OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN

SEISMIC EVENT DETECTION

DT008/3 Software Design 2

Assignment 1 2019/2020

21st November 2019

Talha Tallat (d18124645)
D18124645@mytudublin.ie

Table of Contents

Table of figures

1.0 Introduction

1.1 Background

A very special sensor is called seismometers are used to collect earth motion information including earthquakes and tidal motion. By analysing ground motion from an earthquake using data from a several seismometers, it is possible to determine the epicentre of an earthquake and the intensity of the earthquake (normally measured using the **Richter scale**.

1.2 Description

Writing a C++ Object Oriented program that reads seismometer data (text) file named seismic.dat, determines whether seismic events have occurred, reports the data and findings to the screen in text and graphics format and stores the findings to a data file.

The first line of the seismic data file contains two values: the number of data elements or readings and the time interval in seconds that occurred between consecutive measurements. The time interval is a floating-point value and it may be assumed that all the measurements were taken with the same time interval between them. After reading the data measurements the program should identify possible earthquakes or seismic events using a power ratio.

At a specific point in time this ratio is the quotient of a short-time power measurement divided by a long -time power measurement. If the ratio is greater than a given threshold then an event may have occurred at that point in time.

Given a specific point in the data measurements the **short-time power** is the average power, or average squared value, of the measurements using the specified point plus a small number of points that occurred just before the specified point.

The **long-time power** is the average power of the measurements using the specified point plus a larger number of points that occurred just before the specified point. The set of points used in a calculation is referred to as a data window.

The **threshold** is generally greater than 1 to avoid detecting events in constant data because the short-time power is equal to the long-time power if the data values are all the same value.

The data window length (number of measurements) for short-time power and long-time power are to be read from the keyboard as is the threshold value.

2.0 Procedure

2.1 UML class Diagram

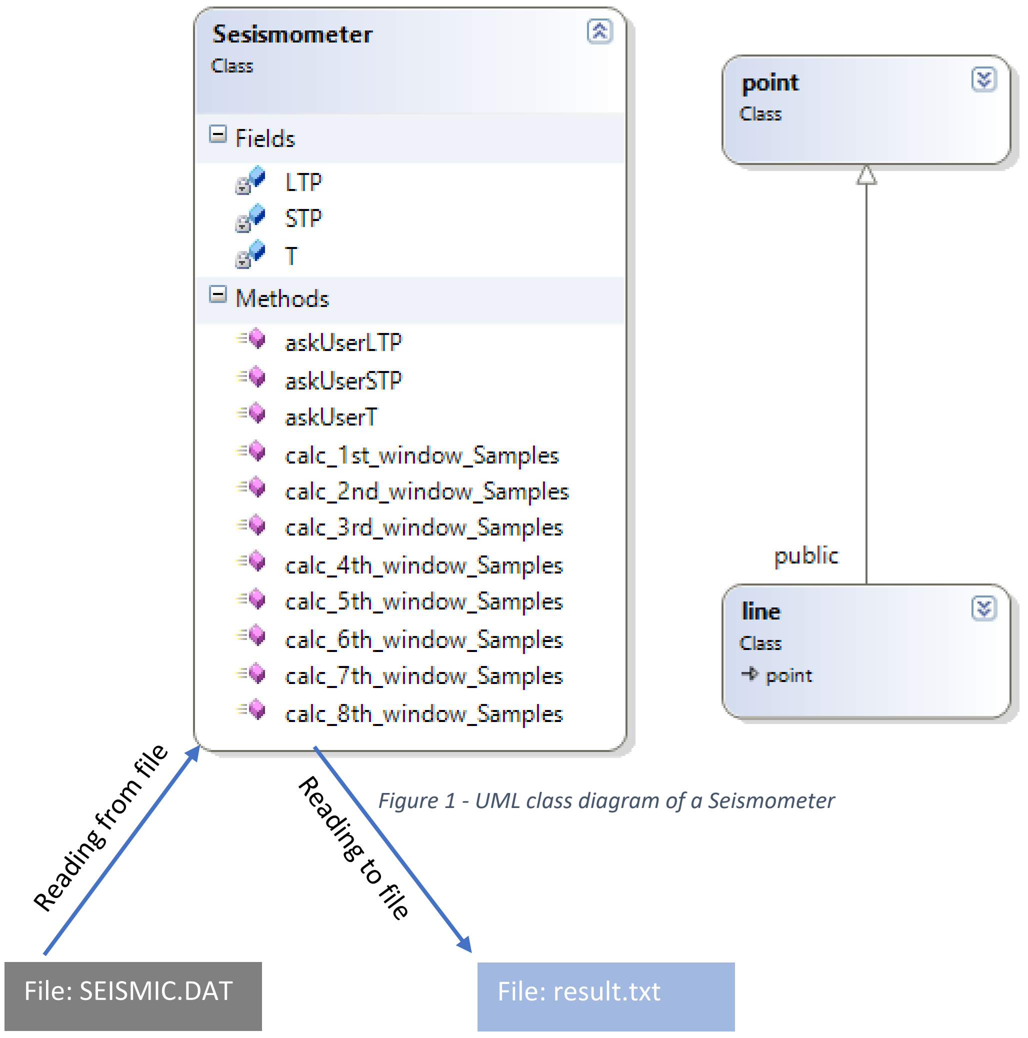


Figure 1 - UML class diagram of a Seismometer

2.2 Code:

2.2.1 Seismometer.h

```
#ifndef SESISMOMETER
#define SESISMOMETER

#include <iostream>
#include <fstream>
#include <string>

using namespace std;


fstream infile;
       ofstream  outfile("result.txt",ios::out);

class Sesismometer // class seismic
{
```

```
private:
	double STP; // short time power
	double LTP; //long time power
	double T; //  threshold

public:
	void calc_1st_window_Samples(float stp_1 , float stp_2 , float ltp_1 , float 
ltp_2 , float ltp_3, float ltp_4, float ltp_5);
	void calc_2nd_window_Samples (float stp_1 , float stp_2 , float ltp_1 , float 
ltp_2 , float ltp_3, float ltp_4, float ltp_5 , float ltp_6);

	void askUserSTP(); // ask user the short time power value
	void askUserLTP(); // ask user the long time power value
	void askUserT(); // ask User the threshold value
};

//ofstream  outfile("result.txt",ios::out);


//------------------------Class Methods------------------------

//ask functions
void Sesismometer:: askUserT() {

cout <<"\n\nEnter sample for short time window !!NOTE: short time can only have 2 
sample!! => " ;
	cin >> STP;
	cout <<"Enter sample for long time window !!NOTE: long time can have 5 or 6 
sample!! => " ;
	cin >> LTP;
	cout <<"Enter value for threshold => " ;
	cin >> T;


}

// calcultaing Short time window of 2 samples and Long time window of 5 samples and 
threshold 1.5
void Sesismometer:: calc_1st_window_Samples (float stp_1 , float stp_2 , float ltp_1 , 
float ltp_2 , float ltp_3, float ltp_4, float ltp_5)
{
	ofstream  outfile("result.txt",ios::out);

	if (STP == 2 && LTP == 5 &&  T)    {
		double tempSTP,tempLTP,tempRatio;
		tempSTP = (((stp_1 * stp_1) + (stp_2 * stp_2) ) /2);
		cout<<"The Short-time power is => " <<tempSTP <<endl;

		tempLTP = (((ltp_1*ltp_1) + (ltp_2*ltp_2) + (ltp_3*ltp_3) + 
(ltp_4*ltp_4) + (ltp_5*ltp_5)) /5);
		cout<<"The long-time power is => " <<tempLTP <<endl;

		tempRatio = (tempSTP / tempLTP);
	    cout <<"The ratio is => " <<tempRatio <<endl;

		cout << "The Threshold is => " << T <<endl;

		if (tempRatio >= T)  {
			cout<<"\n!!Event at this time!!" << endl;
		}

		else {
```

```
                    cout <<"\nNo event at this time" <<endl;
                }

                outfile <<"\nThe Short-time power is => "<<tempSTP
           <<"\nThe long-time power is => "<<tempLTP
           <<"\nThe ratio is => " <<tempRatio
           << "\nThe Threshold is => " << T;
        }

}



// calcultaing Short time window of 2 samples and Long time window of 5 samples and
threshold 1.5
void Sesismometer :: calc_2nd_window_Samples (float stp_1 , float stp_2 , float ltp_1
, float ltp_2 , float ltp_3, float ltp_4, float ltp_5 , float ltp_6) {
       ofstream  outfile ("result.txt",ios::out);
              double stp,ltp,ratio;
       if (STP == 2 && LTP == 6 && T  )  {
              double i=6;
              double time = i-2;
              time = time * 0.01;


              stp = (((stp_1 * stp_1) + (stp_2 * stp_2) ) /2);
              cout<<"The Short-time power is => " <<stp <<endl;

              ltp = (((ltp_1*ltp_1) + (ltp_2*ltp_2) + (ltp_3*ltp_3) + (ltp_4*ltp_4) +
(ltp_5*ltp_5) + (ltp_6*ltp_6)) / 6);
              cout<<"The long-time power is => " <<ltp <<endl;

              ratio = (stp / ltp);
           cout <<"The ratio is => " <<ratio <<endl;

              cout << "The Threshold is => " << T <<endl;

              if (ratio >= T)      {
              cout<<"\n!!Event at this time!!"  <<endl;
              }

              else {
              cout <<"\nNo event at this time"  <<endl;
              }


       }
       // save calculate values into the text file
       outfile <<"\n\t\tAt time => "
              <<"\nThe Short-time power is => " << stp
         <<"\nThe long-time power is => " << ltp
          <<"\nThe ratio is => " << ratio
          << "\nThe Threshold is => " << T;


}

#endif
```

2.2.2 Seismometer.cpp

```
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include "Graphics.h"
#include "Seismometer.h"

#define SIZEOFTHENAME 25


using namespace std;


//----------------------Main---------------------

int main () {

       Sesismometer Values;

       fstream infile;
       infile.open("SEISMIC.DAT");// reads from file


       //check for errors
       if (infile.fail()) {
              cout << "Error Openning File" << endl;
              exit(1);
       }

       // values stored in the file
       float a,b,c,d,e,f,g,h,i,j,k,l,m;
       // reading values individually
       infile >> a >> b >> c >> d >> e >> f >> g >> h >> i >> j >> k >> l >> m;

       //---------------------0,1,2,3,4,5,6,7,8,9,10,11,12
       float testScores [] = {a,b,c,d,e,f,g,h,i,j,k,l,m}; // storing in string array
       int numberOfElements = sizeof (testScores) / sizeof(testScores[0]);

       cout << "Data from the file contains the following data: " <<endl;
       for ( int i = 0 ; i <  numberOfElements; i++){
              cout  <<testScores[i] <<" ";
       }

       //ask user the short time power, long time power and threshold
       Values.askUserT();

       double r=6;
       double timet = r-2;
       timet  = timet  * 0.01;

       cout <<"\n\n\t\t----- First valid point is at time: "<< timet <<"s -----"
<<endl << endl;
       Values.calc_1st_window_Samples (g,f,g,f,e,d,c);
       Values.calc_2nd_window_Samples (h,g,h,g,f,e,d,c);

       cout <<"\n\n\t\t----- Second valid point is at time: "<< timet+0.01 <<"s -----"
<<endl << endl;
       Values.calc_1st_window_Samples(g,h,h,g,f,e,d);
       Values.calc_2nd_window_Samples (i,h,i,h,g,f,e,d);

       cout <<"\n\n\t\t----- Third valid point is at time: "<< timet+0.02 <<"s -----"
<<endl << endl;
```

```
       Values.calc_1st_window_Samples(i,h,i,h,g,f,e);
       Values.calc_2nd_window_Samples (j,i,j,i,h,g,f,e);

       cout <<"\n\n\t\t----- Fourth valid point is at time: "<< timet+0.03 <<"s -----"
<<endl << endl;
       Values.calc_1st_window_Samples(j,i,j,i,h,g,f);
       Values.calc_2nd_window_Samples (k,j,k,j,i,h,g,f);

       cout <<"\n\n\t\t----- Fifth valid point is at time: " << timet+0.04 <<"s -----"
<<endl << endl;
       Values.calc_1st_window_Samples(k,j,k,j,i,h,g);
       Values.calc_2nd_window_Samples (l,k,l,k,j,i,h,g);

       cout <<"\n\n\t\t----- Sixth valid point is at time: " << timet+0.05 <<"s -----"
<<endl << endl;
       Values.calc_1st_window_Samples(l,k,l,k,j,i,h);
       Values.calc_2nd_window_Samples (m,l,m,l,k,j,i,h);

       cout <<"\n\n\t\t----- Seventh valid point is at time: "<< timet+0.06 <<"s -----
" <<endl << endl;
       Values.calc_1st_window_Samples(m,l,m,l,k,j,i);


       //graphics printing out the line
       line  line1( 100, 200,0,200,200, 100); // x,y,w,r,g,b
    HWND hWnd=GetForegroundWindow();
    HDC hDC=GetDC(hWnd);
    line1.draw(hDC);

       return 0;

}
```

2.3 Test results together

The program asks user to input short-time window sample, long-time window sample and threshold. Then the program read files from “SEISMIC.DAT” and perform calculations accordingly for the short time power, long time power and ratio of both power and if the ratio is more than the chosen threshold entered by the user then the program alerts the user, that whether there is an event occurs or not.

2.3.1 Read from file

The program read values from the file.

The file “SEISMIC.DAT” contains these values:

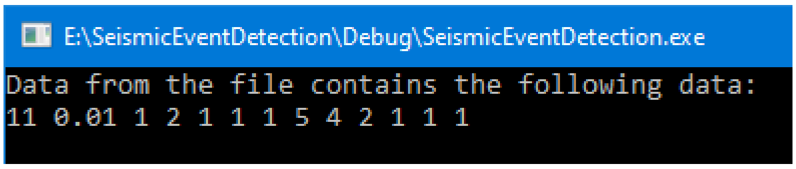
11 0.01 1 2 1 1 1 5 4 2 1 1 1

First value shows the total number of data and second value shows the time interval.

The **code** the was used to read the files:

```
fstream infile;
infile.open("SEISMIC.DAT");// reads from file


       //check for errors
       if (infile.fail()) {
              cout << "Error Openning File" << endl;
              exit(1);
```

```
        }

        // values stored in the file
        float a,b,c,d,e,f,g,h,i,j,k,l,m;
        // reading values individually
        infile >> a >> b >> c >> d >> e >> f >> g >> h >> i >> j >> k >> l >> m;

        //---------------------0,1,2,3,4,5,6,7,8,9,10,11,12
        float testScores [] = {a,b,c,d,e,f,g,h,i,j,k,l,m}; // storing in string array
        int numberOfElements = sizeof (testScores) / sizeof(testScores[0]);

        cout << "Data from the file contains the following data: " <<endl;
        for ( int i = 0 ; i <  numberOfElements; i++){
               cout  <<testScores[i] <<" ";
        }
```



```
E:\SeismicEventDetection\Debug\SeismicEventDetection.exe
Data from the file contains the following data:
11 0.01 1 2 1 1 1 5 4 2 1 1 1
```

Figure 2 - Console application shows Data in the SEISMIC.DAT file contains

SHORT TIME POWER $= \frac{(1st\ data)^2+(2nd\ data)^2}{Total\ number\ of\ data}$

LONG TIME POWER $= \frac{(1st\ data)^2+(2nd\ data)^2+\ (3rd\ data)^2+(4th\ data)^2+(5th\ data)^2}{Total\ number\ of\ data}$

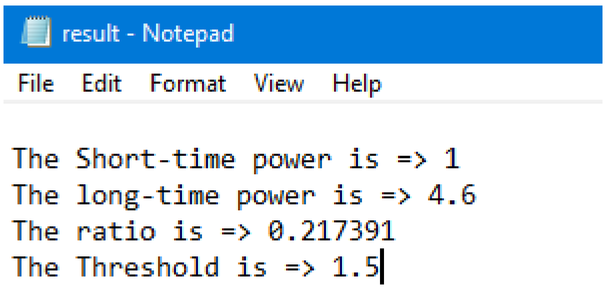
Ratio $= \frac{\text{SHORT TIME POWER}}{\text{LONG TIME POWER}}$

2.3.2 Read to file

The taken data from SEISMIC file was calculated and stored in the "result.text" file.

The **code** for read to file is:

```
outfile <<"\nThe Short-time power is => "<<tempSTP
         <<"\nThe long-time power is => "<<tempLTP
         <<"\nThe ratio is => " <<tempRatio
         << "\nThe Threshold is => " << T;
```



```
The Short-time power is => 1
The long-time power is => 4.6
The ratio is => 0.217391
The Threshold is => 1.5
```

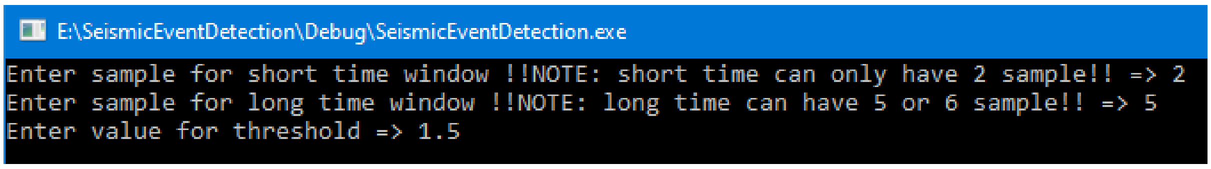
Figure 3 - result.text file contains calculated values

2.3.3 ask User the Values

The short time power, long time power and threshold values are asked which user must enter.

The **code** for askUser is:

```
//ask functions

void Sesismometer:: askUserT() {

cout <<"\n\nEnter sample for short time window !!NOTE: short time can only have 2
sample!! => " ;
       cin >> STP;
       cout <<"Enter sample for long time window !!NOTE: long time can have 5 or 6
sample!! => " ;
       cin >> LTP;
       cout <<"Enter value for threshold => ";
       cin >> T;
}
```



```
E:\SeismicEventDetection\Debug\SeismicEventDetection.exe
Enter sample for short time window !!NOTE: short time can only have 2 sample!! => 2
Enter sample for long time window !!NOTE: long time can have 5 or 6 sample!! => 5
Enter value for threshold => 1.5
```

Figure 4 - Ask user the values

2.3.5 Graphics

The program prints out line, I had difficulties doing the graphics parts how ever was able to get the line.

The **code** for graphics:

Header file for Graphics:

```
//==========================================//
//  OOP_DOS6.CPP  Vr. 2 2012      PK                    //
//  Demonstrates static and dynamic objects  //
//  Updated for GDI 08/10/2012               //
//==========================================//
#include <windows.h>
#include <iostream>   // for cerr
#include <stdio.h>
#include <conio.h>
#include "resource.h"

//=== class declarations ====================================

class   point
        {
   protected:  int   x;
               int   y;
               COLORREF RGB_color;
   public:     point()  {  x = 0;   y = 0; RGB_color=RGB(0,0,0); }
               point( int xval, int yval, int r,int g,int b );
               void  draw(HDC hdc);
               void  set_x( int xval );
               void  set_y( int yval );
               int   get_x();
               int   get_y();
        };

class   line:   public point
        {
   protected:  int   length;
   public:     line()  {  x = 0;   y = 0; RGB_color=RGB(0,0,0);  length = 0;  }
```

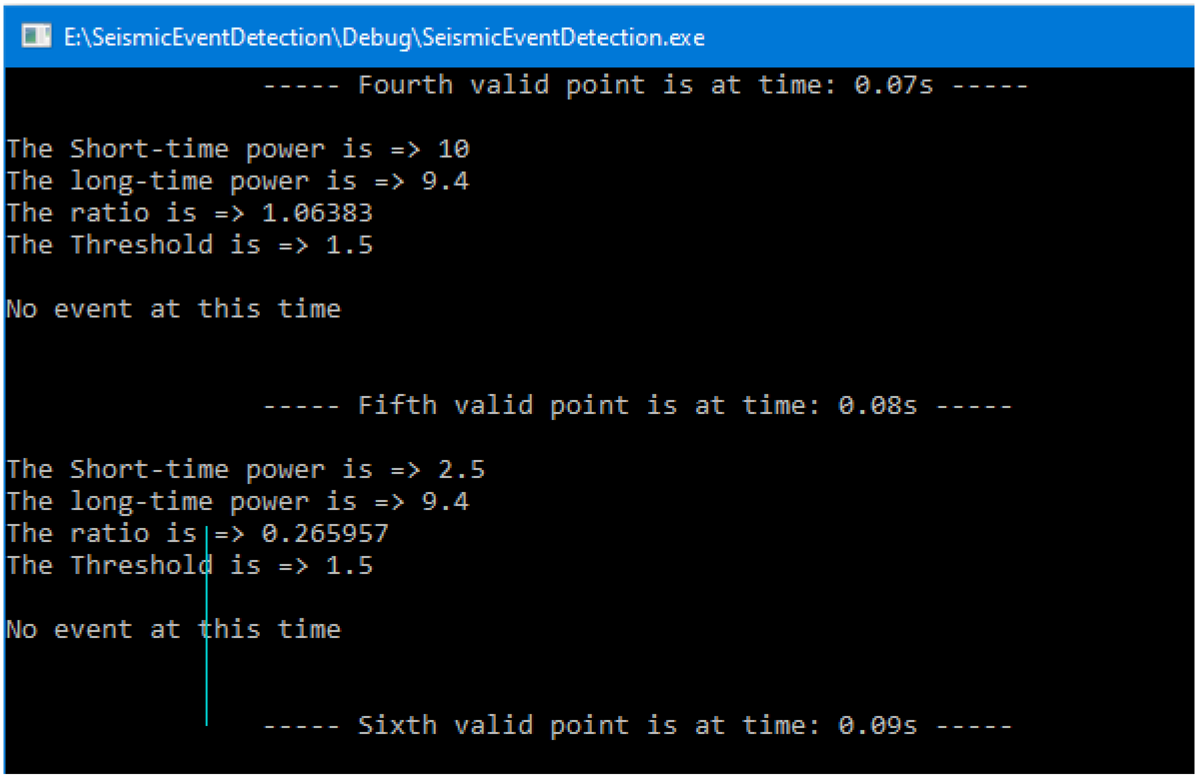
```
              line( int xval, int yval, int r,int g,int b , int ilength);
              void  draw(HDC hDC);
              void  set_len( int ilength );
              int   get_len();
       };




//=== methods for point class ================================
      point::point( int xval, int yval, int r, int g, int b )
                       {      x = xval;
                              y = yval;
                              RGB_color =RGB(r,g,b);  }

void  point::set_x( int xval )       {  x = xval;      }

void  point::set_y( int yval )       {  y = yval;      }

int   point::get_x()                 {  return( x );   }

int   point::get_y()                 {  return( y );   }

void  point::draw(HDC hDC)
      {
         int i,j;
         for( i= -2; i<3; i++ )
            for( j = -2; j<3; j++ )
               SetPixel(hDC, x+i, y+j,RGB_color);
      }

//=== methods for line class ================================
      line::line( int xval, int yval, int r, int g, int b, int ilength): point( xval,
yval, r,g,b)
                                     {  set_len(ilength);  }

void  line::set_len( int ilength )   {  length = ilength;  }

int   line::get_len()                {  return( length );  }

void  line::draw(HDC hDC)
      {

              HPEN pen=NULL;
              if (pen)DeleteObject(pen); //delete previous created pen

              pen=CreatePen(PS_SOLID,1,RGB_color);  // Create a new pen
         SelectObject(hDC,pen); // select pen into context
         MoveToEx (hDC,x,y,NULL);
              LineTo(hDC,x,y-length);
       }



//=== end of methods ========================================

//===============================//
//   global and static objects   //
//===============================//
```

```
point    point1( 10, 20, 255,0,0 );
point    point2( 20, 30, 0,255,0 );
point    point3( 30, 40, 0,0,255 );
line     line1( 100, 200,0,250,0, 200);
line    line2(150,150,0,0,250,400);
```

Graphic main:

```
line  line1( 100, 200,0,200,200, 100); // x,y,w,r,g,b
   HWND hWnd=GetForegroundWindow();
   HDC hDC=GetDC(hWnd);
   line1.draw(hDC);
```

Result:

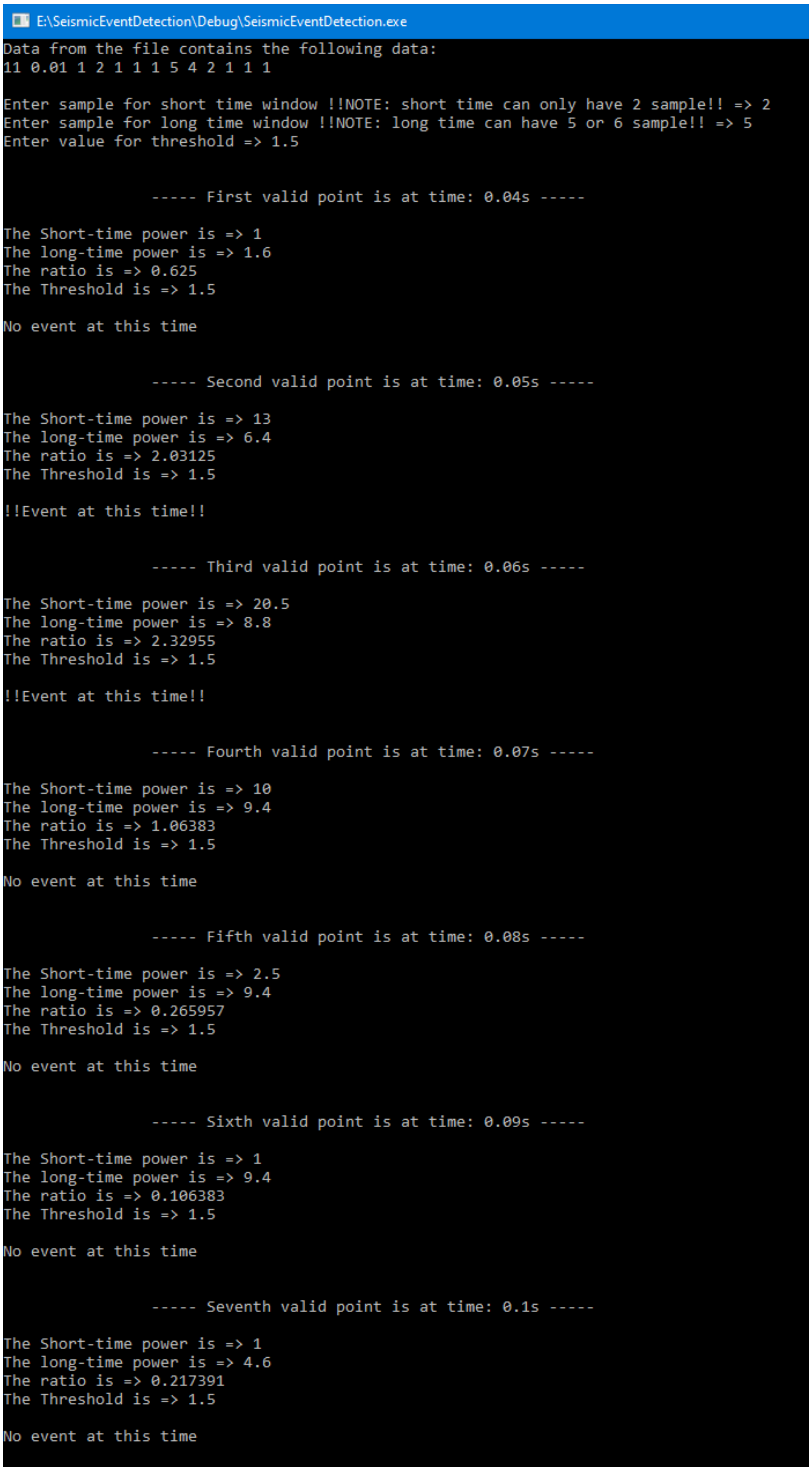


```
E:\SeismicEventDetection\Debug\SeismicEventDetection.exe
                  ----- Fourth valid point is at time: 0.07s -----

The Short-time power is => 10
The long-time power is => 9.4
The ratio is => 1.06383
The Threshold is => 1.5

No event at this time


                  ----- Fifth valid point is at time: 0.08s -----

The Short-time power is => 2.5
The long-time power is => 9.4
The ratio is => 0.265957
The Threshold is => 1.5

No event at this time


                  ----- Sixth valid point is at time: 0.09s -----
```

Figure 5 - Printed out line for graphics

2.3.5 calculations

The program performs calculations when user input the short-time, long-time window samples and threshold as shown down below .


E:\SeismicEventDetection\Debug\SeismicEventDetection.exe

```
Data from the file contains the following data:
11 0.01 1 2 1 1 1 5 4 2 1 1 1

Enter sample for short time window !!NOTE: short time can only have 2 sample!! => 2
Enter sample for long time window !!NOTE: long time can have 5 or 6 sample!! => 5
Enter value for threshold => 1.5


                ----- First valid point is at time: 0.04s -----

The Short-time power is => 1
The long-time power is => 1.6
The ratio is => 0.625
The Threshold is => 1.5

No event at this time


                ----- Second valid point is at time: 0.05s -----

The Short-time power is => 13
The long-time power is => 6.4
The ratio is => 2.03125
The Threshold is => 1.5

!!Event at this time!!


                ----- Third valid point is at time: 0.06s -----

The Short-time power is => 20.5
The long-time power is => 8.8
The ratio is => 2.32955
The Threshold is => 1.5

!!Event at this time!!


                ----- Fourth valid point is at time: 0.07s -----

The Short-time power is => 10
The long-time power is => 9.4
The ratio is => 1.06383
The Threshold is => 1.5

No event at this time


                ----- Fifth valid point is at time: 0.08s -----

The Short-time power is => 2.5
The long-time power is => 9.4
The ratio is => 0.265957
The Threshold is => 1.5

No event at this time


                ----- Sixth valid point is at time: 0.09s -----

The Short-time power is => 1
The long-time power is => 9.4
The ratio is => 0.106383
The Threshold is => 1.5

No event at this time


                ----- Seventh valid point is at time: 0.1s -----

The Short-time power is => 1
The long-time power is => 4.6
The ratio is => 0.217391
The Threshold is => 1.5

No event at this time
```

Figure 6 - Calculations