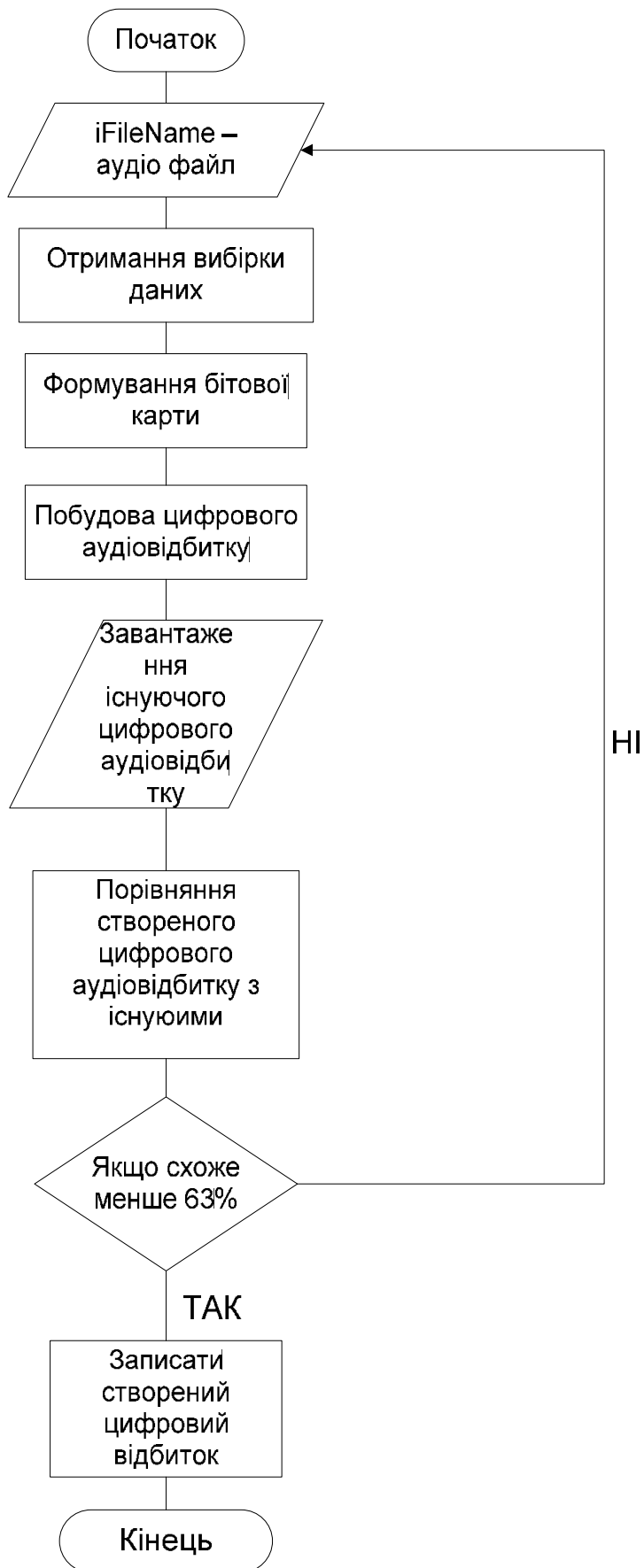


## ДОДАТОК А

### Алгоритм роботи програми



ДОДАТОК Б  
Лістинг програми

```
//-----  
  
#include <vcl.h>  
#include <string>  
#pragma hdrstop  
  
#include "Unit1.h"  
#include "bass.h"  
#include "math.h"  
#include <string>  
  
//-----  
  
#pragma package(smart_init)  
#pragma link "PERFGRAP"  
#pragma resource "*.dfm"  
TForm1 *Form1;  
  
const void *path;  
HSTREAM stream;  
float data[128];  
float buffer[128][128];  
float band[128][32];  
int BitMap[128][32];  
int BitMap2[128][32];  
int seconds=0;  
int chas=0;  
int length_seconds;  
int h=0,s=0;
```

```

//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    DrawGrid1->GridLineWidth=0;
    DrawGrid2->GridLineWidth=0;
}
//-----

void __fastcall TForm1::PlayClick(TObject *Sender)
{
    BASS_ChannelStop(stream);
    PerformanceGraph1->Repaint();
    BASS_StreamFree(stream); //ЧИСТИМО канал
    CreatReference->Enabled=false;
    N10->Enabled=false;
    // DrawGrid1->Canvas->
    Timer2->Enabled=False;
    chas=0;
    seconds=0;

    //Запускаємо файл
    BASS_Init(-1, 44100, 0, 0, NULL);
    stream=BASS_StreamCreateFile(false,path,0,0, BASS_SAMPLE_FLOAT);
    int length_byte=BASS_ChannelGetLength(stream, BASS_POS_BYTE);
    length_seconds=BASS_ChannelBytes2Seconds(stream,length_byte);
    BASS_ChannelPlay(stream,false);
    //Отримуємо дані за допомогою зсуву

```

```

int set=0;
for (int str=0; str<128; str++)
{
set=str*10001;
BASS_ChannelSetPosition(stream, set, BASS_POS_BYTE);
BASS_ChannelGetData(stream, data, BASS_DATA_FFT256);
for (int pos=0; pos<128; pos++)
{
buffer[str][pos]=data[pos];
}
}
CreatReference->Enabled=true;
//Запуск таймера
Timer2->Enabled=true;
Timer2->Interval=1000;
PerformanceGraph1->StepSize=1;
}
//-----
void __fastcall TForm1::PauseClick(TObject *Sender)
{
//Пауза в програванні файлу
BASS_ChannelPause(stream);
}
//-----
void __fastcall TForm1::StopClick(TObject *Sender)
{
//Зупинка в програванні файлу
PerformanceGraph1->Repaint();
BASS_ChannelStop(stream);
Timer2->Enabled=False;
}

```

```

//-----
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    DWORD level=BASS_ChannelGetLevel(stream);
    PerformanceGraph1->DataPoint(clBlue,level);
    PerformanceGraph1->Update();
    chas++;
    s++;
    if (s==60)
    {s=0;
    h++;}
    Label1->Caption=h;
    Label2->Caption=s;
    if (chas==length_seconds)
    {
        Timer2->Enabled=False;
    }
}
//-----

void __fastcall TForm1::CreatReferenceClick(TObject *Sender)
{
    //Сумуємо рядки
    float sum[128];
    memset(sum, 0, sizeof(sum));
    for (int a=0; a<128; a++)
    {
        for (int b=0; b<128; b++)
        {
            sum[a]=sum[a]+pow(buffer[a][b],2);
        }
    }
}

```

```

float step;
//Ділимо на смуги без ділення
for (int i=0; i<128; i++)
{
for (int j=0; j<128; j=j+4)
{
step=pow(buffer[i][j],2)+pow(buffer[i][j+1],2)+pow(buffer[i][j+2],2)+
pow(buffer[i][j+3],2);
buffer[i][j]=step;
}
}

```

```

//Ділимо на смуги з діленням
for (int i=0; i<128; i++)
{
for (int j=0; j<32; j++)
{
band[i][j]=buffer[i][j*4]/sum[i];
}
}

```

```

//Визначаємо 1,0
float bit;
for (int m=0; m<128; m++)
{
for(int n=0; n<32; n++)
{
bit=band[m][n]-band[m+1][n]-(band[m][n-1]-band[m+1][n-1]);

if (bit>0)
{BitMap[m][n]=1;

```

```

DrawGrid1->Canvas->Brush->Color=c1Black;
DrawGrid1->Canvas->FillRect(DrawGrid1->CellRect(n,m));
}
else
{BitMap[m][n]=0;
DrawGrid1->Canvas->Brush->Color=c1White;
DrawGrid1->Canvas->FillRect(DrawGrid1->CellRect(n,m));}
N10->Enabled=true;
}
}
}
//-----

```

```

void __fastcall TForm1::Button1Click(TObject *Sender)
{
int iFileHandle;
if (!OpenDialog2->Execute()) return;
try {
iFileHandle = FileOpen(OpenDialog2->FileName, fmOpenRead);
FileSeek(iFileHandle,0,0);
FileRead(iFileHandle, BitMap2, sizeof(BitMap2));
}
catch(...) {
Application->MessageBox("Під час відкриття сталася помилка",
"Помилка відкриття",
MB_ICONSTOP+MB_OK);
}
FileClose(iFileHandle);

for (int m=0; m<128; m++)

```

```

{
for(int n=0; n<32; n++)
{
if (BitMap2[m][n]==1)
{
DrawGrid2->Canvas->Brush->Color=c1Black;
DrawGrid2->Canvas->FillRect(DrawGrid1->CellRect(n,m));
}
if (BitMap2[m][n]==0)
{
DrawGrid2->Canvas->Brush->Color=c1White;
DrawGrid2->Canvas->FillRect(DrawGrid1->CellRect(n,m));
}
}
}
}
//-----

void __fastcall TForm1::N4Click(TObject *Sender)
{
OpenDialog1->Execute(); //виклик вікна відкриття
Label3->Caption=ExtractFileName(OpenDialog1->FileName);
path=OpenDialog1->FileName.c_str();//перетворюює стрічку в const void
Label3->Width=185;
}
//-----

void __fastcall TForm1::N5Click(TObject *Sender)
{
Form1->Close();
}

```



```

//-----

void __fastcall TForm1::N6Click(TObject *Sender)
{
Play->Click();
}

//-----

void __fastcall TForm1::N7Click(TObject *Sender)
{
Pause->Click();
}

//-----

void __fastcall TForm1::N8Click(TObject *Sender)
{
Stop->Click();
}

//-----

void __fastcall TForm1::N10Click(TObject *Sender)
{
int iFileHandle;
if (!SaveDialog1->Execute()) return;
try {
iFileHandle = FileCreate(SaveDialog1->FileName);
FileWrite(iFileHandle, BitMap, sizeof(BitMap));
}
catch(...) {
Application->MessageBox("Під час запису сталася помилка",
"Помилка запису",
MB_ICONSTOP+MB_OK);
}
}

```

```

}
FileClose(iFileHandle);
}
//-----

void __fastcall TForm1::N11Click(TObject *Sender)
{
int iFileHandle;
if (!OpenDialog2->Execute()) return;
try {
    iFileHandle = FileOpen(OpenDialog2->FileName, fmOpenRead);
    FileSeek(iFileHandle,0,0);
    FileRead(iFileHandle, BitMap2, sizeof(BitMap2));
}
catch(...) {
    Application->MessageBox("Під час відкриття сталася помилка",
        "Помилка відкриття",
        MB_ICONSTOP+MB_OK);
}
FileClose(iFileHandle);

for (int m=0; m<128; m++)
{
for(int n=0; n<32; n++)
{
if (BitMap2[m][n]==1)
{
DrawGrid2->Canvas->Brush->Color=clBlack;
DrawGrid2->Canvas->FillRect(DrawGrid1->CellRect(n,m));
}
if (BitMap2[m][n]==0)

```

```

{
DrawGrid2->Canvas->Brush->Color=c1White;
DrawGrid2->Canvas->FillRect(DrawGrid1->CellRect(n,m));
}
}
}
}
//-----

```

```

void __fastcall TForm1::N12Click(TObject *Sender)

```

```

{
int a=0;
for (int m=0; m<128; m++)
{
for(int n=0; n<32; n++)
{

if (n>0 && n<31)
{
if (BitMap[m][n]==BitMap2[m][n] && BitMap[m][n-1]==BitMap2[m][n-1]
&& BitMap[m][n+1]==BitMap2[m][n+1])
{a++;}
}

if (n==0)
{
if (BitMap[m][n]==BitMap2[m][n] && BitMap[m][n+1]==BitMap2[m][n+1])
{a++;}
}

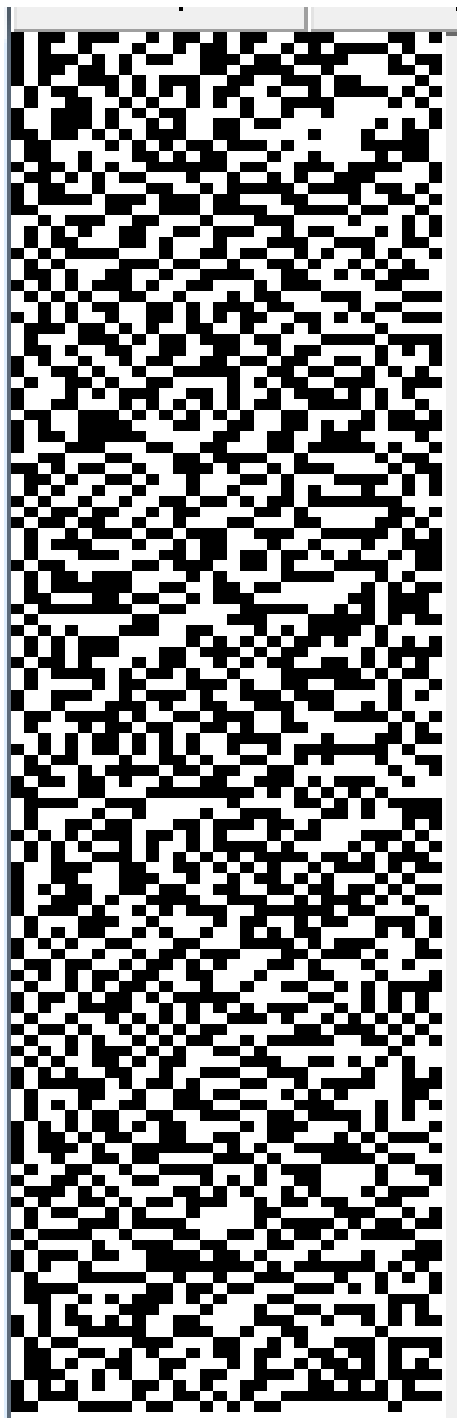
if (n==31)

```

```
{
if (BitMap[m][n]==BitMap2[m][n] && BitMap[m][n-1]==BitMap2[m][n-1])
{a++;}
}
}}
float b=a/4096*100;
AnsiString text="Композиції схожі на: "+FloatToStr(b)+"%.";
MessageBox(NULL,text.c_str(),"Результат",MB_OK);}
//-----
```

## ДОДАТОК В

Приклад цифрового відбитку



## ДОДАТОК Г

Світлокопії виданих публікацій

ДОДАТОК Д  
Довідка про впровадження