Phi(n),Miu(n)

```cpp
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=2e6;
const int mod=7+1e9;
int inv2=0;
ll qpow(ll a,ll b,ll p){a%=p;ll ret=1;for(;b;b>>=1){if(b&1)
ret=ret*a%mod;a=a*a%mod;}return ret;}
ll add(ll x,ll y){
    x%=mod;
    y%=mod;
    x+=y;
    if(x>=mod) x-=mod;
    return x;
}
ll sub(ll x,ll y){
    x%=mod;
    y%=mod;
    x-=y;
    if(x<0) x+=mod;
    return x;
}
ll mul(ll x,ll y){
    x%=mod;
    y%=mod;
    x*=y;
    if(x>=mod)
        x%=mod;
    return x;
}
struct EulerSeive{
    int sz;
    vector<bool> vis;
    vector<int> pri;
    vector<ll>phi,miu;
    EulerSeive(int n):sz(n),vis(n,0),phi(n,0),miu(n,0){init();}
    inline void init(){
        vis[0]=vis[1]=1;
        phi[1]=miu[1]=1;
        for(int i=2;i<sz;i++){
            if(!vis[i]){
                vis[i]=1;
                pri.push_back(i);
                phi[i]=i-1;
                miu[i]=-1;
            }
            for(int j=0;j<(int)pri.size()&&i*pri[j]<sz;j++){
                vis[i*pri[j]]=1;
                if(i%pri[j]==0){
                    phi[i*pri[j]]=phi[i]*pri[j];
                    break;
                }
```

```cpp
                phi[i*pri[j]]=phi[i]*phi[pri[j]];
                miu[i*pri[j]]=-miu[i];
            }
        }
        for(int i=2;i<sz;i++){
            phi[i]=add(phi[i],phi[i-1]);
            miu[i]=add(miu[i],miu[i-1]);
        }
    }
}es(N);
unordered_map<ll,ll> p,mu;
pair<ll,ll> getPhiMiu(ll x){
    if(x<N) return {es.phi[x],es.miu[x]};
    if(p[x]) return {p[x],mu[x]};
    ll ret1=mul(mul(x,x+1),inv2),ret2=1;
    for(ll l=2,r;l<=x;l=r+1){
        r=x/(x/l);
        pair<ll,ll> res=getPhiMiu(x/l);
        ret1=sub(ret1,(r-l+1)*res.first);
        ret2=sub(ret2,(r-l+1)*res.second);
    }
    return {p[x]=ret1,mu[x]=ret2};
}

int main(){
    ll n; scanf("%lld",&n);
    inv2=qpow(2,mod-2,mod);
    for(int i=1;i<=6;i++) printf("%lld ",es.phi[i]); puts("");
    printf("%lld\n",getPhiMiu(n).first);
}
```