**Card Game**

Vova again tries to play some computer card game.

The rules of deck creation in this game are simple. Vova is given an existing deck of n cards and a magic number k. The order of the cards in the deck is fixed. Each card has a number written on it; number ai is written on the i-th card in the deck.

After receiving the deck and the magic number, Vova removes x (possibly x = 0) cards from the top of the deck, y (possibly y = 0) cards from the bottom of the deck, and the rest of the deck is his new deck (Vova has to leave at least one card in the deck after removing cards). So Vova's new deck actually contains cards x + 1, x + 2, ... n - y - 1, n - y from the original deck.

Vova's new deck is considered valid iff the product of all numbers written on the cards in his new deck is divisible by k. So Vova received a deck (possibly not a valid one) and a number k, and now he wonders, how many ways are there to choose x and y so the deck he will get after removing x cards from the top and y cards from the bottom is valid?

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n,k,lef,rig,last,i;

int a[100005];

long long ans=0;

cin >> n >> k;

lef=1,rig=1,last=0;

for(i=1;i<=n;i++) cin >> a[i];

while(rig<=n)

{

long long p=1;

for(i=lef;i<=n;i++)

{

p\*=a[i];

p%=(long long)k;

if(p==0)break;

}

rig=i;

p=1;

for(i=rig;i>=1;i--)

{

p\*=a[i];

p%=(long long)k;

if(p==0)break;

}

lef=i;

ans+=(long long)(lef-last)\*(long long)(n-rig+1);

last=lef;

lef++;

rig++;

}

cout << ans;

return 0;

}