Suppose two numbers(a and b) are prime factorized  
  
a = 2^3 \* 3^4 \* 5^2  
  
b = 2^324154654564654 \* 3^ 4654654564 \* 5^1654654654564 \*7^56465456  
  
how could i know is b%a==0 ??

2.১০০০ এর ফেক্টেরিয়াল কে প্রাইম ফেক্টোরাইজেশন করলে তাতে কয়টা ২ আছে তা ১০০০ এর ফেক্টেরিয়াল না বের করে কিভাবে বের করা যায় । যেমন ৪(চার) এর ফেক্টেরিয়াল করলে হয় ২৪। ২৪ কে প্রাইম ফেক্টোরাইজেশন করলে হয় ২৪= ২\*২\*২\*৩ । Number of two is 3 . ৪(চার) এর ফেক্টেরিয়াল এবং ২৪এর প্রাইম ফেক্টোরাইজেশন না বের করে কিভাবে এটা বের করা যায় ? সরাসরি ৪ থেকে এটা বের করার কোন technic আছে কি ?

Solution: 100! এ দুই কতবার আছে টা বের করার উপায় হলঃ  
  
100/2+100/4+100/8+100/16+100/32+100/64  
=50+25+12+6+3+1  
=97

Code :

h = 0;

while(N!=0){

h = h + N/p;

N = N / p;

}

h holds the highest power of p which can divide N!.

3. কিভাবে ১২৩৪৫৬^৪৫৫ এই সংখ্যাটার প্রথম ৫ টি digits বের করা যায় ? অথাৎ কিভাবে n^k (1000<n<=10^12 and 1000<k<=10^12) এই সংখ্যাটা প্রথম ৫ টি digits বের করা যায় ? কোন technic **থাকলে শেয়ার করবেন প্লীজ।:)**

**Solution: For trailing digits solution:** use bigmod function,if u need last d digits choose mod value as 10^d.

***For leading digits solution:*** Take the log of the value. log (n^k) = k \* log (n). Suppose the value of k \* log (n) is xx.yyyy. To get the original value we have to power it with base ten. That is, original vaule is 10^(xx.yyy). We can write 10^(xx.yyy) = 10^xx \* 10 ^0.yyy. Notice, 10^xx only provides trailing zero to the number. To the leading number comes from 10^0.yyy. So just multiply 10^.yyy with 100000 to get the first 5 digit.

4. We know what a base of a number is and what the properties are. For example, we use decimal number system, where the base is **10** and we use the symbols - **{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}**. But in different bases we use different symbols. For example in binary number system we use only **0** and **1**. Now in this problem, you are given an integer. You can convert it to any base you want to. But the condition is that if you convert it to any base then the number in that base should have at least one trailing zero that means a zero at the end.

For example, in decimal number system **2** doesn't have any trailing zero. But if we convert it to binary then **2** becomes **(10)2** and it contains a trailing zero. Now you are given this task. You have to find the number of bases where the given number contains at least one trailing zero. You can use any base from two to infinite.

***Solution:*** Just find the number of divisor of the given number this is answer .

5. ইউলার টসিয়েন্ট জানতাম কিন্তু সবচেয়ে ফাস্ট টসিয়েন্ট ফাংশন ব্যবহার করে-ও আমি TLE খাইছি। তারপর মেমোরাইজেশন ব্যবহার করলাম। খাইলাম মেমোরি লিমিট। শেষ মেশ ইউলার ফাই টেবিল জেনারেট করে দেখলাম। তখন AC হইছে। For Euler Phi table the code is given below:

#include <iostream>

#include <stdio.h>

#define MAX 5000005

#define MIN -2147483647

using namespace std;

typedef unsigned long long int ll;

int phi[5000001];

ll table[5000001];

void phi\_function()

{ phi[1] = 1;

table[1]=0;

for(int i=2; i<=5000001; ++i)

{

if(phi[i]==0)

{

phi[i]=i-1;

for(int j=i+i; j<=5000001; j+=i)

{

if(phi[j]==0)

phi[j] = j;

phi[j]= phi[j]/i \* (i-1);

}

}

}

//for( int i=1; i<=45; i++)

//cout<<"i="<<i<<" "<<phi[i]<<" "<<endl;

for(int i=2; i<5000001; i++)

{

table[i]=phi[i];

table[i]\*=phi[i];

table[i]+=table[i-1];

}

}

int main()

{ phi\_function();

int T,Case=0;

scanf("%d",&T);

while(T--)

{ int a,b;

scanf("%d %d",&a,&b);

printf("Case %d: %llu\n",++Case,table[b]-table[a-1]);

}

}