Приложение имитирующее работу КЭШа с прямым отображением

https://github.com/MAXXXYMIRON/ECM/tree/master/%D0%9A%D0%AD%D0%A8%20%20%D0%A0%D0%B0%D0%B1%D0%BE%D1%82%D0%B0/%D0%9A%D0%AD%D0%A8_%D0%A0%D0%B0%D0%B1%D0%BE%D1%82%D0%B0_%D0%9F%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5

Код реализации

https://github.com/MAXXXYMIRON/ECM/tree/master/%D0%9A%D0%AD%D0%A8%20%20%D0%A0%D0%B0%D0%B1%D0%BE%D1%82%D0%B0/CashWorking

Листинг

shWorking | CashWorking.MainMemory

```
using System;
using System.IO;

namespace CashWorking
{
    class MainMemory
    {
        BinaryWriter Write;
        BinaryReader Read;

        int CountSegments,
            CountLines,
            CountElements;

        public MainMemory(int countSegments, int countLines, int countElements)
        {
            Random values = new Random();
            CountSegments = countSegments;
            CountLines = countLines;
            CountElements = countElements;

            using (Write = new BinaryWriter(new FileStream("Memory.ini", FileMode.Create)))
            {
                //CountSegments - блоков
                for (int i = 0; i < CountSegments; i++)
                {
                    Write.Write((char)10);

                    //CountLines - строк
                    for (int j = 0; j < CountLines; j++)
                    {
                        //CountElements - элементов
                        for (int k = 0; k < CountElements; k++)
                        {
                            Write.Write(values.Next(1000, 9999));
                            Write.Write(' ');
                        }
```

```
        using (Write = new BinaryWriter(new FileStream("Memory.ini", FileMode.Create)))
        {
            //CountSegments - блоков
            for (int i = 0; i < CountSegments; i++)
            {
                Write.Write((char)10);

                //CountLines - строк
                for (int j = 0; j < CountLines; j++)
                {
                    //CountElements - элементов
                    for (int k = 0; k < CountElements; k++)
                    {
                        Write.Write(values.Next(1000, 9999));
                        Write.Write(' ');
                    }
                    Write.Write((char)10);
                }
            }
        }
    }
```

```
    void Positioning(int segment, int line, IDisposable WriteRead)
    {
        segment--;
        line--;

        int position = (segment + 1) + //Отступы м\у сегментами
            (segment * (CountLines * ((CountElements * 4) + 5))) +  //Пропуск эл. до нужного сегмента
                (line * ((CountElements * 4) + 5)); //Пропуск эл. до нужной строки

        //Позиция каретки с учетом размеров
        switch (WriteRead)
        {
            case BinaryWriter writer:
                writer.BaseStream.Position = position;
                break;
            case BinaryReader reader:
                reader.BaseStream.Position = position;
                break;
        }

    }
```

```
    //Считать строку line в сегменте segment
    public int[] GetLine(int segment, int line)
    {
        int[] temp = new int[CountElements];

        using (Read = new BinaryReader(new FileStream("Memory.ini", FileMode.Open)))
        {
            Positioning(segment, line, Read);

            for (int i = 0; i < CountElements; i++)
            {
                temp[i] = Read.ReadInt32();
                Read.BaseStream.Position++;
            }
        }

        return temp;
    }

    //Записать строку temp в строку line в сегменте segment
    public void SetLine(int segment, int line, int[] temp)
    {
        using (Write = new BinaryWriter(new FileStream("Memory.ini", FileMode.Open)))
        {
            Positioning(segment, line, Write);

            for (int i = 0; i < CountElements; i++)
            {
                Write.Write(temp[i]);
                Write.Write(' ');
            }
        }
    }
}
```

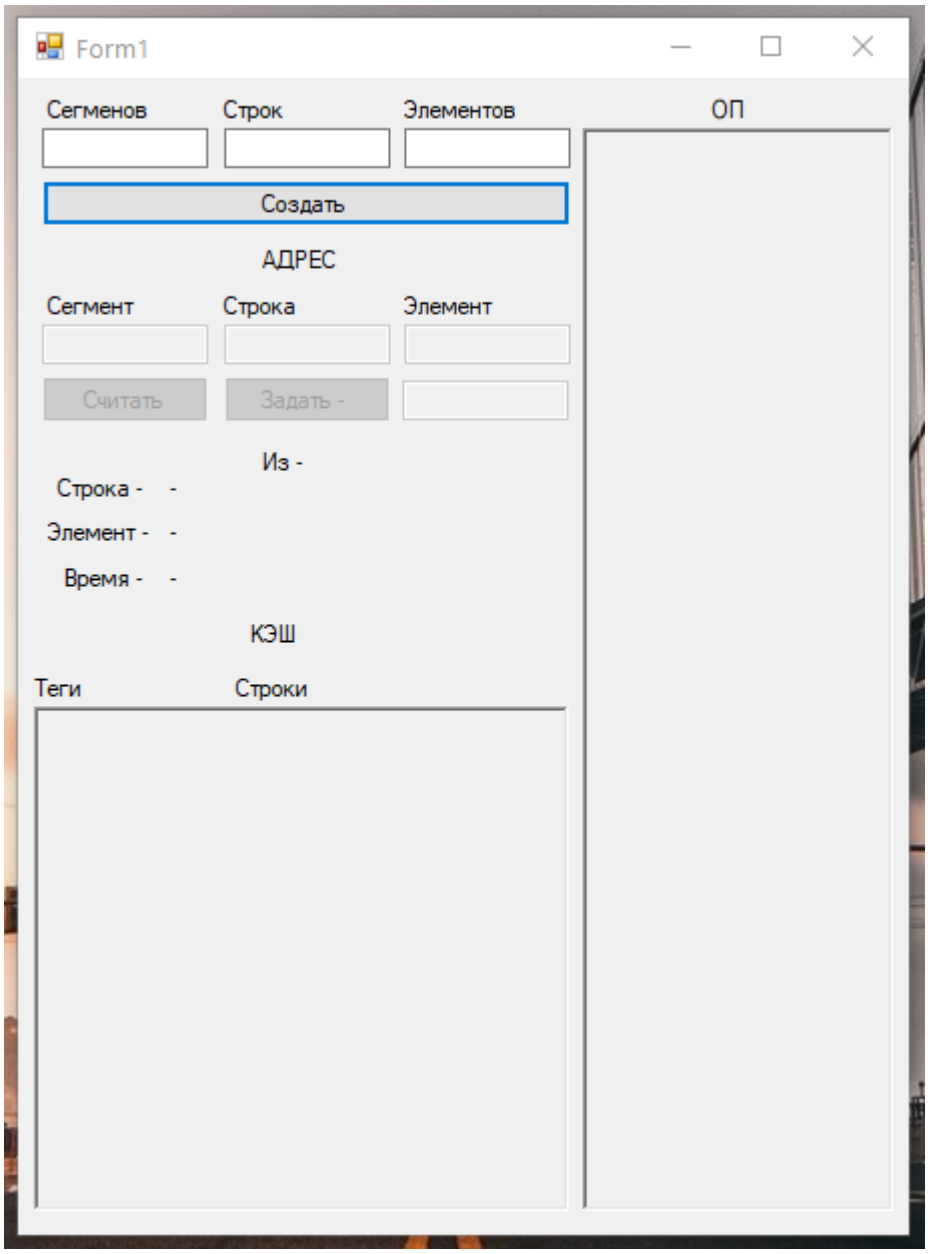
Memory.cs* Cash.cs Controller.cs CPU.cs

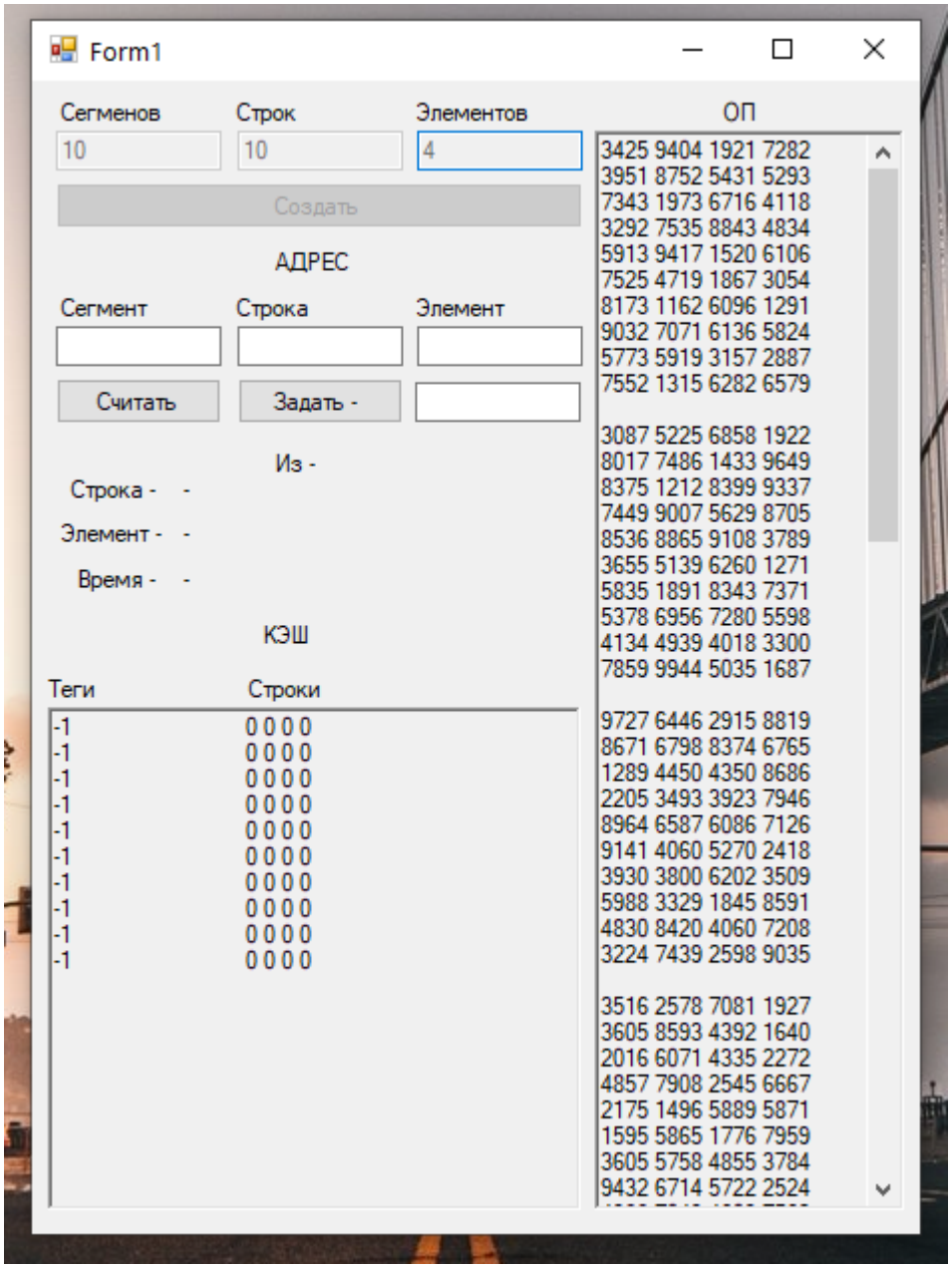
shWorking CashWorking.Cash

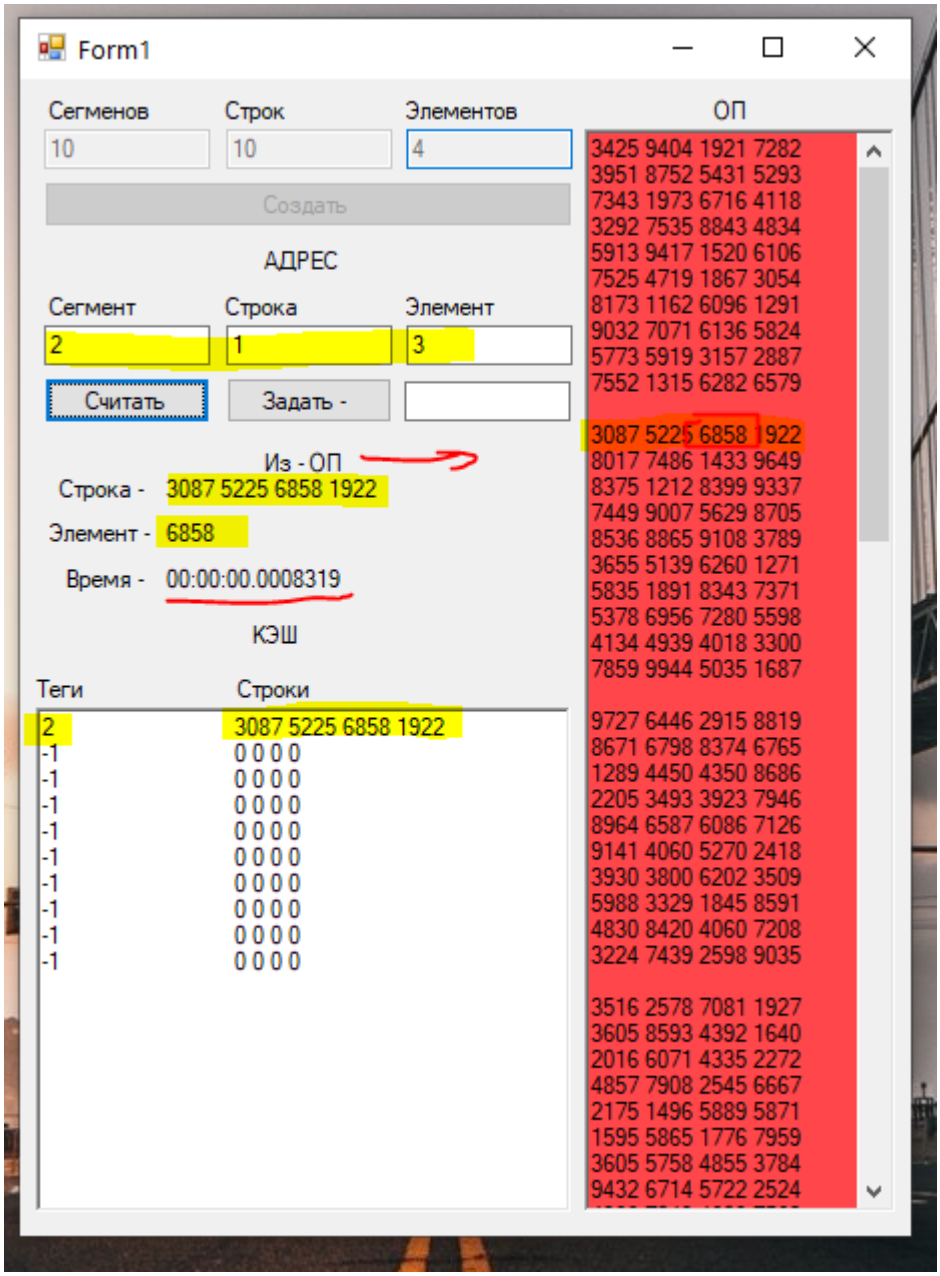
```
namespace CashWorking
{
    class Cash
    {

        (int Tag, int[] Line)[] Page;

        public Cash(int countLines, int countElements)
        {
            Page = new (int Tag, int[] Line)[countLines];
            for (int i = 0; i < Page.Length; i++)
            {
                Page[i].Line = new int[countElements];
                Page[i].Tag = -1;
            }
        }

        //i - строка
        //j - смещение
        //Считать или задать элемент
        public int this[int i, int j]
        {
            get => Page[i - 1].Line[j - 1];
            set => Page[i - 1].Line[j - 1] = value;
        }

        //Считать или задать строку
        public int[] this[int i]
        {
            get => Page[i - 1].Line;
            set => Page[i - 1].Line = value;
        }

        //Работа с тегами
        public int GetTag(int i) => Page[i - 1].Tag;
        public void SetTag(int i, int newTag) => Page[i - 1].Tag = newTag;
    }
}
```

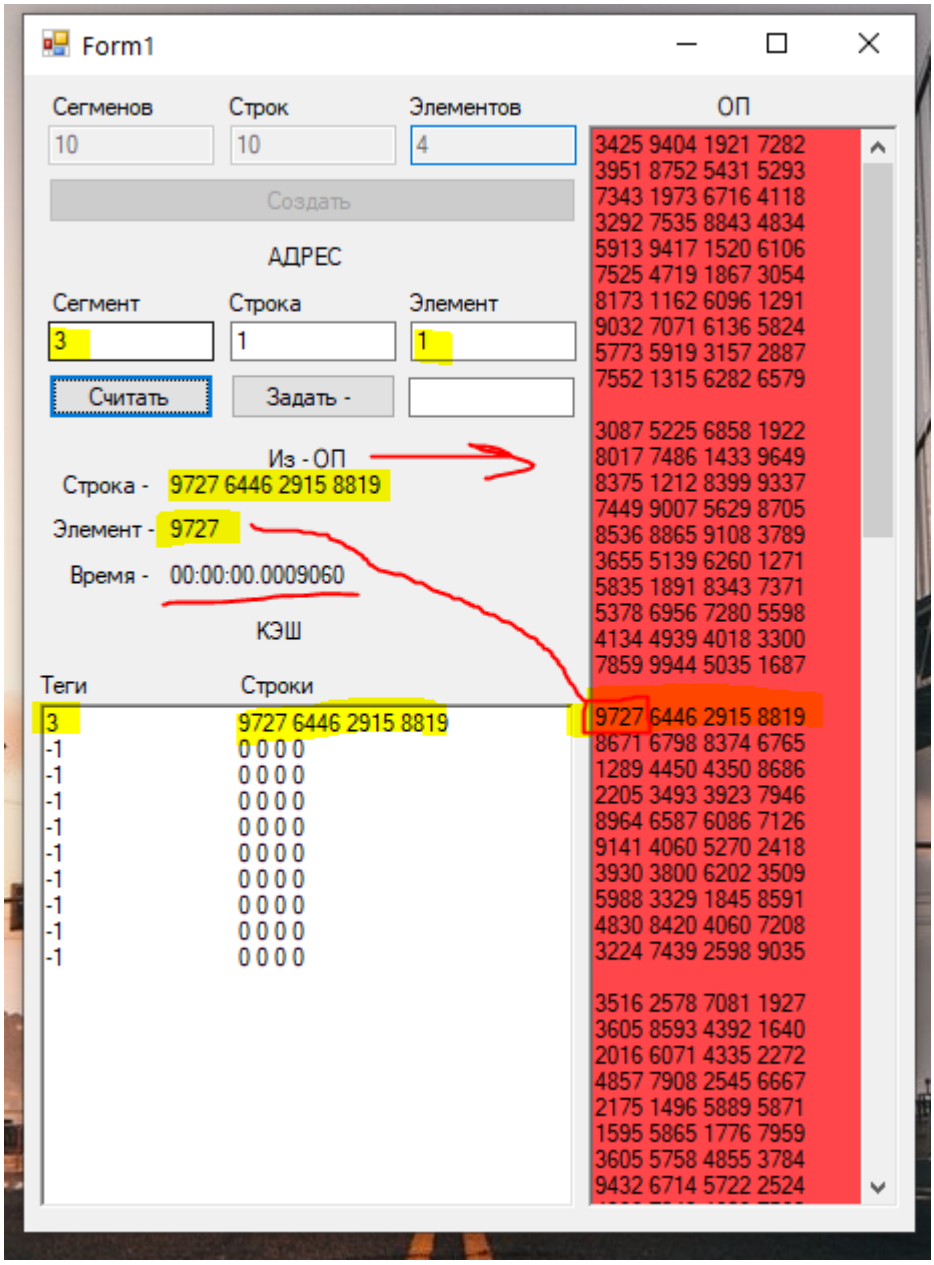
```
hWorking                                                    CashWorking.Controller
namespace CashWorking
{
    class Controller
    {
        static MainMemory MainMemo;
        static Cash L;
        int CountLines;

        public Controller(int countSegments, int countLines, int countElements)
        {
            CountLines = countLines;
            MainMemo = new MainMemory(countSegments, countLines, countElements);
            L = new Cash(countLines, countElements);
        }

        //i - сегмент
        //j - строка
        //k - смещение
        public int this[int i, int j, int k]
        {
            get
            {
                DirectMaping(i, j);
                return L[j, k];
            }
            set
            {
                DirectMaping(i, j);
                L[j, k] = value;
            }
        }

        //Проверяет на совпадение требуемый и текущий теги строки
        void DirectMaping(int i, int j)
        {
            if (i != L.GetTag(j))
            {
                if (L.GetTag(j) != -1)
                    MainMemo.SetLine(L.GetTag(j), j, L[j]);

                L[j] = MainMemo.GetLine(i, j);
                L.SetTag(j, i);
            }
        }

        //Записать из кэша все данные в память
        //и очистить теги
        public void ClearCash()
        {
            for (int i = 1; i <= CountLines; i++)
            {
                MainMemo.SetLine(L.GetTag(i), i, L[i]);
                L.SetTag(i, -1);
            }
        }
    }
}
```

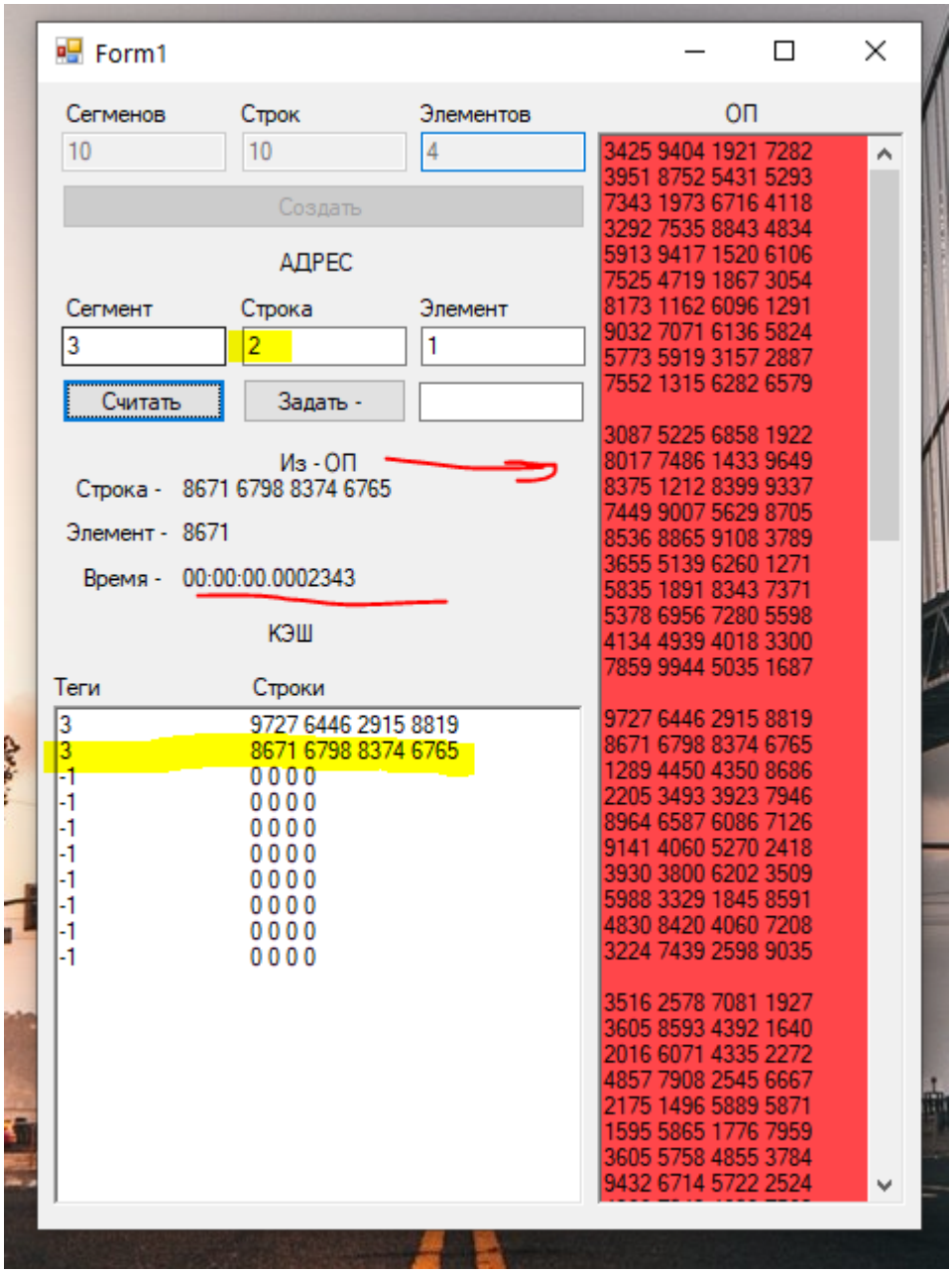
Приложение

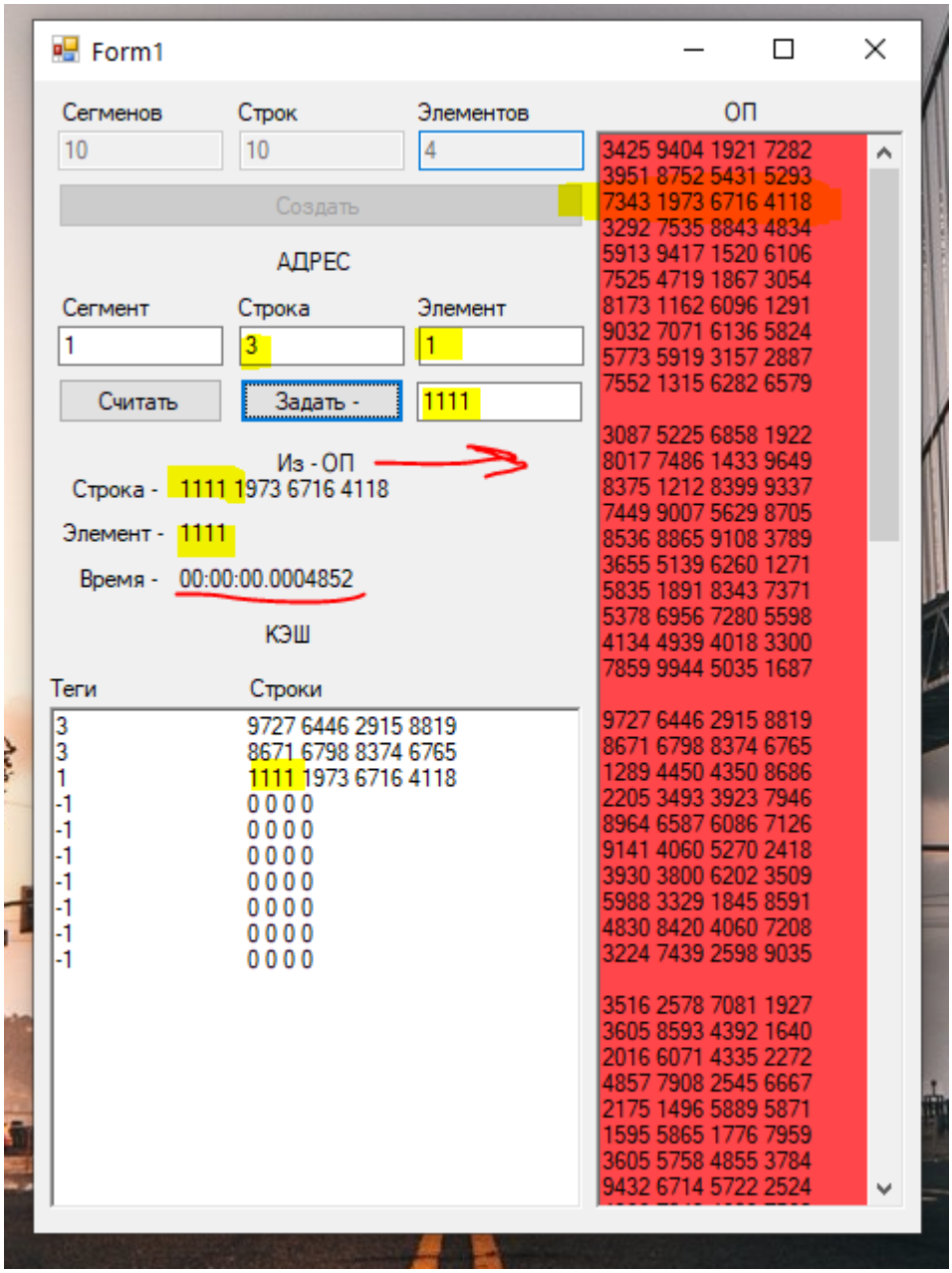

Form1
Сегменов
Строк
Элементов
ОП
Создать
АДРЕС
Сегмент
Строка
Элемент
Считать
Задать -
Из -
Строка -
Элемент -
Время -
КЭШ
Теги
Строки

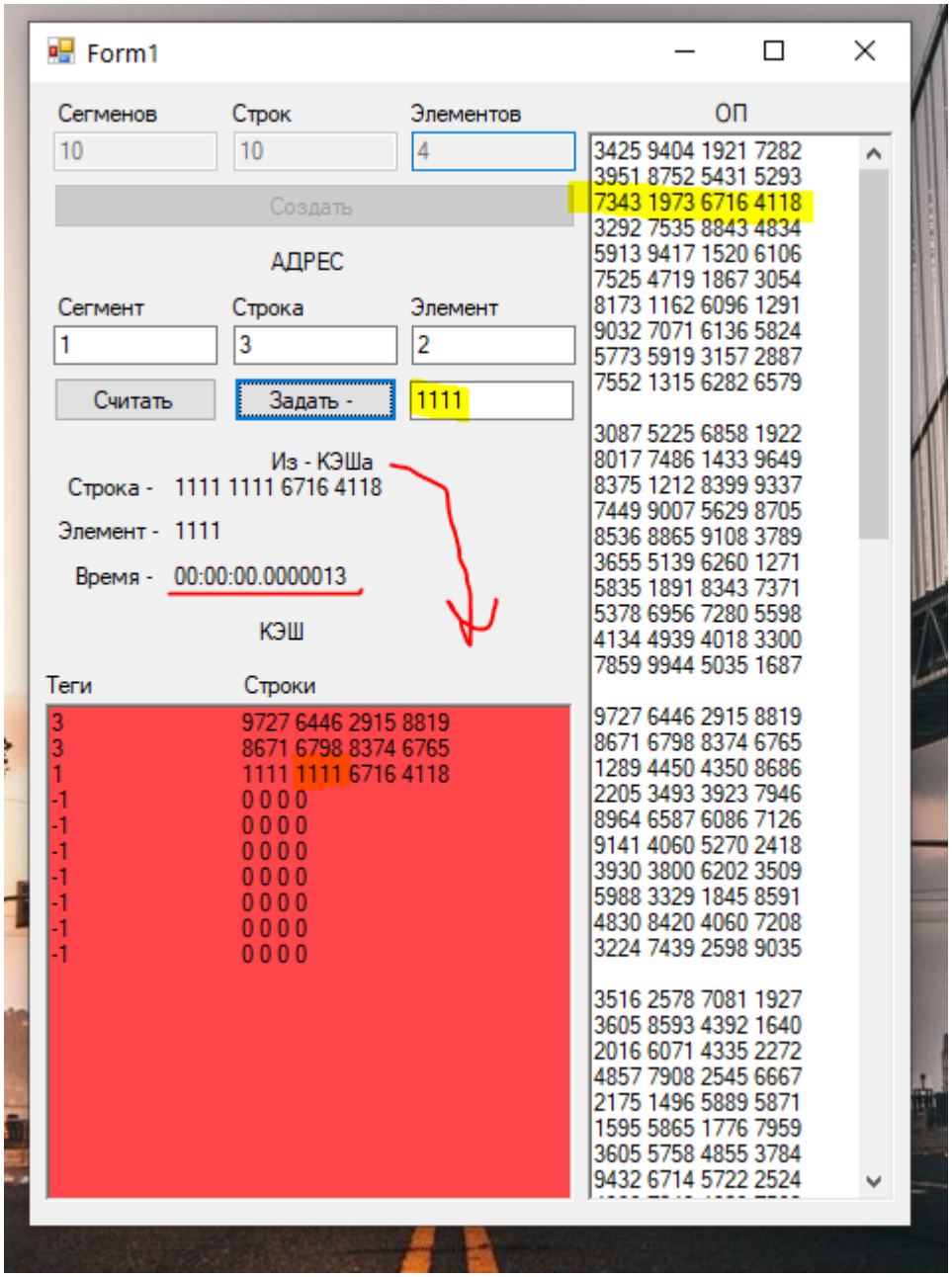

Form1
Сегменов
Строк
Элементов
10
10
4
Создать
АДРЕС
Сегмент
Строка
Элемент
Считать
Задать -
Из -
Строка -
Элемент -
Время -
КЭШ
Теги
Строки
-1 0 0 0 0
ОП
3425 9404 1921 7282
3951 8752 5431 5293
7343 1973 6716 4118
3292 7535 8843 4834
5913 9417 1520 6106
7525 4719 1867 3054
8173 1162 6096 1291
9032 7071 6136 5824
5773 5919 3157 2887
7552 1315 6282 6579
3087 5225 6858 1922
8017 7486 1433 9649
8375 1212 8399 9337
7449 9007 5629 8705
8536 8865 9108 3789
3655 5139 6260 1271
5835 1891 8343 7371
5378 6956 7280 5598
4134 4939 4018 3300
7859 9944 5035 1687
9727 6446 2915 8819
8671 6798 8374 6765
1289 4450 4350 8686
2205 3493 3923 7946
8964 6587 6086 7126
9141 4060 5270 2418
3930 3800 6202 3509
5988 3329 1845 8591
4830 8420 4060 7208
3224 7439 2598 9035
3516 2578 7081 1927
3605 8593 4392 1640
2016 6071 4335 2272
4857 7908 2545 6667
2175 1496 5889 5871
1595 5865 1776 7959
3605 5758 4855 3784
9432 6714 5722 2524

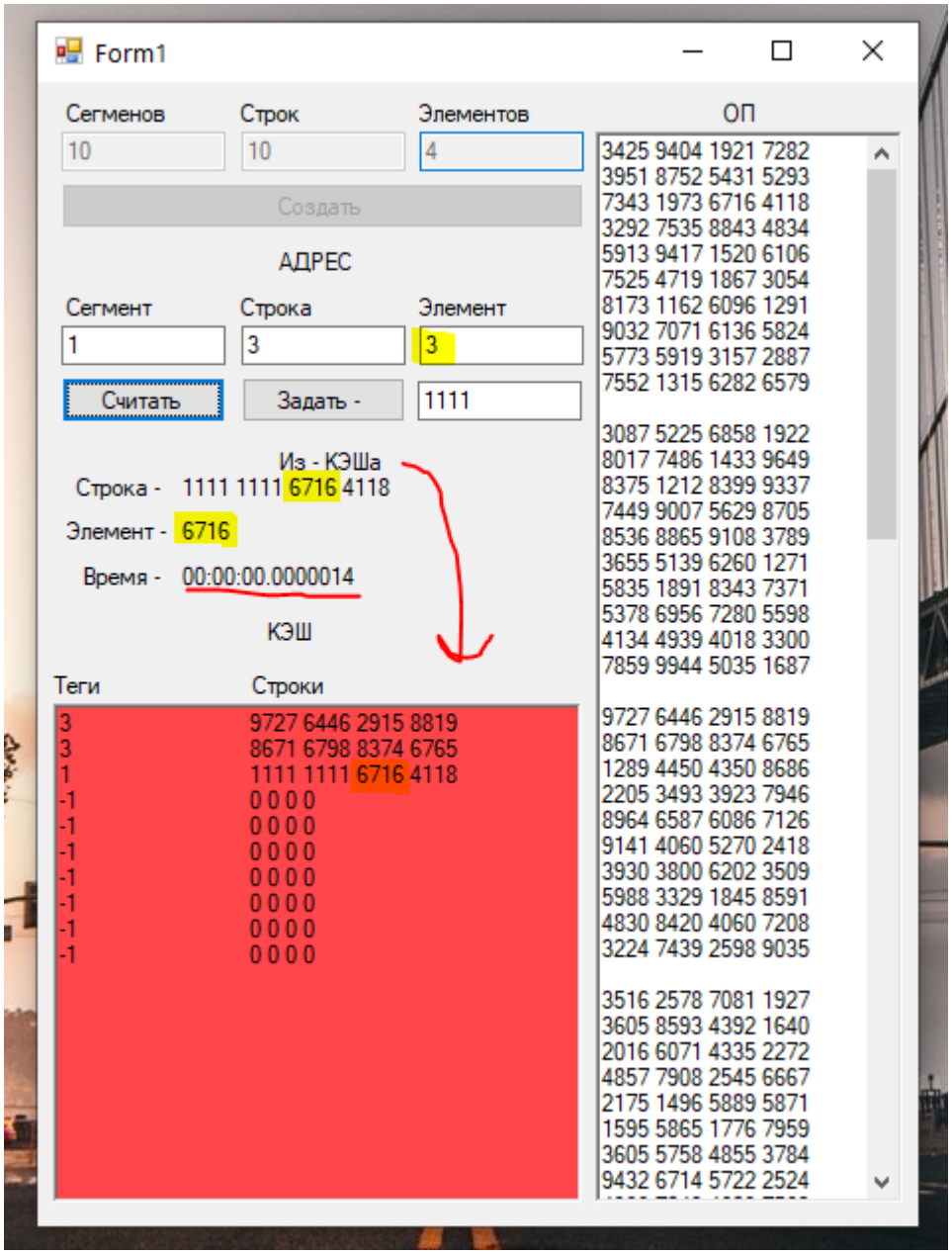

Form1
Сегменов
Строк
Элементов
10
10
4
Создать
АДРЕС
Сегмент
Строка
Элемент
2
1
3
Считать
Задать -
Из - ОП
Строка - 3087 5225 6858 1922
Элемент - 6858
Время - 00:00:00.0008319
КЭШ
Теги
Строки
2 3087 5225 6858 1922
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
ОП
3425 9404 1921 7282
3951 8752 5431 5293
7343 1973 6716 4118
3292 7535 8843 4834
5913 9417 1520 6106
7525 4719 1867 3054
8173 1162 6096 1291
9032 7071 6136 5824
5773 5919 3157 2887
7552 1315 6282 6579
3087 5225 6858 1922
8017 7486 1433 9649
8375 1212 8399 9337
7449 9007 5629 8705
8536 8865 9108 3789
3655 5139 6260 1271
5835 1891 8343 7371
5378 6956 7280 5598
4134 4939 4018 3300
7859 9944 5035 1687
9727 6446 2915 8819
8671 6798 8374 6765
1289 4450 4350 8686
2205 3493 3923 7946
8964 6587 6086 7126
9141 4060 5270 2418
3930 3800 6202 3509
5988 3329 1845 8591
4830 8420 4060 7208
3224 7439 2598 9035
3516 2578 7081 1927
3605 8593 4392 1640
2016 6071 4335 2272
4857 7908 2545 6667
2175 1496 5889 5871
1595 5865 1776 7959
3605 5758 4855 3784
9432 6714 5722 2524

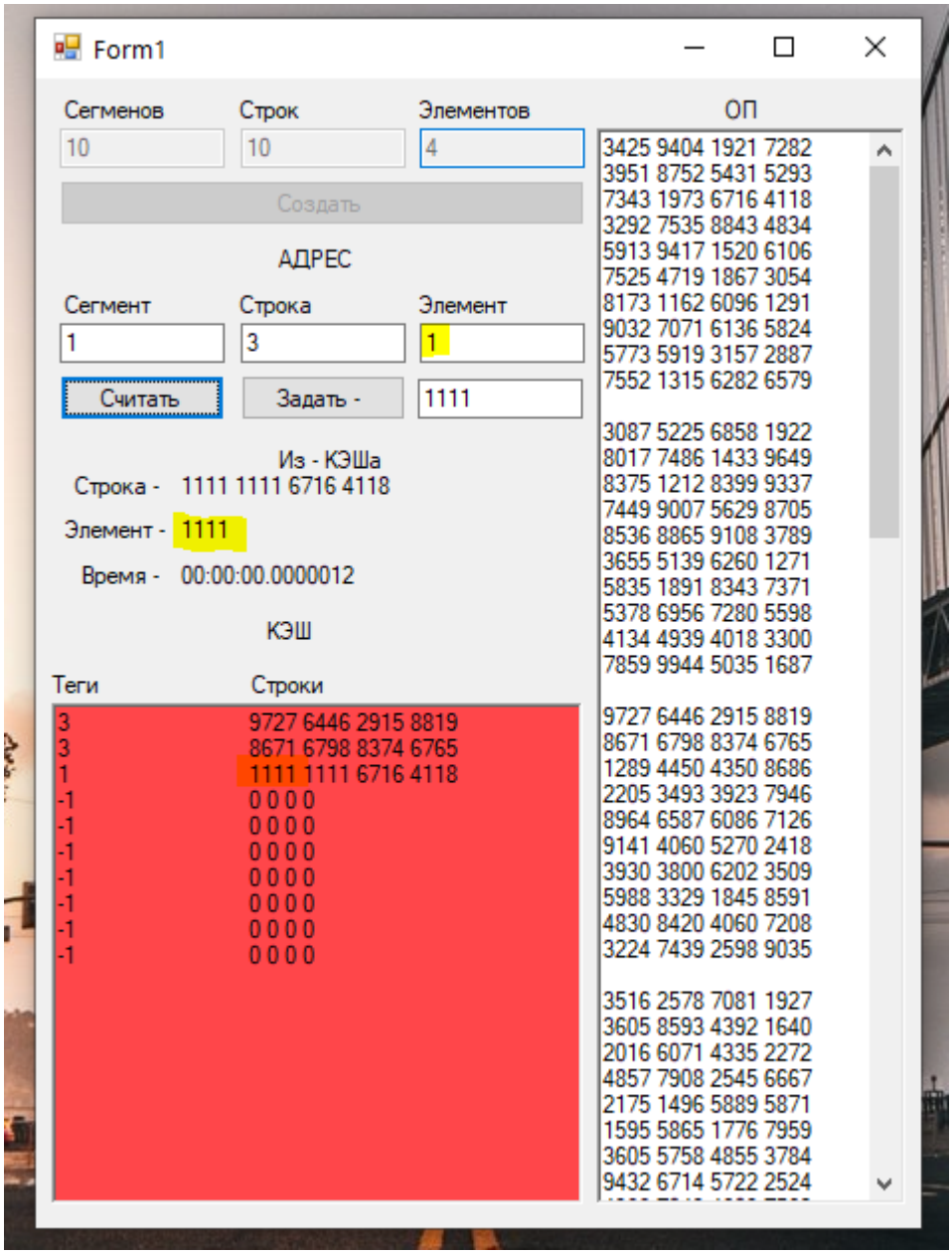

Form1
Сегменов
Строк
Элементов
10
10
4
Создать
АДРЕС
Сегмент
Строка
Элемент
3
1
1
Считать
Задать -
Из - ОП
Строка - 9727 6446 2915 8819
Элемент - 9727
Время - 00:00:00.0009060
КЭШ
Теги
Строки
3 9727 6446 2915 8819
-1 0 0 0 0
ОП
3425 9404 1921 7282
3951 8752 5431 5293
7343 1973 6716 4118
3292 7535 8843 4834
5913 9417 1520 6106
7525 4719 1867 3054
8173 1162 6096 1291
9032 7071 6136 5824
5773 5919 3157 2887
7552 1315 6282 6579
3087 5225 6858 1922
8017 7486 1433 9649
8375 1212 8399 9337
7449 9007 5629 8705
8536 8865 9108 3789
3655 5139 6260 1271
5835 1891 8343 7371
5378 6956 7280 5598
4134 4939 4018 3300
7859 9944 5035 1687
9727 6446 2915 8819
8671 6798 8374 6765
1289 4450 4350 8686
2205 3493 3923 7946
8964 6587 6086 7126
9141 4060 5270 2418
3930 3800 6202 3509
5988 3329 1845 8591
4830 8420 4060 7208
3224 7439 2598 9035
3516 2578 7081 1927
3605 8593 4392 1640
2016 6071 4335 2272
4857 7908 2545 6667
2175 1496 5889 5871
1595 5865 1776 7959
3605 5758 4855 3784
9432 6714 5722 2524

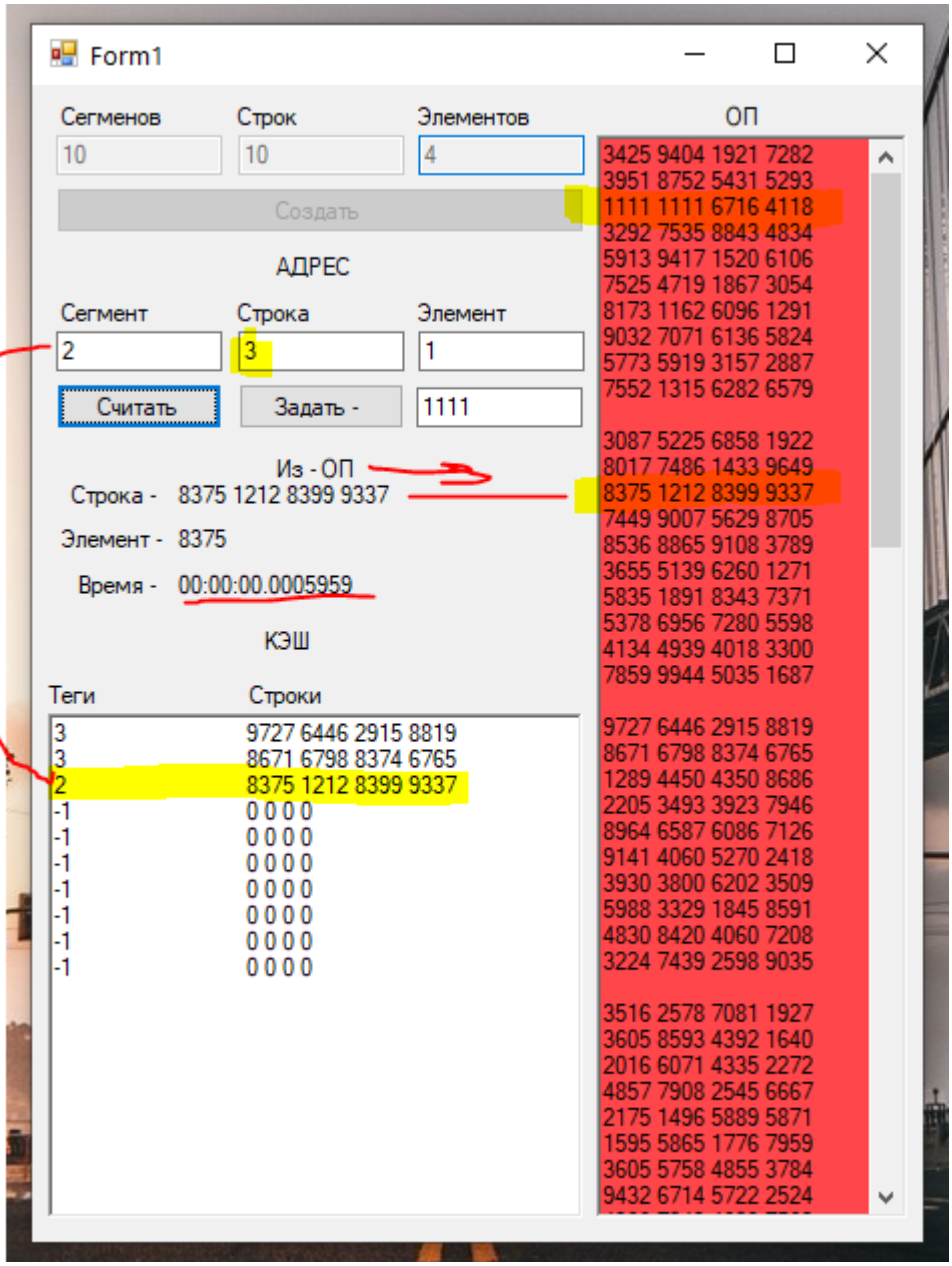

Form1
Сегменов
Строк
Элементов
10
10
4
Создать
АДРЕС
Сегмент
Строка
Элемент
3
2
1
Считать
Задать -
Из - ОП
Строка - 8671 6798 8374 6765
Элемент - 8671
Время - 00:00:00.0002343
КЭШ
Теги
Строки
3 9727 6446 2915 8819
3 8671 6798 8374 6765
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
ОП
3425 9404 1921 7282
3951 8752 5431 5293
7343 1973 6716 4118
3292 7535 8843 4834
5913 9417 1520 6106
7525 4719 1867 3054
8173 1162 6096 1291
9032 7071 6136 5824
5773 5919 3157 2887
7552 1315 6282 6579
3087 5225 6858 1922
8017 7486 1433 9649
8375 1212 8399 9337
7449 9007 5629 8705
8536 8865 9108 3789
3655 5139 6260 1271
5835 1891 8343 7371
5378 6956 7280 5598
4134 4939 4018 3300
7859 9944 5035 1687
9727 6446 2915 8819
8671 6798 8374 6765
1289 4450 4350 8686
2205 3493 3923 7946
8964 6587 6086 7126
9141 4060 5270 2418
3930 3800 6202 3509
5988 3329 1845 8591
4830 8420 4060 7208
3224 7439 2598 9035
3516 2578 7081 1927
3605 8593 4392 1640
2016 6071 4335 2272
4857 7908 2545 6667
2175 1496 5889 5871
1595 5865 1776 7959
3605 5758 4855 3784
9432 6714 5722 2524


Form1
Сегменов
Строк
Элементов
10
10
4
Создать
АДРЕС
Сегмент
Строка
Элемент
1
3
1
Считать
Задать -
1111
Из - ОП
Строка - 1111 1973 6716 4118
Элемент - 1111
Время - 00:00:00.0004852
КЭШ
Теги
Строки
3 9727 6446 2915 8819
3 8671 6798 8374 6765
1 1111 1973 6716 4118
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
ОП
3425 9404 1921 7282
3951 8752 5431 5293
7343 1973 6716 4118
3292 7535 8843 4834
5913 9417 1520 6106
7525 4719 1867 3054
8173 1162 6096 1291
9032 7071 6136 5824
5773 5919 3157 2887
7552 1315 6282 6579
3087 5225 6858 1922
8017 7486 1433 9649
8375 1212 8399 9337
7449 9007 5629 8705
8536 8865 9108 3789
3655 5139 6260 1271
5835 1891 8343 7371
5378 6956 7280 5598
4134 4939 4018 3300
7859 9944 5035 1687
9727 6446 2915 8819
8671 6798 8374 6765
1289 4450 4350 8686
2205 3493 3923 7946
8964 6587 6086 7126
9141 4060 5270 2418
3930 3800 6202 3509
5988 3329 1845 8591
4830 8420 4060 7208
3224 7439 2598 9035
3516 2578 7081 1927
3605 8593 4392 1640
2016 6071 4335 2272
4857 7908 2545 6667
2175 1496 5889 5871
1595 5865 1776 7959
3605 5758 4855 3784
9432 6714 5722 2524


Form1
Сегменов
Строк
Элементов
10
10
4
Создать
АДРЕС
Сегмент
Строка
Элемент
1
3
2
Считать
Задать -
1111
Из - КЭШа
Строка - 1111 1111 6716 4118
Элемент - 1111
Время - 00:00:00.0000013
КЭШ
Теги
Строки
3 9727 6446 2915 8819
3 8671 6798 8374 6765
1 1111 1111 6716 4118
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
ОП
3425 9404 1921 7282
3951 8752 5431 5293
7343 1973 6716 4118
3292 7535 8843 4834
5913 9417 1520 6106
7525 4719 1867 3054
8173 1162 6096 1291
9032 7071 6136 5824
5773 5919 3157 2887
7552 1315 6282 6579
3087 5225 6858 1922
8017 7486 1433 9649
8375 1212 8399 9337
7449 9007 5629 8705
8536 8865 9108 3789
3655 5139 6260 1271
5835 1891 8343 7371
5378 6956 7280 5598
4134 4939 4018 3300
7859 9944 5035 1687
9727 6446 2915 8819
8671 6798 8374 6765
1289 4450 4350 8686
2205 3493 3923 7946
8964 6587 6086 7126
9141 4060 5270 2418
3930 3800 6202 3509
5988 3329 1845 8591
4830 8420 4060 7208
3224 7439 2598 9035
3516 2578 7081 1927
3605 8593 4392 1640
2016 6071 4335 2272
4857 7908 2545 6667
2175 1496 5889 5871
1595 5865 1776 7959
3605 5758 4855 3784
9432 6714 5722 2524


Form1
Сегменов
Строк
Элементов
10
10
4
Создать
АДРЕС
Сегмент
Строка
Элемент
1
3
3
Считать
Задать -
1111
Из - КЭШа
Строка - 1111 1111 6716 4118
Элемент - 6716
Время - 00:00:00.0000014
КЭШ
Теги
Строки
3 9727 6446 2915 8819
3 8671 6798 8374 6765
1 1111 1111 6716 4118
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
ОП
3425 9404 1921 7282
3951 8752 5431 5293
7343 1973 6716 4118
3292 7535 8843 4834
5913 9417 1520 6106
7525 4719 1867 3054
8173 1162 6096 1291
9032 7071 6136 5824
5773 5919 3157 2887
7552 1315 6282 6579
3087 5225 6858 1922
8017 7486 1433 9649
8375 1212 8399 9337
7449 9007 5629 8705
8536 8865 9108 3789
3655 5139 6260 1271
5835 1891 8343 7371
5378 6956 7280 5598
4134 4939 4018 3300
7859 9944 5035 1687
9727 6446 2915 8819
8671 6798 8374 6765
1289 4450 4350 8686
2205 3493 3923 7946
8964 6587 6086 7126
9141 4060 5270 2418
3930 3800 6202 3509
5988 3329 1845 8591
4830 8420 4060 7208
3224 7439 2598 9035
3516 2578 7081 1927
3605 8593 4392 1640
2016 6071 4335 2272
4857 7908 2545 6667
2175 1496 5889 5871
1595 5865 1776 7959
3605 5758 4855 3784
9432 6714 5722 2524


Form1
Сегменов
Строк
Элементов
10
10
4
Создать
АДРЕС
Сегмент
Строка
Элемент
1
3
1
Считать
Задать -
1111
Из - КЭШа
Строка - 1111 1111 6716 4118
Элемент - 1111
Время - 00:00:00.0000012
КЭШ
Теги
Строки
3 9727 6446 2915 8819
3 8671 6798 8374 6765
1 1111 1111 6716 4118
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
ОП
3425 9404 1921 7282
3951 8752 5431 5293
7343 1973 6716 4118
3292 7535 8843 4834
5913 9417 1520 6106
7525 4719 1867 3054
8173 1162 6096 1291
9032 7071 6136 5824
5773 5919 3157 2887
7552 1315 6282 6579
3087 5225 6858 1922
8017 7486 1433 9649
8375 1212 8399 9337
7449 9007 5629 8705
8536 8865 9108 3789
3655 5139 6260 1271
5835 1891 8343 7371
5378 6956 7280 5598
4134 4939 4018 3300
7859 9944 5035 1687
9727 6446 2915 8819
8671 6798 8374 6765
1289 4450 4350 8686
2205 3493 3923 7946
8964 6587 6086 7126
9141 4060 5270 2418
3930 3800 6202 3509
5988 3329 1845 8591
4830 8420 4060 7208
3224 7439 2598 9035
3516 2578 7081 1927
3605 8593 4392 1640
2016 6071 4335 2272
4857 7908 2545 6667
2175 1496 5889 5871
1595 5865 1776 7959
3605 5758 4855 3784
9432 6714 5722 2524


Form1
Сегменов
Строк
Элементов
10
10
4
Создать
АДРЕС
Сегмент
Строка
Элемент
2
3
1
Считать
Задать -
1111
Из - ОП
Строка - 8375 1212 8399 9337
Элемент - 8375
Время - 00:00:00.0005959
КЭШ
Теги
Строки
3 9727 6446 2915 8819
3 8671 6798 8374 6765
2 8375 1212 8399 9337
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
-1 0 0 0 0
ОП
3425 9404 1921 7282
3951 8752 5431 5293
1111 1111 6716 4118
3292 7535 8843 4834
5913 9417 1520 6106
7525 4719 1867 3054
8173 1162 6096 1291
9032 7071 6136 5824
5773 5919 3157 2887
7552 1315 6282 6579
3087 5225 6858 1922
8017 7486 1433 9649
8375 1212 8399 9337
7449 9007 5629 8705
8536 8865 9108 3789
3655 5139 6260 1271
5835 1891 8343 7371
5378 6956 7280 5598
4134 4939 4018 3300
7859 9944 5035 1687
9727 6446 2915 8819
8671 6798 8374 6765
1289 4450 4350 8686
2205 3493 3923 7946
8964 6587 6086 7126
9141 4060 5270 2418
3930 3800 6202 3509
5988 3329 1845 8591
4830 8420 4060 7208
3224 7439 2598 9035
3516 2578 7081 1927
3605 8593 4392 1640
2016 6071 4335 2272
4857 7908 2545 6667
2175 1496 5889 5871
1595 5865 1776 7959
3605 5758 4855 3784
9432 6714 5722 2524