ДОДАТОК А

```
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "math.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;

void from_vector_memo1(vector<bool>& temp,TMemo *Memo1){
      AnsiString s="";
      for (long i=0;i<(temp.size());i++)
      {
      s=s+IntToStr(abs(temp[i]));
      };
      Memo1->Lines->Add(s);
};
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
      : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm1::N2Click(TObject *Sender)
{  Close();
}
//---------------------------------------------------------------------------
void TForm1::from_memo_to_vector(vector<long>* temp,TMemo *Memo1){
for (long i=0;i<Memo1->Lines->Count;i++){
temp->push_back(StrToInt(Memo1->Lines->Strings[i]));
};

};
//***************************************************************************
**********
void TForm1::makevectorkey(AnsiString p0, vector<long>& prime_mods,vector<bool>&  globalvector){
verylong firstmod=0;
verylong P0=0;
zsread(p0.c_str(),&P0);
zsqrt1(P0,&sqrtP0,&sqrtP0diference);
zsadd(sqrtP0,1,&sqrtP0);
verylong one=0;
verylong firststep=0;
zsetbit(&one,0);
long sqrtP0diference0=0;
long step0=0;
long prevsize=0;
long count_of_modulus=StrToInt(Edit2->Text);
vector<bool>stepvector;
zadd(sqrtP0,one,&firststep);
```

```
zsmul(firststep,2,&firststep);
zsub(firststep,one,&firststep);

for (long i=0;i<count_of_modulus;i++)
{
stepvector.resize(prime_mods[i]);
sqrtP0diference0=zsmod(sqrtP0diference,prime_mods[i]);
stepvector[0]=is_square_modulus(sqrtP0diference0,prime_mods[i]);
step0=zsmod(firststep,prime_mods[i]);
   for (long j=1;j<prime_mods[i];j++){
   sqrtP0diference0=(sqrtP0diference0+step0)%prime_mods[i];
     stepvector[j]=is_square_modulus(sqrtP0diference0,prime_mods[i]);
   step0=(step0+2)%prime_mods[i];
   };
   if (i==0){
   globalvector.resize(stepvector.size());
   for(int h=0;h<stepvector.size();h++)globalvector[h]=stepvector[h];

   }else
   {
   prevsize=globalvector.size();
      globalvector.resize(stepvector.size()*globalvector.size());

for(int h=0;h<globalvector.size();h++)
if (globalvector[h%prevsize]&&(!stepvector[h%stepvector.size()]))
{globalvector[h]=false;}else {globalvector[h]=globalvector[h%prevsize];};
   };

 };
  // from_vector_memo1(globalvector,Memo1);
//************************************
};
//---------------------------------------------------------------------------
void __fastcall TForm1::Button4Click(TObject *Sender)
{
firstmod=0;
from_memo_to_vector(&prime_mods,Memo2);
bool finded=false;

zsread(Edit1->Text.c_str(),&P0);
zsqrt1(P0,&sqrtP0,&sqrtP0diference);
zsadd(sqrtP0,1,&sqrtP0);
print(P0," P0 ");
print(sqrtP0," sqrtP0 ");
print(sqrtP0diference," sqrtP0diference ");
find_end_element(sqrtP0,one,&endstep);
find_end_element(sqrtP0,one,&firststep);
print(endstep," endstep ");
print(firststep," firststep ");
from_memo_to_vector(&prime_mods,Memo2);
long sqrtP0diference0=0;
long step0=0;
long prevsize=0;
long count_of_modulus=StrToInt(Edit2->Text);
```

```
vector<bool>stepvector;

AnsiString s=""
;
//************************************
for (long i=0;i<count_of_modulus;i++)
{
s="";
stepvector.resize(prime_mods[i]);
sqrtP0diference0=zsmod(sqrtP0diference,prime_mods[i]);
s=s+IntToStr(is_square_modulus(sqrtP0diference0,prime_mods[i]))+" ";
stepvector[0]=is_square_modulus(sqrtP0diference0,prime_mods[i]);
step0=zsmod(firststep,prime_mods[i]);
   for (long j=1;j<prime_mods[i];j++){
   sqrtP0diference0=(sqrtP0diference0+step0)%prime_mods[i];
   s=s+IntToStr(is_square_modulus(sqrtP0diference0,prime_mods[i]))+" ";
   stepvector[j]=is_square_modulus(sqrtP0diference0,prime_mods[i]);
   step0=(step0+2)%prime_mods[i];
   };
   if (i==0){
   globalvector.resize(stepvector.size());
   for(int h=0;h<stepvector.size();h++)globalvector[h]=stepvector[h];
    //  from_vector_memo1(globalvector,Memo1);
   }else
   {
   prevsize=globalvector.size();
      globalvector.resize(stepvector.size()*globalvector.size());
for(int h=0;h<globalvector.size();h++)
if (globalvector[h%prevsize]&&(!stepvector[h%stepvector.size()]))
{globalvector[h]=false;}else {globalvector[h]=globalvector[h%prevsize];};
   };
 };
from_vector_memo1(globalvector,Memo1);
long i=1;
verylong total=0;
verylong temp=0;
verylong dif=0;
verylong final=0;
bool limit=false;
zcopy(sqrtP0diference,&final);
while (!limit&&i<10000000){
      zadd(final,endstep,&final);
      zsadd(endstep,2,&endstep);
      if (globalvector[i%globalvector.size()]==1) {
           // Memo1->Lines->Add(IntToStr(globalvector[i%globalvector.size()]));
            zsqrt(final,&temp,&dif);
            if (ziszero(dif)==1){limit=true;zsadd(endstep,-2,&endstep);};
      };
      i++;
};
    Memo1->Lines->Add("Факторизовано ...");
    print(endstep," endstep ");
zsadd(endstep,1,&endstep);
zsdiv(endstep,2,&endstep);
```

```
    print(endstep," endstep ");
    verylong temp11=0;
    verylong temp12=0;

 zmul(endstep,endstep,&temp11);
 zsub(temp11,P0,&temp12);
 zsqrt(temp12,&temp,&dif);
 zsub(endstep,temp,&temp12);
   print(temp12," перший множник  ");
 zadd(endstep,temp,&temp12);
 print(temp12," другий множник  ");
//*************************************
}
void TForm1::from_vector_memo(vector<long>& temp,TMemo *Memo1){
      for (long i=0;i<(temp.size());i++)
      {
      long s=temp[i];
      Memo1->Lines->Add(IntToStr(s));
      };
};
void __fastcall TForm1::Button1Click(TObject *Sender)
{
from_memo_to_vector(&prime_mods,Memo2);

}
//---------------------------------------------------------------------------
bool TForm1::is_true_square(verylong part,verylong firststep,verylong end_step,verylong count){
verylong sum=0;
verylong result=0;
verylong diference=0;

zadd(end_step,firststep,&sum);
zmul(sum,count,&result);
zsdiv(result,2,&sum);
zadd(part,sum,&sum);
zsqrt(sum,&result,&diference);
if (ziszero(diference)==1){zfree(&result);zfree(&diference);zfree(&sum);return
true;}else{zfree(&result);zfree(&diference);zfree(&sum);return false;};
};
//---------------------------------------------------------------------------
bool TForm1::is_square_modulus(long part,long modul){
if (part!=0){
long m=1;
bool is=false;
for(long i=1;i<=modul/2;i++){
m=(i*i)%modul;
if (part==m){is=true;break;}
};
return is;   }else{return true;};
};
//---------------------------------------------------------------------------

void __fastcall TForm1::FormCreate(TObject *Sender)
{
```

```
        P0=0;
        sqrtP0=0;
        sqrtP0diference=0;
        firststep=0;
        endstep=0;
        count_elements=0;
        one=0;
        zsetbit(&one,0);}
//---------------------------------------------------------------------------
void TForm1::zsqrt1(verylong P0,verylong *sqrtP0,verylong *sqrtP0diference){
  verylong nextsq=0;
 zsqrt(P0,sqrtP0,sqrtP0diference);
 zadd(*sqrtP0,one,sqrtP0);
 zsq(*sqrtP0,&nextsq);
 zsub(nextsq,P0,sqrtP0diference);
 zsub(*sqrtP0,one,sqrtP0);
  };
//---------------------------------------------------------------------------
       void TForm1::print(verylong p,AnsiString s){
       char q[1024];
       zswrite(q,p);
       s=s+" ";
       s=s+q;
       Memo1->Lines->Add(s);       };
//---------------------------------------------------------------------------
void TForm1::find_end_element(verylong sqrtp0,verylong count,verylong *endelement){

zadd(sqrtp0,count,endelement);
zsmul(*endelement,2,endelement);
zsub(*endelement,one,endelement);};
void __fastcall TForm1::Button5Click(TObject *Sender)
{
from_memo_to_vector(&prime_mods,Memo2);
//globalvector=0;
makevectorkey(Edit1->Text,prime_mods,globalvector);
from_vector_memo1(globalvector,Memo1);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button6Click(TObject *Sender)
{
//TDateTime c=Time();
//Memo1->Lines->Add(Time());
firstmod=0;
bool finded=false;

zsread(Edit1->Text.c_str(),&P0);
zsqrt(P0,&sqrtP0,&sqrtP0diference);
zsadd(sqrtP0,1,&sqrtP0);
print(P0," P0 ");
AnsiString s=""
//1000

sqrtP0diference0=zsmod(sqrtP0diference,prime_mods[i]);
s=s+IntToStr(is_square_modulus(sqrtP0diference0,prime_mods[i]))+" ";
```

```
stepvector[0]=is_square_modulus(sqrtP0diference0,prime_mods[i]);
step0=zsmod(firststep,prime_mods[i]);
   for (long j=1;j<prime_mods[i];j++){
   sqrtP0diference0=(sqrtP0diference0+step0)%prime_mods[i];
   s=s+IntToStr(is_square_modulus(sqrtP0diference0,prime_mods[i]))+" ";
   stepvector[j]=is_square_modulus(sqrtP0diference0,prime_mods[i]);
   step0=(step0+2)%prime_mods[i];
   };
   if (i==0){
   globalvector.resize(stepvector.size());
   for(int h=0;h<stepvector.size();h++)globalvector[h]=stepvector[h];
    //  from_vector_memo1(globalvector,Memo1);
   }else
   {
   prevsize=globalvector.size();
      globalvector.resize(stepvector.size()*globalvector.size());
for(int h=0;h<globalvector.size();h++)
if (globalvector[h%prevsize]&&(!stepvector[h%stepvector.size()]))
{globalvector[h]=false;}else {globalvector[h]=globalvector[h%prevsize];};
   };
};
from_vector_memo1(globalvector,Memo1);
long i=1;
verylong total=0;
verylong temp=0;
verylong dif=0;
verylong final=0;
bool limit=false;
verylong temp=0;
verylong temp1=0;
verylong temp2=0;
bool limit=false;
long i=1;
while (!limit&&i<10000000){

      zadd(sqrtP0,one,&sqrtP0);
      zsq(sqrtP0,&temp);
      zsub(temp,P0,&temp);
            if (ziszero(temp2)==1){limit=true;print(temp,"  Квадрат з якого знаходиться корінь ");};

      i++;
};
    Memo1->Lines->Add("Факторизовано ...");

    verylong dif=0;
//*************************************
   verylong temp11=0;
    verylong temp12=0;

 zmul(endstep,endstep,&temp11);
 zsub(temp11,P0,&temp12);
 zsqrt(temp12,&temp,&dif);
 zsub(endstep,temp,&temp12);
  print(temp12," перший множник  ");
```

```
  zadd(endstep,temp,&temp12);
  print(temp12," другий множник  ");

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button7Click(TObject *Sender)
{
TDateTime c=Time();
 Memo1->Lines->Add(Time());
for (int i=0; i<StrToInt(Edit4->Text);i++)
Button4->Click();
        Memo1->Lines->Add(Time());
        Memo1->Lines->Add(FloatToStr(Time()-c));

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button8Click(TObject *Sender)
{
TDateTime c=Time();
 Memo1->Lines->Add(Time());
for (int i=0; i<StrToInt(Edit4->Text);i++)
Button6->Click();
        Memo1->Lines->Add(Time());
        Memo1->Lines->Add(FloatToStr(Time()-c));
}
//---------------------------------------------------------------------------
```