

1 guess

东西不多，一道例题

```
//langman
#include <bits/stdc++.h>
using namespace std;

#define clr(shu,x) memset(shu,x,sizeof(shu))
#define INF 0x3f3f3f3f
#define pi acos(-1)
#define loge exp(1)
#define ll long long
#define pb push_back
#define ios_close ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL)
const int mod = 1e9+7;
const double eps = 1e-6;
char shu[20][20];
double dp[230]; // 这个表示的是现在这个点到终点的期望
double op[230][230];
int n,m;
int cal(int i,int j) //算出一维坐标
{
    return i*m+j;
}
int vis[20][20];
int Hash[20][20];
bool check(int i,int j)
{
    if(i<0 || j <0 || i>=n || j>=m || shu[i][j] == '#') return false;
    return true;
}
int num(int i,int j)
{
    int tot = 0;
    if(check(i+1,j)) tot++;
    if(check(i,j+1)) tot++;
    if(check(i-1,j)) tot++;
    if(check(i,j-1)) tot++;
    return tot;
}
bool tag;
int flag = 0;
void dfs(int x,int y) // 判可行
{
    if(shu[x][y] == '$') tag = 1;
    if(Hash[x][y] == -1) Hash[x][y] = flag++;
    vis[x][y] = 1;
    for(int dx = -1;dx<=1;dx++)
    {
```

```

for(int dy = -1;dy <= 1;dy++)
{
    if(dx == dy || dx+dy == 0) continue;
    if(check(x+dx,y+dy) && vis[x+dx][y+dy]==0)

        dfs(x+dx,y+dy);
}
}
}
void build()
{
    for(int i = 0;i<n;i++)
    {
        for(int j = 0;j<m;j++)
        {
            if(Hash[i][j] == -1)
            {
                continue;
            }
            if(shu[i][j] == '$')
            {
                op[Hash[i][j]][Hash[i][j]] = 1;
                dp[Hash[i][j]] = 0;
                continue;
            }
            else
            {
                if(check(i+1,j)) op[Hash[i][j]][Hash[i+1][j]] = -1.0;
                if(check(i-1,j)) op[Hash[i][j]][Hash[i-1][j]] = -1;
                if(check(i,j+1)) op[Hash[i][j]][Hash[i][j+1]] = -1;
                if(check(i,j-1)) op[Hash[i][j]][Hash[i][j-1]] = -1;
                op[Hash[i][j]][Hash[i][j]] = num(i,j);
                dp[Hash[i][j]] = num(i,j);
            }
        }
    }
}
int var,equ;
int Gauss()//Gauss消元求解
{
    var = equ = flag;
    for(int k = 0, col = 0; k < equ && col < var; k++, col++)
    {
        int max_r = k;
        for(int i = k + 1; i < equ; i++)
            if(fabs(op[i][col]) > fabs(op[max_r][col]))
                max_r = i;
        if(fabs(op[max_r][col]) < eps) return 0;
        if(k != max_r)

```

```

    {
        for(int j = col; j < var; j++)
            swap(op[k][j], op[max_r][j]);
        swap(dp[k], dp[max_r]);
    }
    dp[k] /= op[k][col];
    for(int j = col + 1; j < var; j++) op[k][j] /= op[k][col];
    op[k][col] = 1;
    for(int i = 0; i < equ; i++)
        if(i != k)
        {
            dp[i] -= dp[k]*op[i][col];
            for(int j = col + 1; j < var; j++) op[i][j] -= op[k][j]*op[i][col];
            op[i][col] = 0;
        }
    }
    return 1;
}
int main()
{
    // ios_close;
    //freopen("in.txt", "r", stdin);
    //freopen("out.txt", "w", stdout);
    while(scanf("%d%d", &n, &m) == 2)
    {
        flag = 0;
        clr(dp, 0);
        clr(op, 0);
        clr(vis, 0);
        clr(Hash, -1);
        int a, b;
        int x, y;
        tag = 0;
        for(int i = 0; i < n; i++)
        {
            scanf("%s", shu[i]);
        }
        for(int i = 0; i < n; i++)
        {
            for(int j = 0; j < m; j++)
            {
                if(shu[i][j] == '@') x = i, y = j, b = cal(i, j);
            }
        }
        dfs(x, y);
        build();
        // cout<<flag<<endl;
        // for(int i = 0; i < flag; i++)
        // {

```

```

// for(int j = 0; j < flag; j++)
// {
//     cout << op[i][j] << " ";
// }
// cout << dp[i] << endl;
// }
if(tag == 0)
{
    printf("-1\n");
    continue;
}
Gauss();
printf("%.6lf\n", dp[Hash[x][y]]);
}
return 0;
}

```

2 shai

欧拉函数前缀和

```

#include <cstdio>
#include <cstring>
#include <algorithm>
#include <cmath>
#include <vector>
#define MAXN 500010
#define MAXP 500010
#define MOD 1000000007
#define SF scanf
#define PF printf
using namespace std;
bool isprime[MAXN];
int primes[MAXN], cnt;
long long fi[MAXN];
vector<pair<long long, long long> > ans[MAXN+10];
void prepare(){
    fi[1]=1;
    for(int i=2; i<=MAXN-10; i++){
        fi[i]=MOD;
        if(isprime[i]==0){
            primes[++cnt]=i;
            fi[i]=i-1;
        }
        for(int j=1; j<=cnt && i*primes[j]<=MAXN-10; j++){
            isprime[i*primes[j]]=1;
            if(i%primes[j]==0){
                fi[i*primes[j]]=fi[i]*primes[j];
                break;
            }
        }
    }
}

```

```

        }
        fi[i*primes[j]]=fi[i]*fi[primes[j]];
    }
}
for(int i=2;i<=MAXN-10;i++)
    fi[i]=(fi[i-1]+fi[i])%MOD;
}
long long get_num(long long n){
    long long px=n%MAXP;
    for(int i=0;i<ans[px].size();i++)
        if(ans[px][i].first==n)
            return ans[px][i].second;
    return -1;
}
void push_num(long long n,long long res){
    long long px=n%MAXP;
    ans[px].push_back(make_pair(n,res));
}
long long sum(long long n){
    if(n<=MAXN-10)
        return fi[n];
    long long res=get_num(n);
    if(res>=0)
        return res;
    if(n%2==0)
        res=((n/2)%MOD)*((n+11)%MOD)%MOD;
    else
        res=((n%MOD)*((n+11)/2)%MOD)%MOD;
    for(long long p=2;p<=n;){
        long long len=n/(n/p)+11-p;
        res=(res-len*sum(n/p))%MOD;
        if(res<0)
            res+=MOD;
        p+=len;
    }
    push_num(n,res);
    return res;
}
int main(){
    prepare();
    long long n;
    /*for(int i=1;i<=10000;i++)
        if(fi[i]!=sum(i)){
            PF("Error!%d: %d %d\n",i,fi[i],sum(i));
            break;
        }*/
    SF("%lld",&n);
    PF("%lld",sum(n));
}

```

莫比乌斯函数前缀和

```
#include <cstdio>
#include <cstring>
#include <algorithm>
#include <cmath>
#include <vector>
#include <map>
#define SF scanf
#define PF printf
#define MAXN 5000010
#define MAXP 2181271
using namespace std;
bool isprime[MAXN], found;
int primes[MAXN], cnt, mu[MAXN];
map<long long, int> ans;
/*vector<pair<long long, long long> > ans[MAXP+10];
long long get_num(long long x){
    found=1;
    long long px=x%MAXP;
    for(int i=0; i<ans[px].size(); i++)
        if(ans[px][i].first==x)
            return ans[px][i].second;
    found=0;
    return 0;
}
void push_num(long long x, long long val){
    long long px=x%MAXP;
    ans[px].push_back(make_pair(x, val));
}*/
void prepare(){
    mu[1]=1;
    for(int i=2; i<=MAXN-10; i++){
        if(isprime[i]==0){
            mu[i]=-1;
            primes[++cnt]=i;
        }
        for(int j=1; j<=cnt&&primes[j]*i<=MAXN-10; j++){
            isprime[i*primes[j]]=1;
            if(i%primes[j]==0){
                mu[i*primes[j]]=0;
                break;
            }
            mu[i*primes[j]]=-mu[i];
        }
    }
    for(int i=1; i<=MAXN-10; i++)
        mu[i]=mu[i-1]+mu[i];
}
long long l, r;
```

```

long long sum(long long n){
    if (n<=MAXN-10)
        return mu[n];
    if (ans.count(n)!=0)
        return ans[n];
    /*long long res=get_num(n);
    if(found==1)
        return res;*/
    long long res=1ll;
    for(long long q=2;q<=n;){
        long long len=n/(n/q)+1ll-q;
        res-=sum(n/q)*len;
        q+=len;
    }
    ans[n]=res;
    //push_num(n,res);
    return res;
}
int main(){
    prepare();
    SF("%lld%lld",&l,&r);
    PF("%lld",sum(r)-sum(l-1));
}

```

3 min25 筛

$$f(x) = 1[x == 1]$$

$$f(x) = x^2 + 2x - 1[xisprime]$$

$$f(x) = -3[xisp^k p isprime]$$

```

#include <bits/stdc++.h>
using namespace std;
const int N = 1e9 + 5;
typedef long long ll;
const ll mod = 1e9 + 7;
const int MAX = 1e5 + 5;
ll prime[MAX];
ll spg[10000];
ll sph[10000];
ll spi[10000];
ll w[MAX];
ll id1[MAX];
ll id2[MAX];
ll g[MAX], h[MAX], ii[MAX];
int tot;
bool vis[MAX];
ll Pow(ll a, int b)
{

```

```

ll ans = 1;
while (b)
{
    if (b & 1)
    {
        ans = ans * a % mod;
    }
    a = a * a % mod;
    b >>= 1;
}
return ans;
}
void pre(int n)
{
    memset(vis, 0, sizeof(vis));
    vis[1] = 1;
    prime[tot = 0] = 1;
    sph[0] = spg[0] = spi[0] = 0;
    for (int i = 2; i <= n; i++)
    {
        if (!vis[i])
        {
            prime[++tot] = i;
            spg[tot] = (spg[tot - 1] + i * i % mod) % mod;
            sph[tot] = (sph[tot - 1] + 2 * i) % mod;
            spi[tot] = spi[tot - 1] + 1;
        }
        for (int j = 1; j <= tot && prime[j] * i <= n; j++)
        {
            vis[i * prime[j]] = 1;
            if (i % prime[j] == 0)
                break;
        }
    }
}

ll sqr, n, m;

ll S(ll x, int y)
{
    if (x <= 1 || prime[y] > x)
    {
        return 0;
    }
    //printf("%lld %d:\n", x, y);
    int k = (x <= sqr) ? id1[x] : id2[n / x];
    ll ret = (g[k] - spg[y - 1] + h[k] - sph[y - 1] - (ii[k] - spi[y - 1]) + mod) % mod;
    //cout << ret << endl;
    for (int i = y; i <= tot && 1ll * prime[i] * prime[i] <= x; i++)

```



```

{
    ll t1 = prime[i], t2 = 1ll * prime[i] * prime[i];
    for (int e = 1; t2 <= x; e++, t1 = t2, t2 *= prime[i])
    {
        if (e == 1)
        {
            ret = (ret + 1ll * S(x / t1, i + 1) * (prime[i] * prime[i] % mod + 2 * prime[i] %
            continue;
        }
        ret = (ret - 1ll * S(x / t1, i + 1) * 3 % mod - 3 + mod) % mod;
    }
}
//printf("%lld\n", ret);
return ret;
}

void solve()
{
    m = 0;
    sqr = sqrt(n);
    pre(sqr);
    int l, r;
    ll inv6 = Pow(6, mod - 2);
    for (l = 1; l <= n; l = r + 1)
    {
        r = n / (n / l);
        w[++m] = n / l;
        g[m] = w[m] * (w[m] + 1) % mod * (2 * w[m] + 1) % mod * inv6 % mod; //二次项
        g[m] = (g[m] - 1 + mod) % mod; //减去1
        h[m] = w[m] * (w[m] + 1) % mod;
        h[m] = (h[m] - 2 + mod) % mod;
        ii[m] = w[m] - 1;
        if (n / l <= sqr)
        {
            id1[w[m]] = m;
        }
        else
        {
            id2[r] = m;
        }
    }
    for (int j = 1; j <= tot; j++)
    {
        for (int i = 1; i <= m && prime[j] * prime[j] <= w[i]; i++)
        {
            int k = (w[i] / prime[j] <= sqr) ? id1[w[i] / prime[j]] : id2[n / (w[i] / prime[j])];
            g[i] = (g[i] - 1ll * prime[j] * prime[j] % mod * (g[k] - spg[j - 1]) + mod) % mod;
            h[i] = (h[i] - 1ll * prime[j] * (h[k] - sph[j - 1]) + mod) % mod;
            ii[i] = (ii[i] - (ii[k] - spi[j - 1]) + mod) % mod;
        }
    }
}

```

```

    }
}
ll ans = S(n, 1) + 1;
printf("%lld\n", ans);
}
int main()
{
    while (scanf("%lld", &n) == 1)
    {
        solve();
    }
    return 0;
}

```

4 meissel lehmer

用来算n里面有多少个素数，n可以很大。

```

#include <bits/stdc++.h>
using namespace std;
typedef long long ll;

const ll MAXN = 10000000;

ll N;
bool isp[MAXN];
int pcnt[MAXN];
vector<int> primes;
void init()
{
    for(int i=2; i<MAXN; i++){
        isp[i] = true;
    }

    for(int i=2; i<MAXN; i++)
    {
        if(!isp[i]) continue;
        primes.push_back(i);
        for(int j=2*i; j<MAXN; j+=i){
            isp[j] = false;
        }
    }

    pcnt[0] = 0;
    for(int i=1; i<MAXN; i++){
        pcnt[i] = pcnt[i-1] + isp[i];
    }
}

```

```

11 prime_count(11 lim);

11 phi(11 m, int n)
{
    if(m <= 0) return 0;
    if(n == 0) return m;
    if(primes[n-1] * primes[n-1] >= m) return prime_count(m) - n + 1;
    return phi(m, n-1) - phi(m / primes[n-1], n-1);
}

11 prime_count(11 lim)
{
    if(lim < MAXN) return pcnt[lim];

    11 m3 = 1, m2 = 1;
    while(m3*m3*m3<=lim) m3++;
    while(m2*m2<=lim) m2++;
    m3--;
    m2--;

    11 y = m3;
    11 n = prime_count(y);
    11 p2 = 0;
    for(11 p=y+1; p<=m2; p++)
    {
        if(isp[p]){
            p2 += prime_count(lim / p) - prime_count(p) + 1;
        }
    }

    11 ph = phi(lim, n);
    11 res = ph + n - 1 - p2;
    //cout<<lim<<" y "<<y<<" n "<<n<<" p2 "<<p2<<" phi "<<ph<<endl;
    return res;
}

int main()
{
    init();
    while(~scanf("%lld",&N))
    {
        11 ans = prime_count(N);
        printf("%lld\n",ans);
    }
    return 0;
}

```