/\*using System;

using System.Diagnostics;

namespace z1

{

class Timing

{

TimeSpan a;

TimeSpan[] b;

public Timing()

{

a = new TimeSpan(0);

b = new TimeSpan[Process.GetCurrentProcess().

Threads.Count];

}

public void StartTime()

{

GC.Collect();

GC.WaitForPendingFinalizers();

for (int i = 0; i < b.Length; i++)

b[i] = Process.GetCurrentProcess().Threads[i].

UserProcessorTime;

}

public void StopTime()

{

TimeSpan tmp;

for (int i = 0; i < b.Length; i++)

{

tmp = Process.GetCurrentProcess().Threads[i].

UserProcessorTime.Subtract(b[i]);

if (tmp > TimeSpan.Zero)

a = tmp;

}

}

public TimeSpan Result()

{

return a;

}

}

class Arrs

{

public static Random r = new Random();

public static int[] C(int[] Mas)

{

for (int i = 0; i < Mas.Length; i++)

Mas[i] = r.Next(0, 100);

return Mas;

}

public static void PrintArr(string name, int[] Mas)

{

Console.Write(" Массив " + name);

for (int i = 0; i < Mas.Length; i++) Console.Write(" " + Mas[i]);

Console.WriteLine();

}

public static int[] Metodfkl(int[] Mas)

{

for (int i = 1; i < Mas.Length; i++)

{

int l = Mas[i];

int j;

for (j = i - 1; j >= 0 && Mas[j] > l; j--)

Mas[j + 1] = Mas[j];

Mas[j + 1] = l;

}

return Mas;

}

static void Swap(ref int a, ref int b)

{

var t = a;

a = b;

b = t;

}

public static int[] MetodShella(int[] Mas)

{

var d = Mas.Length / 2;

while (d >= 1)

{

for (var i = d; i < Mas.Length; i++)

{

var j = i;

while ((j >= d) && (Mas[j - d] > Mas[j]))

{

Swap(ref Mas[j], ref Mas[j - d]);

j = j - d;

}

}

d = d / 2;

}

return Mas;

}

}

class Program

{

static void Main()

{

int[] A = new int[10], B = new int[50];

Arrs.C(A);

Arrs.C(B);

Timing d = new Timing();

Stopwatch stpWatch = new Stopwatch();

d.StartTime();

stpWatch.Start();

Arrs.MetodShella(A);

stpWatch.Stop();

d.StopTime();

Console.WriteLine("Время сортировки массива на 10 элементов методом Шелла:");

Console.WriteLine("StopWatch: " + stpWatch.Elapsed.ToString());

Console.WriteLine("Timing: " + d.Result().ToString());

Timing d1 = new Timing();

Stopwatch stpWatch1 = new Stopwatch();

d1.StartTime();

stpWatch1.Start();

Arrs.MetodShella(B);

stpWatch1.Stop();

d1.StopTime();

Console.WriteLine("Время сортировки массива на 50 элементов методом Шелла:");

Console.WriteLine("StopWatch: " + stpWatch1.Elapsed.ToString());

Console.WriteLine("Timing: " + d1.Result().ToString());

Timing d2 = new Timing();

Stopwatch stpWatch2 = new Stopwatch();

d2.StartTime();

stpWatch2.Start();

Arrs.Metodfkl(A);

stpWatch2.Stop();

d2.StopTime();

Console.WriteLine("Время сортировки массива на 10 элементов методом включения:");

Console.WriteLine("StopWatch: " + stpWatch2.Elapsed.ToString());

Console.WriteLine("Timing: " + d2.Result().ToString());

Timing d3 = new Timing();

Stopwatch stpWatch3 = new Stopwatch();

d3.StartTime();

stpWatch3.Start();

Arrs.Metodfkl(B);

stpWatch3.Stop();

d3.StopTime();

Console.WriteLine("Время сортировки массива на 50 элементов методом включения:");

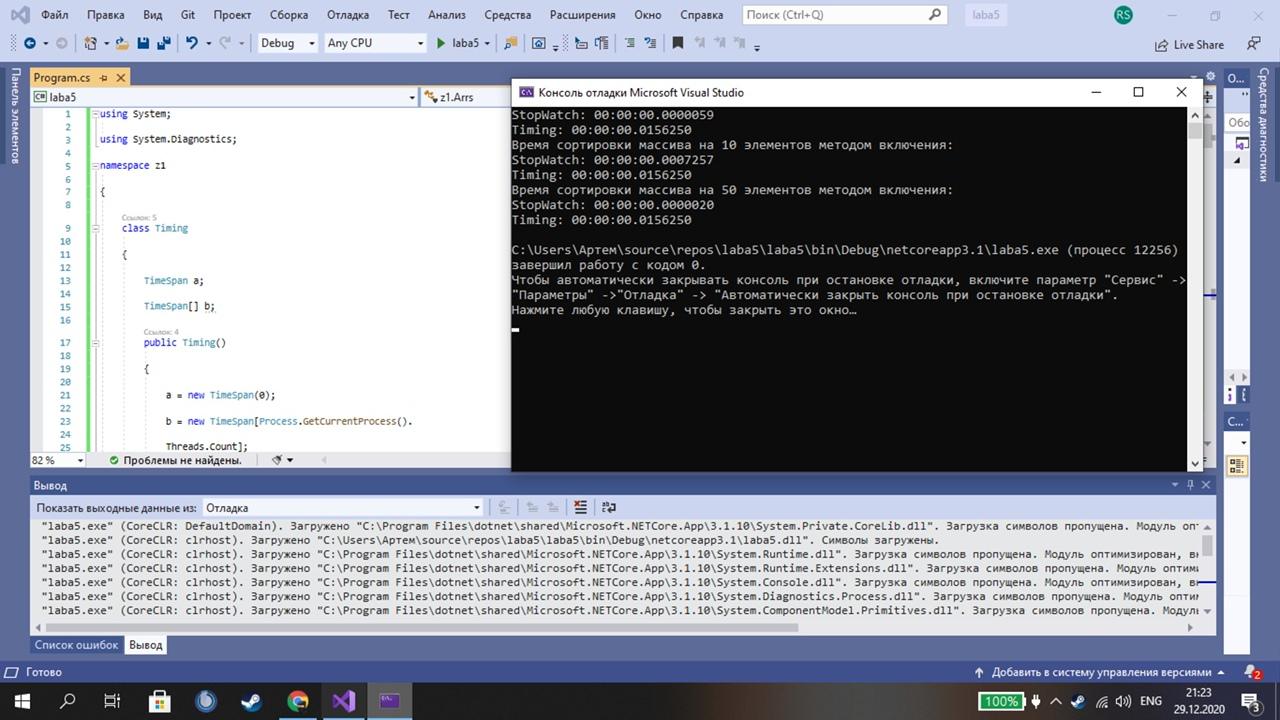
Console.WriteLine("StopWatch: " + stpWatch3.Elapsed.ToString());

Console.WriteLine("Timing: " + d3.Result().ToString());

}

}

}



using System;

using System.ComponentModel;

using System.Diagnostics;

namespace z2

{

class Timing

{

TimeSpan ris;

TimeSpan[] threads;

public Timing()

{

ris = new TimeSpan(0);

threads = new TimeSpan[Process.GetCurrentProcess().

Threads.Count];

}

public void StartTime()

{

GC.Collect();

GC.WaitForPendingFinalizers();

for (int i = 0; i < threads.Length; i++)

threads[i] = Process.GetCurrentProcess().Threads[i].

UserProcessorTime;

}

public void StopTime()

{

TimeSpan tmp;

for (int i = 0; i < threads.Length; i++)

{

tmp = Process.GetCurrentProcess().Threads[i].

UserProcessorTime.Subtract(threads[i]);

if (tmp > TimeSpan.Zero)

ris = tmp;

}

}

public TimeSpan Result()

{

return ris;

}

}

class Arrs

{

public static Random r = new Random();

public static int[] CreatAr(int[] Mas)

{

for (int i = 0; i < Mas.Length; i++)

Mas[i] = r.Next(0, 100);

return Mas;

}

public static void PrintArr1(string name, int[] Mas)

{

Console.Write(" Массив " + name);

for (int i = 0; i < Mas.Length; i++) Console.Write(" " + Mas[i]);

Console.WriteLine();

}

public static int linePoisk(int b, int[] Mas)

{

int q = -1;

int counter = 0;

for (int i = 0; i < Mas.Length; i++)

{

counter++;

if (Mas[i] == b)

{

q = i; break;

}

}

return q;

}

public static int Binarn(int b, int[] Mas)

{

int q;

int L = 0;

int R = Mas.Length - 1;

q = (R + L) / 2;

int counter = 0;

while (L < R - 1)

{

counter++;

q = (R + L) / 2;

if (Mas[q] == b)

{

q = Mas[q];

return q;

}

counter++;

if (Mas[q] < b)

L = q;

else

R = q;

}

if (Mas[q] != b)

{

if (Mas[L] == b)

q = L;

else

{

if (Mas[R] == b)

q = R;

else

q = -1;

};

}

return q;

}

public static int[] Metodfkl(int[] Mas)

{

for (int i = 1; i < Mas.Length; i++)

{

int l = Mas[i];

int j;

for (j = i - 1; j >= 0 && Mas[j] > l; j--)

Mas[j + 1] = Mas[j];

Mas[j + 1] = l;

}

return Mas;

}

}

class Program

{

static void Main()

{

int[] A = new int[10], B = new int[50];

Arrs.CreatAr(A);

Arrs.CreatAr(B);

Arrs.PrintArr1("A", A);

Arrs.PrintArr1("B", B);

Arrs.Metodfkl(A);

Arrs.Metodfkl(B);

Console.Write("Введите число, которое нужно найти: ");

int b = int.Parse(Console.ReadLine());

Timing t = new Timing();

Stopwatch stpWatch = new Stopwatch();

t.StartTime();

stpWatch.Start();

Arrs.linePoisk(b, A);

stpWatch.Stop();

t.StopTime();

Console.WriteLine("Время поиска методом простого поиска в массиве на 10 элементов:");

Console.WriteLine("StopWatch: " + stpWatch.Elapsed.ToString());

Console.WriteLine("Timing: " + t.Result().ToString());

Timing t1 = new Timing();

Stopwatch stpWatch1 = new Stopwatch();

t1.StartTime();

stpWatch1.Start();

Arrs.linePoisk(b, B);

stpWatch1.Stop();

t1.StopTime();

Console.WriteLine("Время поиска методом простого поиска в массиве на 50 элементов:");

Console.WriteLine("StopWatch: " + stpWatch1.Elapsed.ToString());

Console.WriteLine("Timing: " + t1.Result().ToString());

Timing t2 = new Timing();

Stopwatch stpWatch2 = new Stopwatch();

t2.StartTime();

stpWatch2.Start();

Arrs.Binarn(b, A);

stpWatch2.Stop();

t2.StopTime();

Console.WriteLine("Время поиска методом бинарного поиска в массиве на 10 элементов:");

Console.WriteLine("StopWatch: " + stpWatch2.Elapsed.ToString());

Console.WriteLine("Timing: " + t2.Result().ToString());

Timing t3 = new Timing();

Stopwatch stpWatch3 = new Stopwatch();

t3.StartTime();

stpWatch3.Start();

Arrs.Binarn(b, B);

stpWatch3.Stop();

t3.StopTime();

Console.WriteLine("Время поиска методом бинарного поиска в массиве на 50 элементов:");

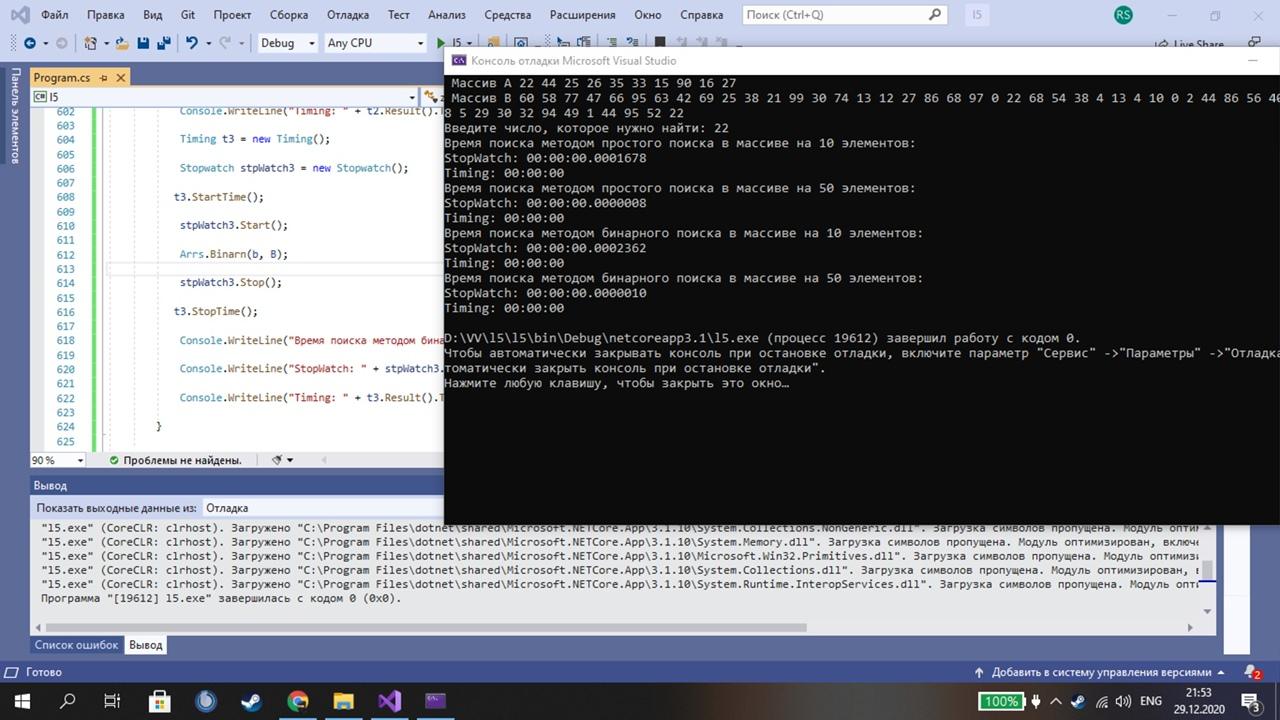
Console.WriteLine("StopWatch: " + stpWatch3.Elapsed.ToString());

Console.WriteLine("Timing: " + t3.Result().ToString());

}

}

}



using System;

using System.ComponentModel;

using System.Diagnostics;

namespace z3

{

class Timing

{

TimeSpan q;

TimeSpan[] threads;

public Timing()

{

q = new TimeSpan(0);

threads = new TimeSpan[Process.GetCurrentProcess().

Threads.Count];

}

public void StartTime()

{

GC.Collect();

GC.WaitForPendingFinalizers();

for (int i = 0; i < threads.Length; i++)

threads[i] = Process.GetCurrentProcess().Threads[i].

UserProcessorTime;

}

public void StopTime()

{

TimeSpan tmp;

for (int i = 0; i < threads.Length; i++)

{

tmp = Process.GetCurrentProcess().Threads[i].

UserProcessorTime.Subtract(threads[i]);

if (tmp > TimeSpan.Zero)

q = tmp;

}

}

public TimeSpan Result()

{

return q;

}

}

class Arrs

{

public static Random r = new Random();

public static int[] CreatAr(int[] Mas)

{

for (int i = 0; i < Mas.Length; i++)

Mas[i] = r.Next(0, 100);

return Mas;

}

public static void PrintArr1(string name, int[] Mas)

{

Console.Write(" Массив " + name);

for (int i = 0; i < Mas.Length; i++) Console.Write(" " + Mas[i]);

Console.WriteLine();

}

public static int Poisk(int b, int[] Mas)

{

int z = -1;

int counter = 0;

for (int i = 0; i < Mas.Length; i++)

{

counter++;

if (Mas[i] == b)

{

z = i; break;

}

}

return z;

}

public static int Binarnii(int b, int[] Mas)

{

int k;

int L = 0;

int R = Mas.Length - 1;

k = (R + L) / 2;

int counter = 0;

while (L < R - 1)

{

counter++;

k = (R + L) / 2;

if (Mas[k] == b)

{

k = Mas[k];

return k;

}

counter++;

if (Mas[k] < b)

L = k;

else

R = k;

}

if (Mas[k] != b)

{

if (Mas[L] == b)

k = L;

else

{

if (Mas[R] == b)

k = R;

else

k = -1;

};

}

return k;

}

public static int[] Metodfkl(int[] Mas)

{

for (int i = 1; i < Mas.Length; i++)

{

int l = Mas[i];

int j;

for (j = i - 1; j >= 0 && Mas[j] > l; j--)

Mas[j + 1] = Mas[j];

Mas[j + 1] = l;

}

return Mas;

}

}

class Program

{

static void Main()

{

int[] A = new int[10], B = new int[50];

Arrs.CreatAr(A);

Arrs.CreatAr(B);

Arrs.PrintArr1("A", A);

Arrs.PrintArr1("B", B);

Console.Write("Введите число, которое нужно найти: ");

int b = int.Parse(Console.ReadLine());

Timing t = new Timing();

Stopwatch stpWatch = new Stopwatch();

t.StartTime();

stpWatch.Start();

Arrs.Binarnii(b, A);

stpWatch.Stop();

t.StopTime();

Console.WriteLine("Время поиска методом бинарного поиска в массиве на 10 элементов:");

Console.WriteLine("StopWatch: " + stpWatch.Elapsed.ToString());

Console.WriteLine("Timing: " + t.Result().ToString());

Timing t1 = new Timing();

Stopwatch stpWatch1 = new Stopwatch();

t1.StartTime();

stpWatch1.Start();

Arrs.Binarnii(b, B);

stpWatch1.Stop();

t1.StopTime();

Console.WriteLine("Время поиска методом бинарного поиска в массиве на 50 элементов:");

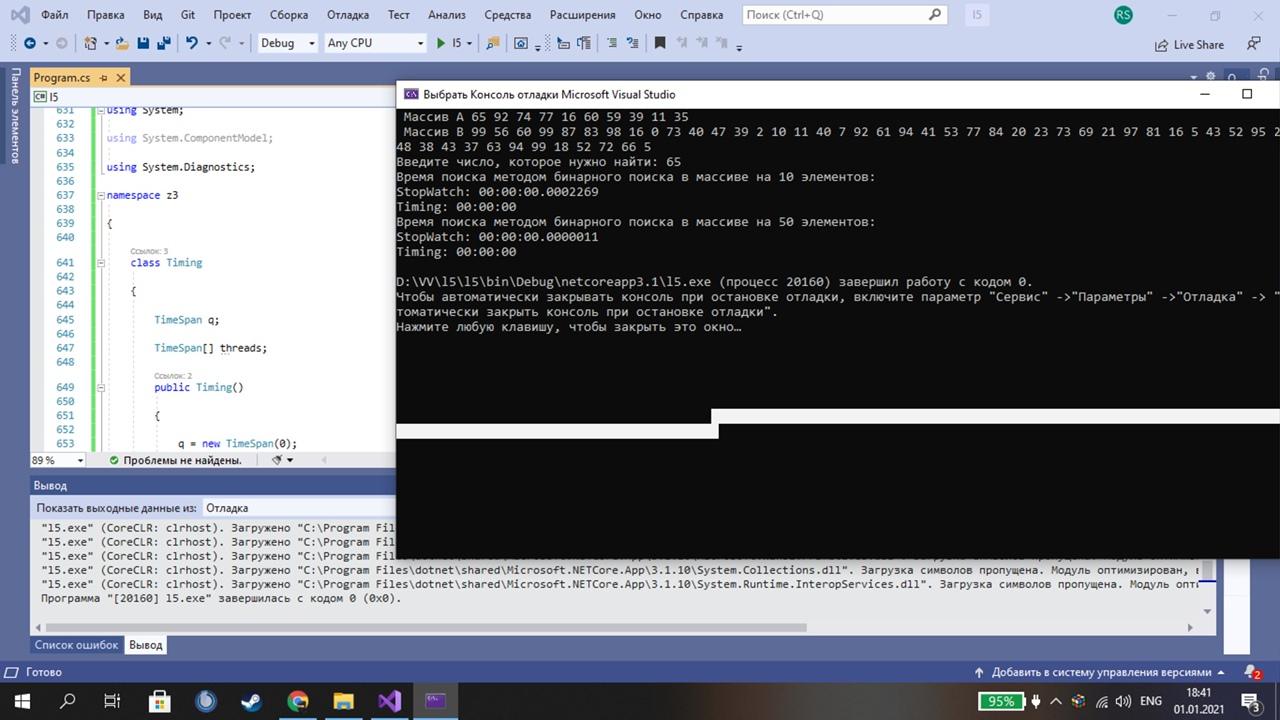
Console.WriteLine("StopWatch: " + stpWatch1.Elapsed.ToString());

Console.WriteLine("Timing: " + t1.Result().ToString());

}

}

} \*/



using System;

using System.ComponentModel;

using System.Diagnostics;

namespace z4

{

class Timing

{

TimeSpan q;

TimeSpan[] threads;

public Timing()

{

q = new TimeSpan(0);

threads = new TimeSpan[Process.GetCurrentProcess().

Threads.Count];

}

public void StartTime()

{

GC.Collect();

GC.WaitForPendingFinalizers();

for (int i = 0; i < threads.Length; i++)

threads[i] = Process.GetCurrentProcess().Threads[i].

UserProcessorTime;

}

public void StopTime()

{

TimeSpan tmp;

for (int i = 0; i < threads.Length; i++)

{

tmp = Process.GetCurrentProcess().Threads[i].

UserProcessorTime.Subtract(threads[i]);

if (tmp > TimeSpan.Zero)

q = tmp;

}

}

public TimeSpan Result()

{

return q;

}

}

class Arrs

{

public static Random r = new Random();

public static int[] CreatAr(int[] Mas)

{

for (int i = 0; i < Mas.Length; i++)

Mas[i] = r.Next(0, 100);

return Mas;

}

public static void wow(string name, int[] Mas)

{

Console.Write(" Массив " + name);

for (int i = 0; i < Mas.Length; i++) Console.Write(" " + Mas[i]);

Console.WriteLine();

}

public static int line(int b, int[] Mas)

{

int k = -1;

int counter = 0;

for (int i = 0; i < Mas.Length; i++)

{

counter++;

if (Mas[i] == b)

{

k = i; break;

}

}

return k;

}

public static int Binarn(int b, int[] Mas)

{

int z;

int L = 0;

int R = Mas.Length - 1;

z = (R + L) / 2;

int cou = 0;

while (L < R - 1)

{

cou++;

z= (R + L) / 2;

if (Mas[z] == b)

{

z = Mas[z];

return z;

}

cou++;

if (Mas[z] < b)

L = z;

else

R = z;

}

if (Mas[z] != b)

{

if (Mas[L] == b)

z = L;

{

if (Mas[R] == b)

z = R;

else

z = -1;

};

}

return z;

}

public static int[] Metodfkl(int[] Mas)

{

for (int i = 1; i < Mas.Length; i++)

{

int l = Mas[i];

int j;

for (j = i - 1; j >= 0 && Mas[j] > l; j--)

Mas[j + 1] = Mas[j];

Mas[j + 1] = l;

}

return Mas;

}

public static int Barrier(int b, int[] Mas, int \_size)

{

int counter = 0;

int position = 0;

if (Mas[\_size - 1] != b)

{

Mas[\_size - 1] = b;

for (; Mas[position] != b; position++)

counter++;

}

else return \_size;

return counter;

}

}

class Program

{

static void Main()

{

int[] A = new int[10], B = new int[50];

Arrs.CreatAr(A);

Arrs.CreatAr(B);

Arrs.wow("A", A);

Arrs.wow("B", B);

Console.Write("Введите число, которое нужно найти: ");

int b = int.Parse(Console.ReadLine());

int CounterlinePoisk = Arrs.line(b, A) + Arrs.line(b, B);

int CounterBinarn = Arrs.Binarn(b, A) + Arrs.Binarn(b, B); ;

int CounterBarrier = Arrs.Barrier(b, A, A.Length) + Arrs.Barrier(b, B, B.Length);

Timing t = new Timing();

Stopwatch stpWatch = new Stopwatch();

t.StartTime();

stpWatch.Start();

Arrs.line(b, A);

stpWatch.Stop();

t.StopTime();

Console.WriteLine("Время поиска методом простого поиска в массиве на 10 элементов:");

Console.WriteLine("StopWatch: " + stpWatch.Elapsed.ToString());

Console.WriteLine("Timing: " + t.Result().ToString());

Timing t1 = new Timing();

Stopwatch stpWatch1 = new Stopwatch();

t1.StartTime();

stpWatch1.Start();

Arrs.line(b, B);

stpWatch1.Stop();

t1.StopTime();

Console.WriteLine("Время поиска методом простого поиска в массиве на 50 элементов:");

Console.WriteLine("StopWatch: " + stpWatch1.Elapsed.ToString());

Console.WriteLine("Timing: " + t1.Result().ToString());

Timing t2 = new Timing();

Stopwatch stpWatch2 = new Stopwatch();

t2.StartTime();

stpWatch2.Start();

Arrs.Binarn(b, A);

stpWatch2.Stop();

t2.StopTime();

Console.WriteLine("Время поиска методом бинарного поиска в массиве на 10 элементов:");

Console.WriteLine("StopWatch: " + stpWatch2.Elapsed.ToString());

Console.WriteLine("Timing: " + t2.Result().ToString());

Timing t3 = new Timing();

Stopwatch stpWatch3 = new Stopwatch();

t3.StartTime();

stpWatch3.Start();

Arrs.Binarn(b, B);

stpWatch3.Stop();

t3.StopTime();

Console.WriteLine("Время поиска методом бинарного поиска в массиве на 50 элементов:");

Console.WriteLine("StopWatch: " + stpWatch3.Elapsed.ToString());

Console.WriteLine("Timing: " + t3.Result().ToString());

Timing t4 = new Timing();

Stopwatch stpWatch4 = new Stopwatch();

t4.StartTime();

stpWatch4.Start();

Arrs.Binarn(b, A);

stpWatch4.Stop();

t4.StopTime();

Console.WriteLine("Время поиска методом барьера поиска в массиве на 10 элементов:");

Console.WriteLine("StopWatch: " + stpWatch4.Elapsed.ToString());

Console.WriteLine("Timing: " + t4.Result().ToString());

Timing t5 = new Timing();

Stopwatch stpWatch5 = new Stopwatch();

t5.StartTime();

stpWatch5.Start();

Arrs.Binarn(b, B);

stpWatch5.Stop();

t5.StopTime();

Console.WriteLine("Время поиска методом барьера поиска в массиве на 50 элементов:");

Console.WriteLine("StopWatch: " + stpWatch5.Elapsed.ToString());

Console.WriteLine("Timing: " + t5.Result().ToString());

Console.WriteLine(CounterlinePoisk);

Console.WriteLine(CounterBinarn);

Console.WriteLine(CounterBarrier);

}

}

}

