come first in alphabetical order."

B. You Are Given a Decimal String...

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

output: standard output

Suppose you have a special x-y-counter. This counter can store some value as a decimal number; at first, the counter has value 0.

The counter performs the following algorithm: it prints its lowest digit and, after that, adds either x or y to its value. So all sequences this counter generates are starting from 0. For example, a 4-2-counter can act as follows:

1. it prints 0, and adds 4 to its value, so the current value is 4, and the output is 0;
2. it prints 4, and adds 4 to its value, so the current value is 8, and the output is 04;
3. it prints 8, and adds 4 to its value, so the current value is 12, and the output is 048;
4. it prints 2, and adds 2 to its value, so the current value is 14, and the output is 0482;
5. it prints 4, and adds 4 to its value, so the current value is 18, and the output is 04824.

This is only one of the possible outputs; for example, the same counter could generate 0246802468024 as the output, if we chose to add 2 during each step.

You wrote down a printed sequence from one of such x-y-counters. But the sequence was corrupted and several elements from the sequence could be erased.

Now you'd like to recover data you've lost, but you don't even know the type of the counter you used. You have a decimal string s — the remaining data of the sequence.

For all $0 \le x, y < 10$, calculate the minimum number of digits you have to insert in the string s to make it a possible output of the x-y-counter. Note that you can't change the order of digits in string s or erase any of them; only insertions are allowed.

Input
The first line contains a single string s ($1 \le |s| \le 2 \cdot 10^6$, $s_i \in \{0 - 9\}$) — the remaining data you have. It's guaranteed that $s_1 = 0$.

Output
Print a 10×10 matrix, where the j-th integer (0-indexed) on the i-th line (0-indexed too) is equal to the minimum number of digits you have to insert in the string s to make it a possible output of the i-j-counter, or -1 if there is no way to do so.

Example

input
0840
output
-1 17 7 7 7 -1 2 17 2 7 17 17 7 5 5 5 2 7 2 7 7 7 7 4 3 7 1 7 2 5 7 5 4 7 3 3 2 5 2 3 7 5 3 3 7 7 1 7 2 7 -1 5 7 3 7 -1 2 9 2 7 2 2 1 2 1 2 2 2 0 1 17 7 7 5 7 9 2 17 2 3 2 2 2 2 2 2 0 2 2 2 7 7 5 3 7 7 1 3 2 7

Note
Let's take, for example, 4-3-counter. One of the possible outcomes the counter could print is $0(4)8(1)4(7)0$ (lost elements are in the brackets).

One of the possible outcomes a 2-3-counter could print is $0(35)8(1)4(7)0$.

The 6-8-counter could print exactly the string 0840.

C. You Are Given a WASD-string...

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

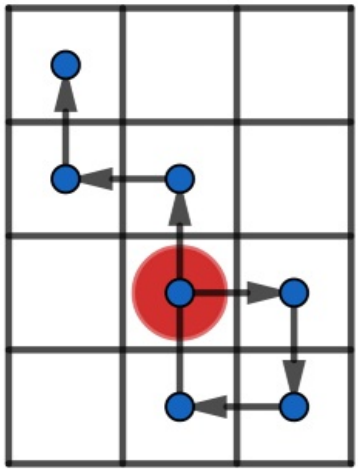
You have a string s — a sequence of commands for your toy robot. The robot is placed in some cell of a **rectangular** grid. He can perform four commands:

- 'W' — move one cell up;
- 'S' — move one cell down;
- 'A' — move one cell left;
- 'D' — move one cell right.

Let $Grid(s)$ be the grid of minimum possible area such that there is a position in the grid where you can place the robot in such a way that it will not fall from the grid while running the sequence of commands s. For example, if s = DSAWWAW then $Grid(s)$ is the 4×3 grid:

1. you can place the robot in the cell $(3, 2)$;
2. the robot performs the command 'D' and moves to $(3, 3)$;
3. the robot performs the command 'S' and moves to $(4, 3)$;

4. the robot performs the command 'A' and moves to $(4, 2)$;
5. the robot performs the command 'W' and moves to $(3, 2)$;
6. the robot performs the command 'W' and moves to $(2, 2)$;
7. the robot performs the command 'A' and moves to $(2, 1)$;
8. the robot performs the command 'W' and moves to $(1, 1)$.



You have 4 extra letters: one 'W', one 'A', one 'S', one 'D'. You'd like to insert **at most one of these letters** in any position of sequence s to minimize the area of $Grid(s)$.

What is the minimum area of $Grid(s)$ you can achieve?

Input
The first line contains one integer T ($1 \le T \le 1000$) — the number of queries.

Next T lines contain queries: one per line. This line contains single string s ($1 \le |s| \le 2 \cdot 10^5$, $s_i \in \{\text{W}, \text{A}, \text{S}, \text{D}\}$) — the sequence of commands.

It's guaranteed that the total length of s over all queries doesn't exceed $2 \cdot 10^5$.

Output
Print T integers: one per query. For each query print the minimum area of $Grid(s)$ you can achieve.

Example

input
3 DSAWWAW D WA
output
8 2 4

Note
In the first query you have to get string $\text{DSAWW}\underline{D}\text{AW}$.

In second and third queries you can not decrease the area of $Grid(s)$.

D. Print a 1337-string...

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

The subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements.

You are given an integer n.

You have to find a sequence s consisting of digits $\{1, 3, 7\}$ such that it has exactly n subsequences equal to 1337.

For example, sequence 337133377 has 6 subsequences equal to 1337:

1. $33\underline{7}1\underline{33}\underline{3}7\underline{7}$ (you can remove the second and fifth characters);
2. $33\underline{7}1\underline{3}\underline{33}7\underline{7}$ (you can remove the third and fifth characters);

3. 3371337 (you can remove the fourth and fifth characters);
4. 3371337 (you can remove the second and sixth characters);
5. 3371337 (you can remove the third and sixth characters);
6. 3371337 (you can remove the fourth and sixth characters).

Note that the length of the sequence s must not exceed 10^5.

You have to answer t independent queries.

Input
The first line contains one integer t ($1 \le t \le 10$) — the number of queries.

Next t lines contains a description of queries: the i-th line contains one integer n_i ($1 \le n_i \le 10^9$).

Output
For the i-th query print one string s_i ($1 \le |s_i| \le 10^5$) consisting of digits $\{1, 3, 7\}$. String s_i must have exactly n_i subsequences 1337. If there are multiple such strings, print any of them.

Example

input
2 6 1
output
113337 1337

E. You Are Given Some Strings...

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a string t and n strings $s_1, s_2, \ldots, s_n$. All strings consist of lowercase Latin letters.

Let $f(t, s)$ be the number of occurences of string s in string t. For example, $f(\text{'aaabacaa'}, \text{'aa'}) = 3$, and $f(\text{'ababa'}, \text{'aba'}) = 2$.

Calculate the value of $\sum_{i=1}^{n} \sum_{j=1}^{n} f(t, s_i + s_j)$, where $s + t$ is the concatenation of strings s and t. Note that if there are two pairs i_1, j_1 and i_2, j_2 such that $s_{i_1} + s_{j_1} = s_{i_2} + s_{j_2}$, you should include both $f(t, s_{i_1} + s_{j_1})$ and $f(t, s_{i_2} + s_{j_2})$ in answer.

Input
The first line contains string t ($1 \le |t| \le 2 \cdot 10^5$).

The second line contains integer n ($1 \le n \le 2 \cdot 10^5$).

Each of next n lines contains string s_i ($1 \le |s_i| \le 2 \cdot 10^5$).

It is guaranteed that $\sum_{i=1}^{n} |s_i| \le 2 \cdot 10^5$. All strings consist of lowercase English letters.

Output
Print one integer — the value of $\sum_{i=1}^{n} \sum_{j=1}^{n} f(t, s_i + s_j)$.

Examples

input
aaabacaa 2 a aa
output
5

input
aaabacaa 4 a a a

b
output
33

F. You Are Given Some Letters...

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

You are given a uppercase Latin letters 'A' and b letters 'B'.

The period of the string is the smallest such positive integer k that $s_i = s_{i \bmod k}$ (0-indexed) for each i. Note that this implies that k won't always divide $a + b = |s|$.

For example, the period of string "ABAABAA" is 3, the period of "AAAA" is 1, and the period of "AABBB" is 5.

Find the number of different periods over all possible strings with a letters 'A' and b letters 'B'.

Input
The first line contains two integers a and b ($1 \le a, b \le 10^9$) — the number of letters 'A' and 'B', respectively.

Output
Print the number of different periods over all possible strings with a letters 'A' and b letters 'B'.

Examples

input
2 4
output
4

input
5 3
output
5

Note
All the possible periods for the first example:

- 3 "BBABBA"
- 4 "BBAABB"
- 5 "BBBAAB"
- 6 "AABBBB"

All the possible periods for the second example:

- 3 "BAABAABA"
- 5 "BAABABAA"
- 6 "BABAAABA"
- 7 "BAABAAAB"
- 8 "AAAAABBB"

Note that these are not the only possible strings for the given periods.

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
The only programming contests Web 2.0 platform

Processing math: 100%