

```

/*Программа ищет простые числа вида  $2 \cdot 7^m - 1$  используя библиотеку
gmp и написана на языке c++.
Программа работает только в системах поддерживающих сигналы
UNIX.*/
#include <iostream>
#include <fstream>
#include <signal.h>
#include <gmp.h>
#include <gmpxx.h>
#include <ctime>
using namespace std;

//Функция нахождения числа a по модулю b
mpz_class mod(const mpz_class a, const mpz_class b)
{
    mpz_class c = a % b;
    if (c >= 0)
        return c;
    else
        return b + c;
}

//Функция возведения числа q в степень n по модулю m
mpz_class pw(mpz_class q, mpz_class n, mpz_class m)
{
    mpz_class qn = q, l = n;
    q = 1;
    if (m >= 0) {
        while (l != 0) {
            if (l % 2 == 1)
                q = mod((q * qn), m);
            l /= 2;
            qn = mod((qn * qn), m);
        }
        return mod(q, m);
    } else {
        while (l != 0) {
            if (l % 2 == 1)
                q = q * qn;
            l /= 2;
            qn = qn * qn;
        }
        return q;
    }
}

fstream mfile;
mpz_class m1 = 0;
char* s;

//Обработчик сигналов
void handler(int signo)

```

```

{
    mfile.close();
    mfile.open(s, ios::out | ios::trunc);
    mfile<<m1-1;
    mfile.close();
    raise(SIGKILL);
}

int main(int argc, char** argv)
try {
    if(argc < 2)
        throw "Bad command line!";
    s = argv[2];
    mfile.open(s, ios::in);
    ofstream outfile;
    long int tm = time(0);
    mpz_class v = 42, u = 1, num = 1, v2, u2, v3, v4, u4, vd = 42,
ud = 1, m2 = 9000, n = 1, ppowm = 7, q = -7, qn, N = 1;
    mfile>>m1;
    signal(SIGINT, handler);
    signal(SIGTERM, handler);
    if ((m1 <= 0) || !mfile) {
        if (!mfile)
            m1 = 1;
        if (mfile) cout<<s<<"\n";
        outfile.open(argv[1], ios::out | ios::trunc);
        outfile<<"Номер#m#Время#Тип#Число знаков"<<endl;
    } else
        outfile.open(argv[1], ios::out | ios::app);
    ppowm = pw(7,m1,-1);
    mpz_t rop;
    mpz_init(rop);
    while(m1 != m2) {
        tm = time(0);
        n = 2 * ppowm - 1;
        vd = v = 42;
        ud = u = 1;
        num = 1;
        //Нахождение последовательностей U и V
        for (mpz_class i = 0; i < m1 - 1; ++i) {
            qn = pw(q,num,n);
            u = mod(u * mod(v * v * mod(v * v * mod(v * v - 5 *
qn, n) + 6 * qn * qn, n) - qn * qn * qn, n), n);
            v = mod(v * mod(v * v * mod(v * v * mod(v * v - 7 *
qn, n) + 14 * qn * qn,n) - 7 * qn * qn * qn, n), n);
            num *= 7;
        }
        if (v != 0) {
            mpz_class uc = u, vc = v;
            qn = pw(q,num,n);
            u = mod((u * v),n);
            v = mod((v * v - 2 * qn),n);
        }
    }
}

```

```

    qn *= qn;
    mpz_gcd(rop,u.get_mpz_t(),n.get_mpz_t());
    if (mpz_class(rop) == 1) {
        u = mod(u * mod(v * v * mod(v * v * mod(v * v - 5
* qn, n) + 6 * qn * qn, n) - qn * qn * qn, n), n);
        if (u == 0) {
            //Простое число
            cout<<"time = "<<time(0) - tm<<endl;
            cout<<"m = "<<m1<<endl;
            cout<<"Prime n = "<<n<<endl<<endl;
            outfile<<N<<'# '<<m1<<'# '<<time(0) -
tm<<"#prime"<<'# '<<mpz_sizeinbase(n.get_mpz_t(),10)<<endl;
            N++;
        }
    }
} else{
    //Неопределенное
    cout<<"time = "<<time(0) - tm<<endl;
    cout<<"m = "<<m1<<endl;
    cout<<"Undefined n = "<<n<<endl<<endl;
    outfile<<N<<'# '<<m1<<'# '<<time(0) -
tm<<"#undefined"<<'# '<<mpz_sizeinbase(n.get_mpz_t(),10)<<endl;
    N++;
}
++m1;
ppowm *= 7;
}
outfile.clear();
outfile.close();
mfile.close();
mfile.open(argv[2],ios::out | ios::trunc);
mfile<<m1;
mfile.close();
return 0;
} catch(const char* s) {
    cout<<s<<endl;
} catch(mpz_class a) {
    cout<<a<<endl;
}
}

```