

```

//mengegut9.cpp
#include <iostream>
#include <conio.h>//getche
#include <cmath>//pow
using namespace std;

const int MAX = 654;
const int MAX2 = 33;

int gerade[MAX];
int geradeNat[MAX];
int positiv[MAX];
int negativ[MAX];
int nat[MAX];
int prim[MAX];
int fibo[MAX2];

bool istEnthalten(int*,int);
void pfz(int);

int main()
{
    const unsigned char ue = static_cast<unsigned char>(129);//ue
    //Einfache Mengen berechnen:
    for(int z=0; z<MAX; z++)
    {
        nat[z]=z;//natuerliche mit Null
        geradeNat[z]= z*2;//gerade natuerliche
        positiv[z]=z+1;
        negativ[z]=-z-1;
        gerade[z]=z*2-MAX;//gerade, also auch negative gerade
    }//for z
    //Primzahlen berechnen:
    unsigned int limit=4900;
    bool *Stroke = new bool[limit+1];
    Stroke[0]=1;
    Stroke[1]=1;
    for(unsigned i=2; i<=limit; ++i) Stroke[i] = 0;
    unsigned int prime=2;
    while(pow((double)prime,2.0)<=limit){
        for(unsigned int i = (int)pow((double)prime,2.0); i<=limit;
i+=prime)
            Stroke[i]=1;
        do ++prime; while(Stroke[prime]);
    }
}

```

```

} //while
int p = 0;
    for(unsigned int i=0; i<=limit; ++i){
        if(!Stroke[i]) prim[p++]=i; //p ist anders als i.
    }
delete[] Stroke;
//Fibonacci-Zahlen berechnen:
int i = 0;
int low = 1;
    int high = 1;
    fibo[i++]=low;
    while (high <= 4000000)
    {
        fibo[i++]=high;
        high = low + high; //neues high
        low = high - low; //neues low
    } //while
//
//for(int y=0; y<MAX2; y++) cout << fibo[y] << " " << y << " # ";
//cout<<endl;
//
cout << "Nat" << ue << "rlich, ganz oder auch Primzahl ..." << endl;
cout << "-----" << endl;
cout << "Dieses kleine Programm macht Aussagen wie:" << endl;
cout << "13 ist eine Primzahl. Und mehr ..." << endl;
cout << "Erforsche es doch selbst!" << endl;
cout << endl;
char janein='a';
do
{
    cout << endl;
    cout << "Gib eine ganze Zahl ein (-652 bis 652): ";
    int zahl; cin >> zahl;
    cout << endl << endl;
    cout << "    Ja                Nein" << endl;
    bool bo = istEnthalten(nat, zahl);
    cout << (bo?"": " ") << zahl << " ist " <<
(bo?"": "k") <<
    "eine nat" << ue << "rliche Zahl (mit 0; No)." << endl;
    //
    bool b2 = istEnthalten(gerade, zahl);
    cout << (b2?"": " ") << zahl << " ist " <<
(b2?"": "k") <<
    "eine gerade Zahl." << endl;
}

```



```
bool istEnthalten(int menge[], int zahl)
{
    for(int a=0; a<MAX; a++)
        if (zahl==menge[a]) return true;
    return false;
} //Funktion istEnthalten()
```

```
void pfz(int z)
{
    cout << "Primfaktorzerlegung:" << endl;
    for(int i=0; i<MAX; i++)
    {
        while(z/prim[i] == (float)z/prim[i])
        {
            cout << z << " ist teilbar durch " << prim[i] << endl;
            z /= prim[i];
        } //while
    } //for
} //funktion pfz()
```