20180409_京东三道题

输入一个数n，求能被1~n 内整除的最小整数。网上查的解法是n 以内的所有素数相乘。

```
#include <stdio.h>
#include <memory.h>
#include <math.h>
#include <iostream>
#include <vector>

using namespace std;

long long find(long long n){
    long long i =0;
    n = n>>1;
    while (n >0){
        n = n >> 1;
        ++i;
    }

    return  pow(2,i);
}
long long  maxmi(long long n,long long i);

void Primes (long long  n, vector<long long>& vi) {
    if (n == 1){
        return;
    }

    vi.push_back(2);
    if(n == 2)
        return;

    for (long long  i = 3; i <= n; i += 2) { //偶数不是质数，步长可以加大
        //float temp = static_cast<float>(i);
        long long mid = static_cast<int>(sqrt(i));
        long long j;
        for (j = 3; j <= mid; j += 2)//i是奇数，当然不能被偶数整除，步长也可以加大。
            if (i % j == 0)
                break;

        if (j > mid){
            vi.push_back(maxmi(n,i));
        }
    }

}

long long  maxmi(long long n,long long i){
    long long t = log(n)/log(i);
    return pow(i,t) ;

}

long long  Division (long long  n) {
    vector<long long > vi;
    Primes(n, vi);


    long long min = 1;
    for (long long  i=0; i<vi.size(); ++i) {
        if( vi[i] == 2 )
            min = min*find(n);
        else
            min *= vi[i];

    }

    if(min >= 987654321)
        min = min % 987654321;
    return min;
}

int main(void)
{
    long long n ;
    cin >> n;

    int min = Division(n);

    cout  << min << endl;
    return 0;
}
```

这个题上面的还不对，应该是每个素数的最大次幂

https://blog.csdn.net/cillyb/article/details/75008137

```
#include<iostream>
#include<cstdio>
#include<cstring>
```

```
using namespace std;
typedef long long ll;
const int maxn = 100000004;
const int mod = 1e8+7;
int n, cnt, p[6200000];
bool vis[maxn];

int main(void)
{
    scanf("%d", &n);
    ll ans = 1;
    for(int i = 2; i <= n; i++)
    {
        if(!vis[i])
        {
            p[cnt++] = i;
            for(ll s = i; s <= n; s *= i)
                ans = ans*i%mod;
        }
        for(int j = 0; j < cnt; j++)
        {
            ll v = i*p[j];
            if(v > n) break;
            vis[v] = 1;
            if(i%p[j] == 0) break;
        }
    }
    printf("%lld\n", ans);
    return 0;
}
```

第二题是求回文序列的

第三题，是象棋棋谱，给定步骤数，求马能走到指定位置有多少种情况。