

<https://codeforces.com/problemset>

chenjiuri_fsfsfsjinzhifjsdkfjsdklfjsd;kljfuoi

Inflation has occurred in Berlandia, so the store needs to change the price of goods.

The current price of good n is given. It is allowed to increase the price of the good by k times, with $1 \leq k \leq m$, k is an integer. Output the roundest possible new price of the good. That is, the one that has the maximum number of zeros at the end.

For example, the number 481000 is more round than the number 1000010 (three zeros at the end of 481000 and only one at the end of 1000010).

- 简介
 - 给定 a 和 b 。
 - 对 a 进行一个乘法，变成倍数的操作。其中倍数选取范围是 $1-b$ ，
 - 求出一个前缀0最多，且最大的解。
- solve
 - 首先关注一个数中因数对答案的影响。
 - 其中一个因数2，一个因数5，对后缀0有一个贡献。
 - 发现为了使得后缀0最多，本身 a 的0个数不受个数的影响。于是关注当前的多出的2或者5的因数的个数。
 - 最有操作是，在乘数中，分配相关的补充的因数。
 - 分完之后，看有没有剩余的可以补10。
 - 现在已经找到最大的0的个数的最小解。 t
 - 对 t 乘某一个倍数，不影响缀0的个数，影响前缀的有效数字的个数。全部相乘即可。

code

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const int maxn = 2e5 + 10;

void solve()
{
    ll a, b;
    cin >> a >> b;
    ll x = a, y = b;
    int c2 = 0, c5 = 0;
```

```

while (x % 5 == 0)
    c5++, x /= 5;
while (x % 2 == 0)
    c2++, x /= 2;
//求出了两种质因子的情况。
ll t = 1; //现在开始累积。
while (t * 2 <= b && c2 < c5) //现在对5进行一个配对。
{
    t *= 2;
    c2++;
}
while (t * 5 <= b && c2 > c5)
{
    t *= 5;
    c5++;
}
//当前5的情况已经讨论完成
while (t * 10 <= b)
{
    c2++, c5++;
    t *= 10;
}
//当前的情况。
// if (c2 || c5) //有什么特殊情况？什么情况下直接乘？
cout << b / t * t * a << '\n';
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(nullptr), cout.tie(nullptr);
    int t;
    cin >> t;
    while (t--)
        solve();
}

```

生长思考：

- 关注问题特性
 - 解的结构形成，和乘法相关，自然可以联想到因数的问题。
 - 关注的解的结构的某些鲜明的属性：素数，某个数的倍数，某个数的最小公倍数，质因数个数情况。等等。