

Codeforces Round #146 (Div. 2)**A. Boy or Girl**

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Those days, many boys use beautiful girls' photos as avatars in forums. So it is pretty hard to tell the gender of a user at the first glance. Last year, our hero went to a forum and had a nice chat with a beauty (he thought so). After that they talked very often and eventually they became a couple in the network.

But yesterday, he came to see "her" in the real world and found out "she" is actually a very strong man! Our hero is very sad and he is too tired to love again now. So he came up with a way to recognize users' genders by their user names.

This is his method: if the number of distinct characters in one's user name is odd, then he is a male, otherwise she is a female. You are given the string that denotes the user name, please help our hero to determine the gender of this user by his method.

Input

The first line contains a non-empty string, that contains only lowercase English letters — the user name. This string contains at most 100 letters.

Output

If it is a female by our hero's method, print "CHAT WITH HER!" (without the quotes), otherwise, print "IGNORE HIM!" (without the quotes).

Sample test(s)

input
wjmbzmr
output
CHAT WITH HER!

input
xiaodao
output
IGNORE HIM!

input
sevenkplus
output
CHAT WITH HER!

Note

For the first example. There are 6 distinct characters in "wjmbzmr". These characters are: "w", "j", "m", "z", "b", "r". So wjmbzmr is a female and you should print "CHAT WITH HER!".

B. Easy Number Challenge

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

Let's denote $d(n)$ as the number of divisors of a positive integer n . You are given three integers a, b and c . Your task is to calculate the following sum:

$$\sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^c d(i \cdot j \cdot k).$$

Find the sum modulo 1073741824 (2^{30}).

Input

The first line contains three space-separated integers a, b and c ($1 \leq a, b, c \leq 100$).

Output

Print a single integer — the required sum modulo 1073741824 (2^{30}).

Sample test(s)

input
2 2 2
output
20

input
5 6 7
output
1520

Note

For the first example.

- $d(1 \cdot 1 \cdot 1) = d(1) = 1$;
- $d(1 \cdot 1 \cdot 2) = d(2) = 2$;
- $d(1 \cdot 2 \cdot 1) = d(2) = 2$;
- $d(1 \cdot 2 \cdot 2) = d(4) = 3$;
- $d(2 \cdot 1 \cdot 1) = d(2) = 2$;
- $d(2 \cdot 1 \cdot 2) = d(4) = 3$;
- $d(2 \cdot 2 \cdot 1) = d(4) = 3$;
- $d(2 \cdot 2 \cdot 2) = d(8) = 4$.

So the result is $1 + 2 + 2 + 3 + 2 + 3 + 3 + 4 = 20$.

C. LCM Challenge

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Some days ago, I learned the concept of LCM (least common multiple). I've played with it for several times and I want to make a big number with it.

But I also don't want to use many numbers, so I'll choose three positive integers (they don't have to be distinct) which are not greater than n . Can you help me to find the maximum possible least common multiple of these three integers?

Input

The first line contains an integer n ($1 \leq n \leq 10^6$) — the n mentioned in the statement.

Output

Print a single integer — the maximum possible LCM of three not necessarily distinct positive integers that are not greater than n .

Sample test(s)

input
9
output
504

input
7
output
210

Note

The least common multiple of some positive integers is the least positive integer which is multiple for each of them.

The result may become very large, 32-bit integer won't be enough. So using 64-bit integers is recommended.

For the last example, we can chose numbers 7, 6, 5 and the LCM of them is $7 \cdot 6 \cdot 5 = 210$. It is the maximum value we can get.

D. Let's Play Osu!

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

You're playing a game called Osu! Here's a simplified version of it. There are n clicks in a game. For each click there are two outcomes: correct or bad. Let us denote correct as "O", bad as "X", then the whole play can be encoded as a sequence of n characters "O" and "X".

Using the play sequence you can calculate the score for the play as follows: for every maximal consecutive "O"s block, add the square of its length (the number of characters "O") to the score. For example, if your play can be encoded as "O O X O O O X X O O", then there's three maximal consecutive "O"s block "OO", "OOO", "OO", so your score will be $2^2 + 3^2 + 2^2 = 17$. If there are no correct clicks in a play then the score for the play equals to 0.

You know that the probability to click the i -th ($1 \leq i \leq n$) click correctly is p_i . In other words, the i -th character in the play sequence has p_i probability to be "O", $1 - p_i$ to be "X". Your task is to calculate the expected score for your play.

Input

The first line contains an integer n ($1 \leq n \leq 10^5$) — the number of clicks. The second line contains n space-separated real numbers p_1, p_2, \dots, p_n ($0 \leq p_i \leq 1$).

There will be at most six digits after the decimal point in the given p_i .

Output

Print a single real number — the expected score for your play. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6} .

Sample test(s)

input
3 0.5 0.5 0.5
output
2.7500000000000000
input
4 0.7 0.2 0.1 0.9
output
2.4892000000000000
input
5 1 1 1 1 1
output
25.0000000000000000

Note

For the first example. There are 8 possible outcomes. Each has a probability of 0.125.

- "OOO" $\rightarrow 3^2 = 9$;
- "OOX" $\rightarrow 2^2 = 4$;
- "OXO" $\rightarrow 1^2 + 1^2 = 2$;
- "OXX" $\rightarrow 1^2 = 1$;
- "XOO" $\rightarrow 2^2 = 4$;
- "XOX" $\rightarrow 1^2 = 1$;
- "XXO" $\rightarrow 1^2 = 1$;
- "XXX" $\rightarrow 0$.

So the expected score is $\frac{9+4+2+1+4+1+1}{8} = 2.75$

E. Cyclical Quest

time limit per test: 3 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Some days ago, WJMZBMR learned how to answer the query "how many times does a string x occur in a string s " quickly by preprocessing the string s . But now he wants to make it harder.

So he wants to ask "how many consecutive substrings of s are cyclical isomorphic to a given string x ". You are given string s and n strings x_i , for each string x_i find, how many consecutive substrings of s are cyclical isomorphic to x_i .

Two strings are called *cyclical isomorphic* if one can rotate one string to get the other one. 'Rotate' here means 'to take some consecutive chars (maybe none) from the beginning of a string and put them back at the end of the string in the same order'. For example, string "abcde" can be rotated to string "deabc". We can take characters "abc" from the beginning and put them at the end of "de".

Input

The first line contains a non-empty string s . The length of string s is not greater than 10^6 characters.

The second line contains an integer n ($1 \leq n \leq 10^5$) — the number of queries. Then n lines follow: the i -th line contains the string x_i — the string for the i -th query. The total length of x_i is less than or equal to 10^6 characters.

In this problem, strings only consist of lowercase English letters.

Output

For each query x_i print a single integer that shows how many consecutive substrings of s are cyclical isomorphic to x_i . Print the answers to the queries in the order they are given in the input.

Sample test(s)

input
baabaabaaa 5 a ba baa aabaa aaba
output
7 5 7 3 5

input
aabbbaa 3 aa aabb abba
output
2 3 3