

```


#include<bits/stdc++.h>
using namespace std;
const int N=1e7+5;
int p[N]; // primes
int phi[N]; // Euler's totient function
int mu[N]; // Möbius function
int d[N]; // number of divisors
int mi[N]; // sum of divisors
int sigma[N]; // sum of divisors
int mp[N]; // sum of divisors of squares
bool b[N]; // sieve

int main(){
// freopen("num.txt", "w", stdout);
b[0]=b[1]=phi[1]=mu[1]=d[1]=mi[1]=sigma[1]=mp[1]=1;
for(int i=2;i<=N-5;i++){
    if(!b[i]){
        p[++p[0]]=i; phi[i]=i-1; mu[i]=-1; d[i]=2;
        mi[i]=1; sigma[i]=1+i; mp[i]=1+i;
    }
    for(int j=1;j<=p[0]&&i*p[j]<=N-5;j++){
        b[i*p[j]]=1;
        if(i%p[j]==0){
            phi[i*p[j]]=phi[i]*p[j];
            mu[i*p[j]]=0;
            d[i*p[j]]=d[i]/(mi[i]+1)*(mi[i]+2);
            mi[i*p[j]]=mi[i]+1;
            sigma[i*p[j]]=sigma[i]/mp[i]*(mp[i]*p[j]+1);
            mp[i*p[j]]=mp[i]*p[j]+1;
            break;
        }
        else{
            phi[i*p[j]]=phi[i]*phi[p[j]];
            mu[i*p[j]]=-mu[i];
            d[i*p[j]]=d[i]*d[p[j]];
            mi[i*p[j]]=1;
            sigma[i*p[j]]=sigma[i]*sigma[p[j]];
            mp[i*p[j]]=p[j]+1;
        }
    }
}
/*
int n;
scanf("%d",&n);
for(int i=2;i<=n;i++){
    printf("%d %d %d %d %d %d\n",i,b[i]?0:1,phi[i],mu[i],d[i],sigma[i]);
}
*/
return 0;

```

}

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:%E7%BA%BF%E6%80%A7%E7%AD%9B_lgwza&rev=1594210667 

Last update: **2020/07/08 20:17**