

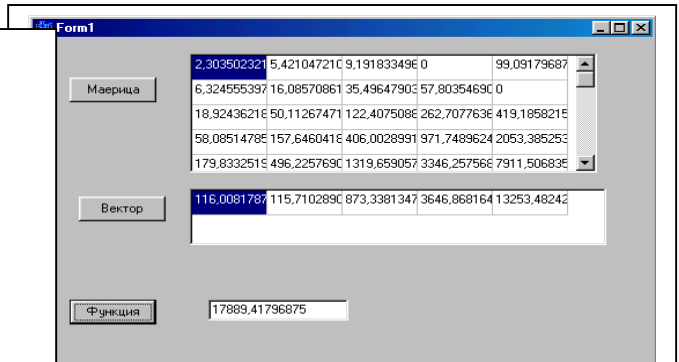
# Matrix and vector calculations using DLL

```
#include "UMyDLL.h"
float **a;
float* x;
int n=5;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
    a=new float*[n];
    for (int i=0;i<n;i++)
    a[i]=new float[n];
    x=new float[n];
}

Button1Click
{
    matr(a,n);
    for (int i=0;i<n;i++)
    for (int j=0;j<n;j++)
    SG1->Cells[j][i]=FloatToStr(a[i][j]);
}

Button3Click
{
    float g=G(x,n);
    Edit1->Text=g;
}

```



```
Button1Click
{
    matr(a,n);
    for (int i=0;i<n;i++)
    for (int j=0;j<n;j++)
    SG1->Cells[j][i]=FloatToStr(a[i][j]);
}

```

```
Button2Click
{
    vector(a,x,n);
    for (int i=0;i<n;i++)
    SG2->Cells[i][0]=x[i];
}

```

UMyDLL.cpp

UMyDLL.h

```
#include <vcl.h>
#include <windows.h>
#pragma hdrstop
#include <Math.h>
#pragma argsused
#include "UMyDLL.h"

void matr(float** a, int n)
{
    for (int i=0;i<n;i++)
    for (int j=0;j<n;j++)
    a[i][j]=pow(3,i+j)*sqrt(i+2.6+j)*
    (2*fabs(i-j+3))/(3*(i+1.4));
}

void vector(float** a, float b[], int n)
{
    float s;
    for (int i=0;i<n;i++)
    {
        s=0;
        for (int j=0;j<n;j++)
        s+=a[i][j];
        b[i]=s;
    }
}

float G(float b[], int n)
{
    float s=0;
    for (int i=0;i<n-1;i++)
    s+=1/(1+b[i])+b[i+1];
    return s;
}

```

```
#ifndef _UMYDLL_H
#define _UMYDLL_H
#ifdef __DLL__
#define DLL_EI __declspec(dllexport)
#else
#define DLL_EI __declspec(dllimport)
#endif

extern "C" void DLL_EI matr(float** a, int n);

extern "C" void DLL_EI vector(float** a, float b[],
                             int n);

extern "C" float DLL_EI G(float b[], int n);

#endif

```