

***Дисциплина: Учебная практика***

**Отчет по выполнению задания на тему: «Lesson 4».**

Выполнил:

Группа: **1ИСП-11-18**

Студент: Белякова А.А.

Ищенко Т.С.

Проверил преподаватель:

Мартиросян Ваган Артемович

Москва 2022 г.

Задание 1.

int[] nums = new int[12];

for (int k = 0; k < nums.Length; k++)

{

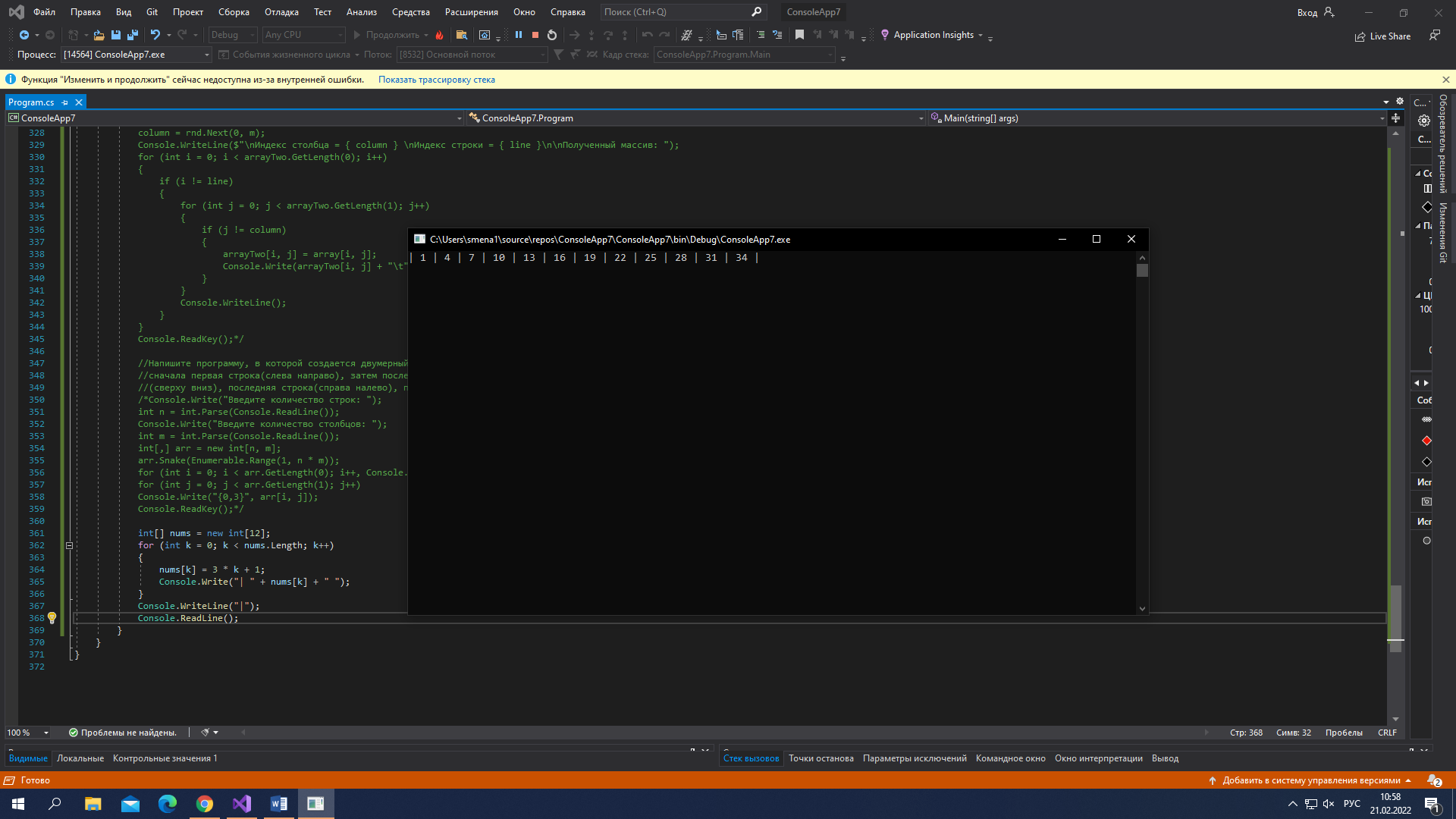
nums[k] = 3 \* k + 1;

Console.Write("| " + nums[k] + " ");

}

Console.WriteLine("|");

Console.ReadLine();



Random rnd = new Random();

int size = 10, k;

char[] symbs = new char[size];

Console.WriteLine("Массив случайных символов: ");

for (k = 0; k < symbs.Length; k++)

{

symbs[k] = (char)('A' + rnd.Next(26));

Console.Write("| " + symbs[k] + " ");

}

Console.WriteLine("|");

Console.WriteLine("Массив в обратном порядке:");

for (k = symbs.Length -1; k >= 0; k--)

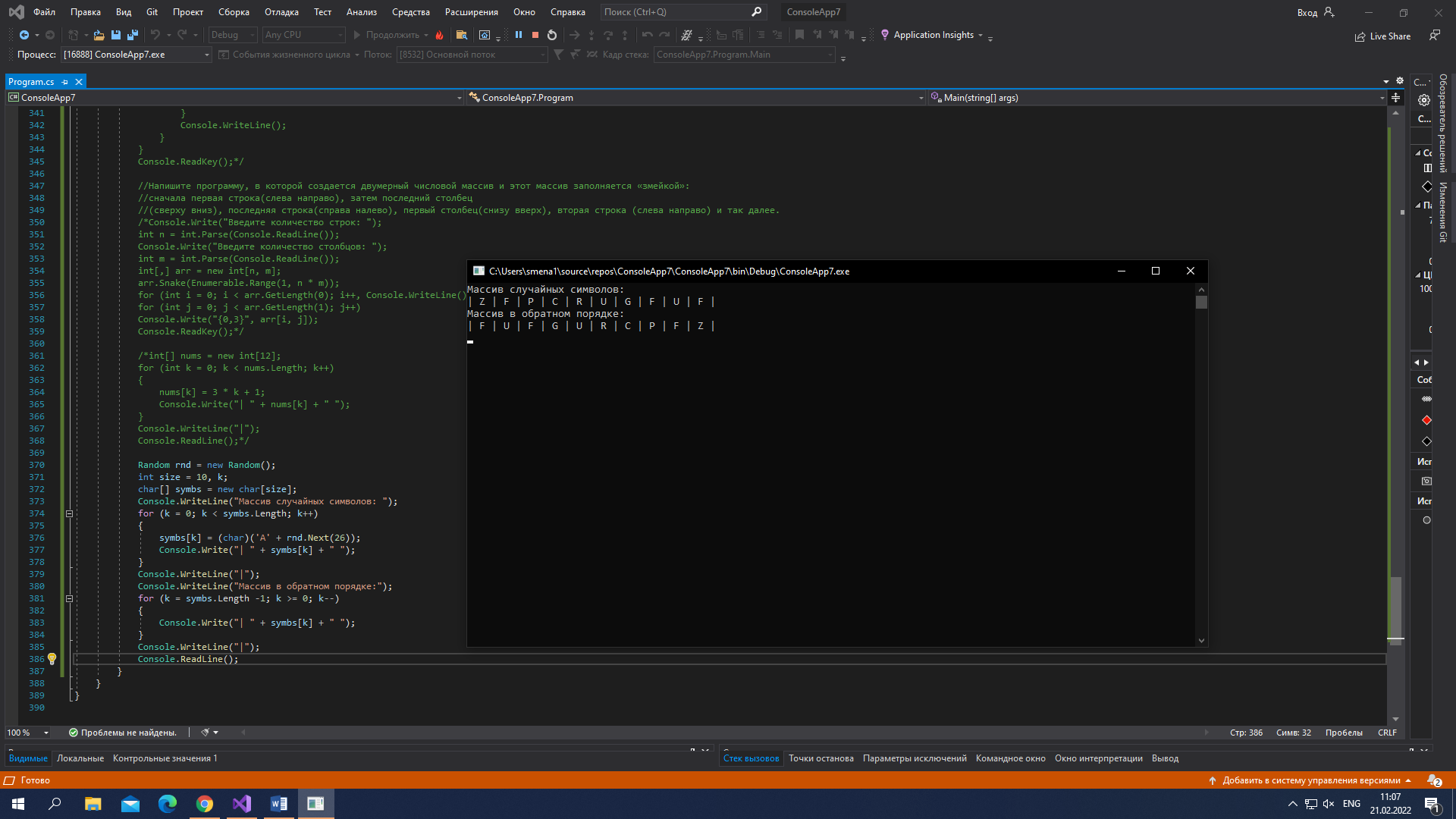
{

Console.Write("| " + symbs[k] + " ");

}

Console.WriteLine("|");

Console.ReadLine();



int[] nums = { 1, 3, 5, 7, 6, 5, 4 };

char[] symbs = new char[] { 'A', 'Z', 'B', 'Y' };

string[] txts = new string[3] { "один", "два", "три" };

Console.WriteLine("Массив nums:");

for (int k = 0; k < nums.Length; k++)

{

Console.Write(nums[k] + " ");

}

Console.WriteLine("\nМассив symbs:");

for (int k = 0; k < symbs.Length; k++)

{

Console.Write(symbs[k] + " ");

}

Console.WriteLine("\nМассив txts:");

for (int k = 0; k < txts.Length; k++)

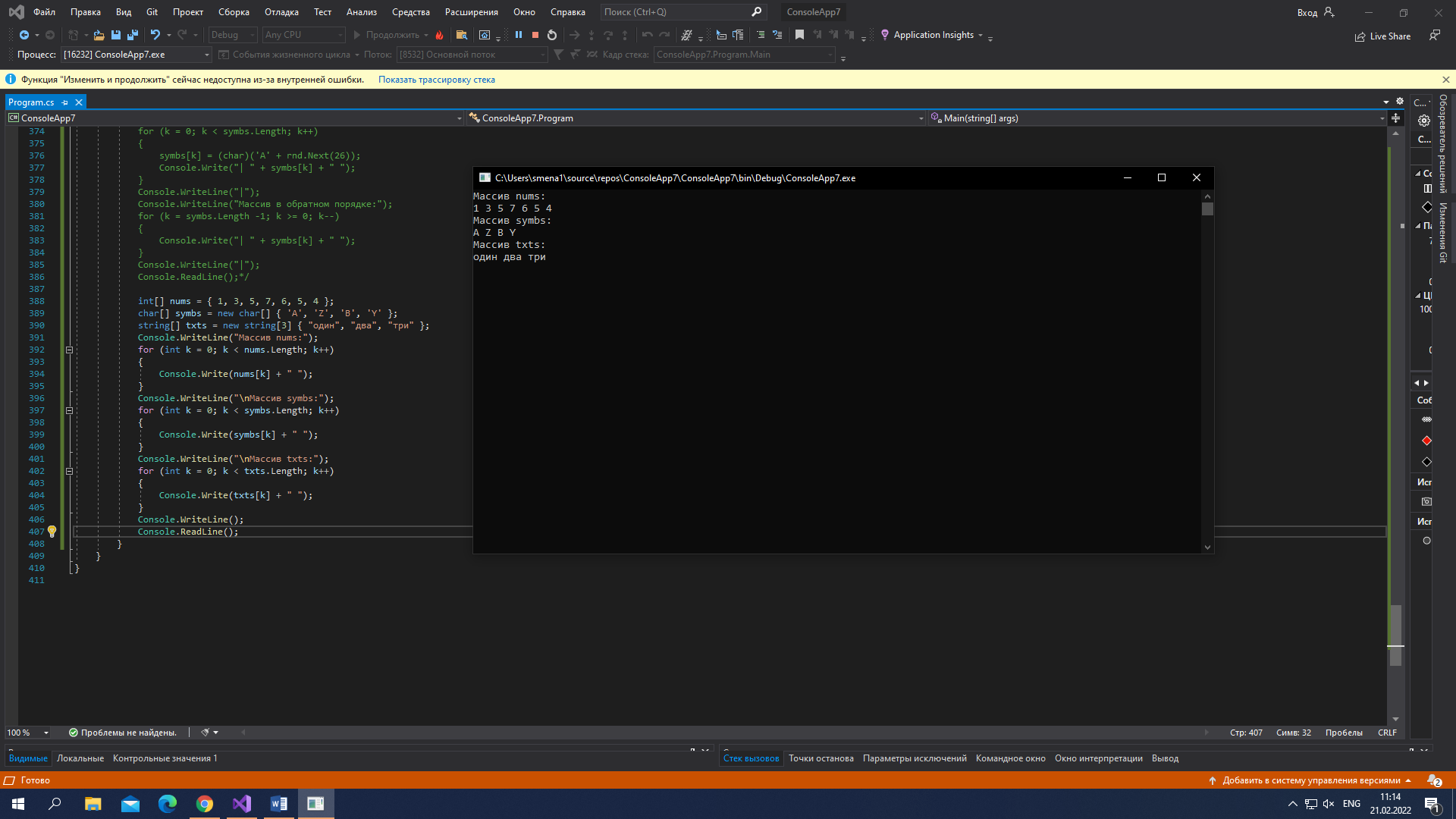
{

Console.Write(txts[k] + " ");

}

Console.WriteLine();

Console.ReadLine();



int[] A = { 1, 3, 5, 7, 9 };

int[] B, C;

B = A;

C = new int[A.Length];

for (int k = 0; k < A.Length; k++)

{

C[k] = A[k];

}

A[0] = 0;

B[B.Length - 1] = 0;

Console.WriteLine("A:\tB:\tC:");

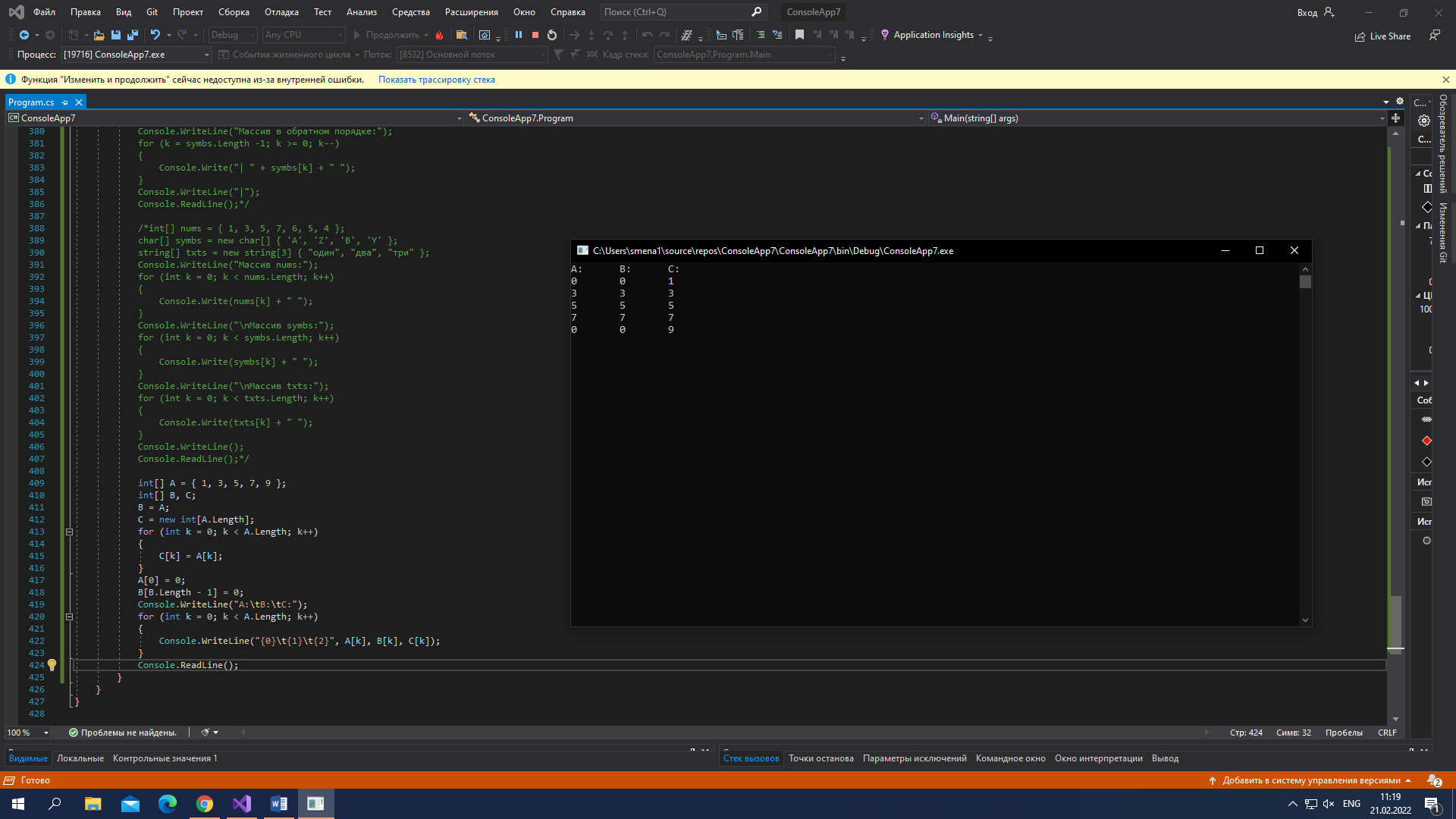
for (int k = 0; k < A.Length; k++)

{

Console.WriteLine("{0}\t{1}\t{2}", A[k], B[k], C[k]);

}

Console.ReadLine();



int value, index;

int size = 15;

Random rnd = new Random();

int[] nums = new int[size];

for (int k = 0; k < nums.Length; k++)

{

nums[k] = rnd.Next(1, 101);

Console.Write(nums[k] + " ");

}

Console.WriteLine();

index = 0;

value = nums[index];

for (int k = 1; k < nums.Length; k++)

{

if (nums[k] > value)

{

value = nums[k];

index = k;

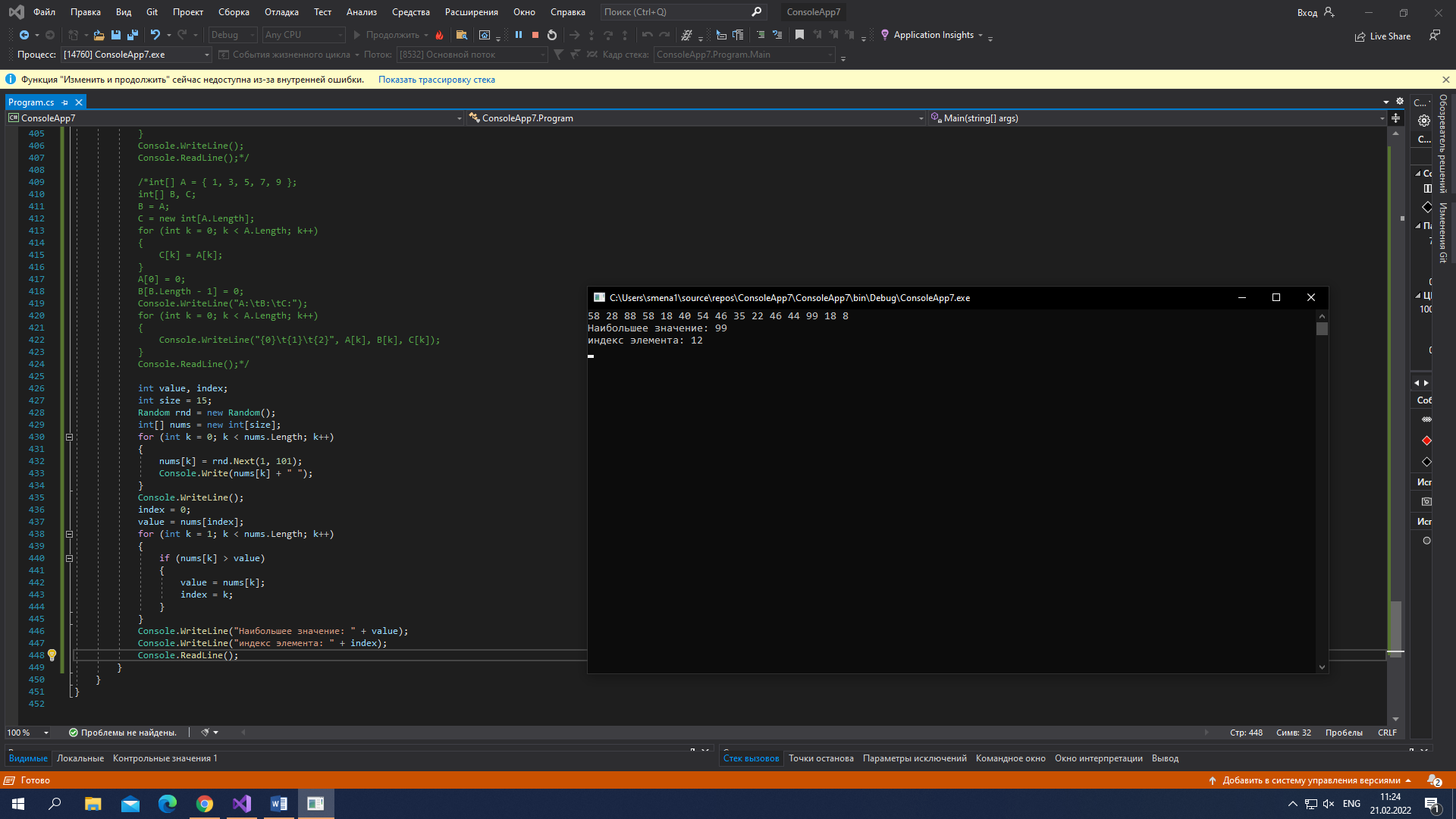
}

}

Console.WriteLine("Наибольшее значение: " + value);

Console.WriteLine("индекс элемента: " + index);

Console.ReadLine();



char s;

char[] symbs = { 'Q', 'Ы', 'a', 'B', 'R', 'A', 'r', 'q', 'b' };

Console.WriteLine("Массив до сортировки:");

for (int k = 0; k < symbs.Length; k++)

{

Console.Write(symbs[k] + " ");

}

Console.WriteLine();

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

{

for (int j = 0; j < symbs.Length - i; j++)

{

if (symbs[j] > symbs[j + 1])

{

s = symbs[j + 1];

symbs[j + 1] = symbs[j];

symbs[j] = s;

}

}

}

Console.WriteLine("Массив после сортировки:");

for (int k = 0; k < symbs.Length; k++)

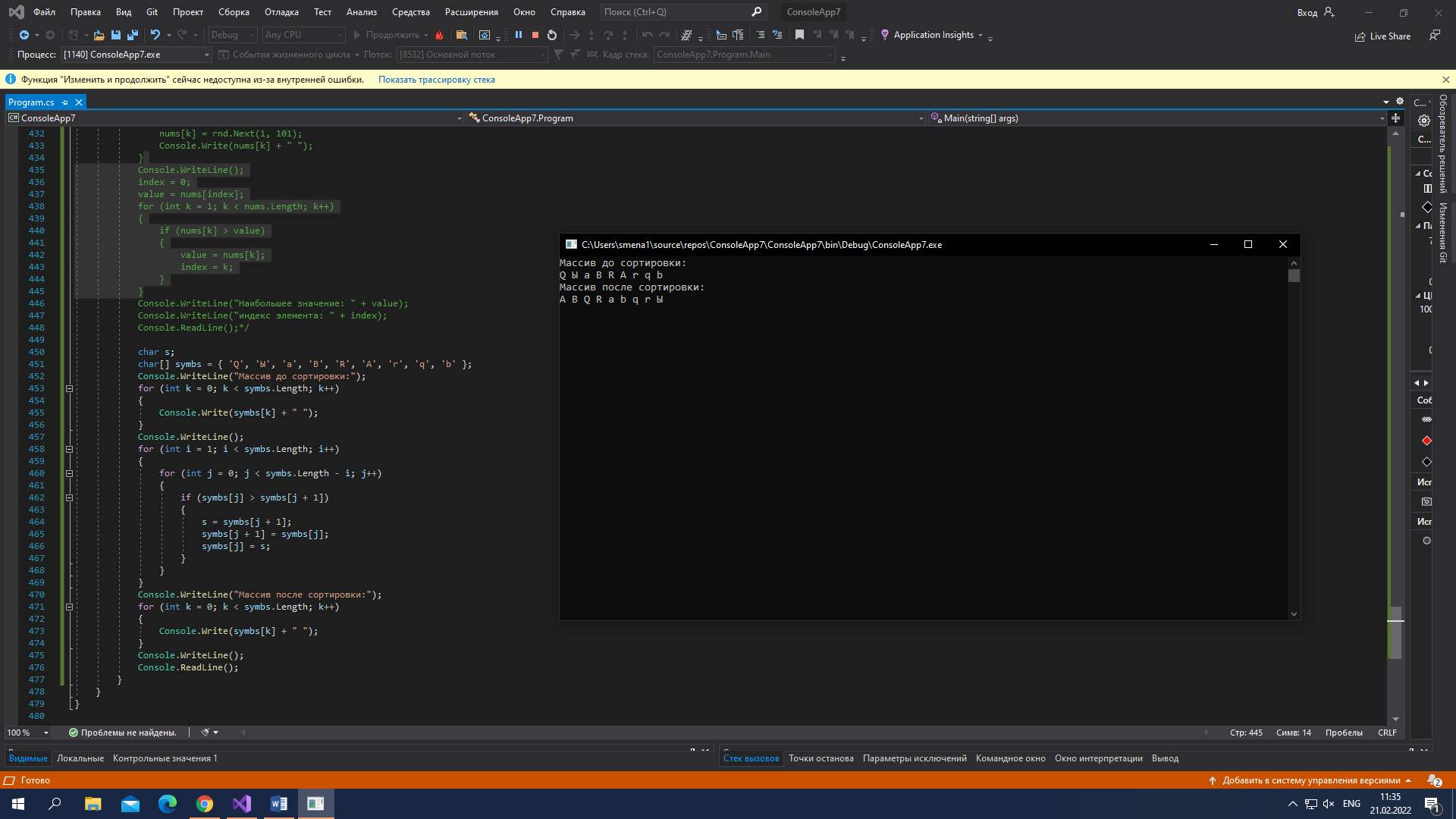
{

Console.Write(symbs[k] + " ");

}

Console.WriteLine();

Console.ReadLine();



int[] nums = { 1, 3, 4, 8, 9 };

char[] symbs = { 'a', 'b', 'A', 'B', 'Ы' };

string[] txts = { "красный", "жёлтый", "синий" };

Console.WriteLine("Целочисленный массив");

foreach (int s in nums)

{

Console.WriteLine("Число {0} - {1}", s, s % 2 == 0 ? "чётное" : "нечётное");

}

Console.WriteLine("Символьный массив");

foreach (char s in symbs)

{

Console.WriteLine("Код символа {0} - {1}", s, (int)s);

}

Console.WriteLine("Текстовый массив");

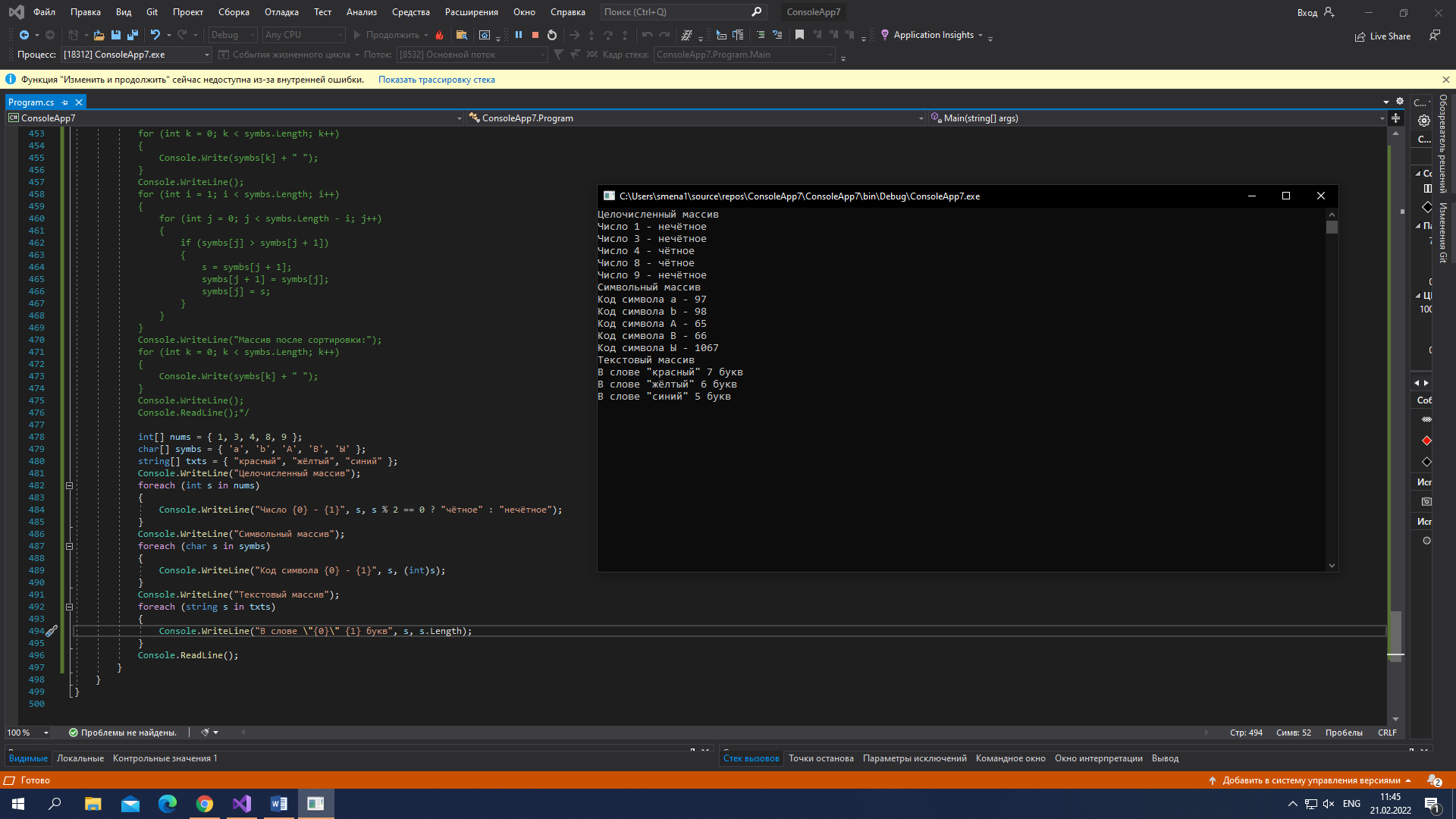
foreach (string s in txts)

{

Console.WriteLine("В слове \"{0}\" {1} букв", s, s.Length);

}

Console.ReadLine();



int rows = 3, cols = 5;

int[,] nums = new int[rows, cols];

int value = 1;

for (int i = 0; i < nums.GetLength(0); i++)

{

for (int j = 0; j < nums.GetLength(1); j++)

{

nums[i, j] = value;

value++;

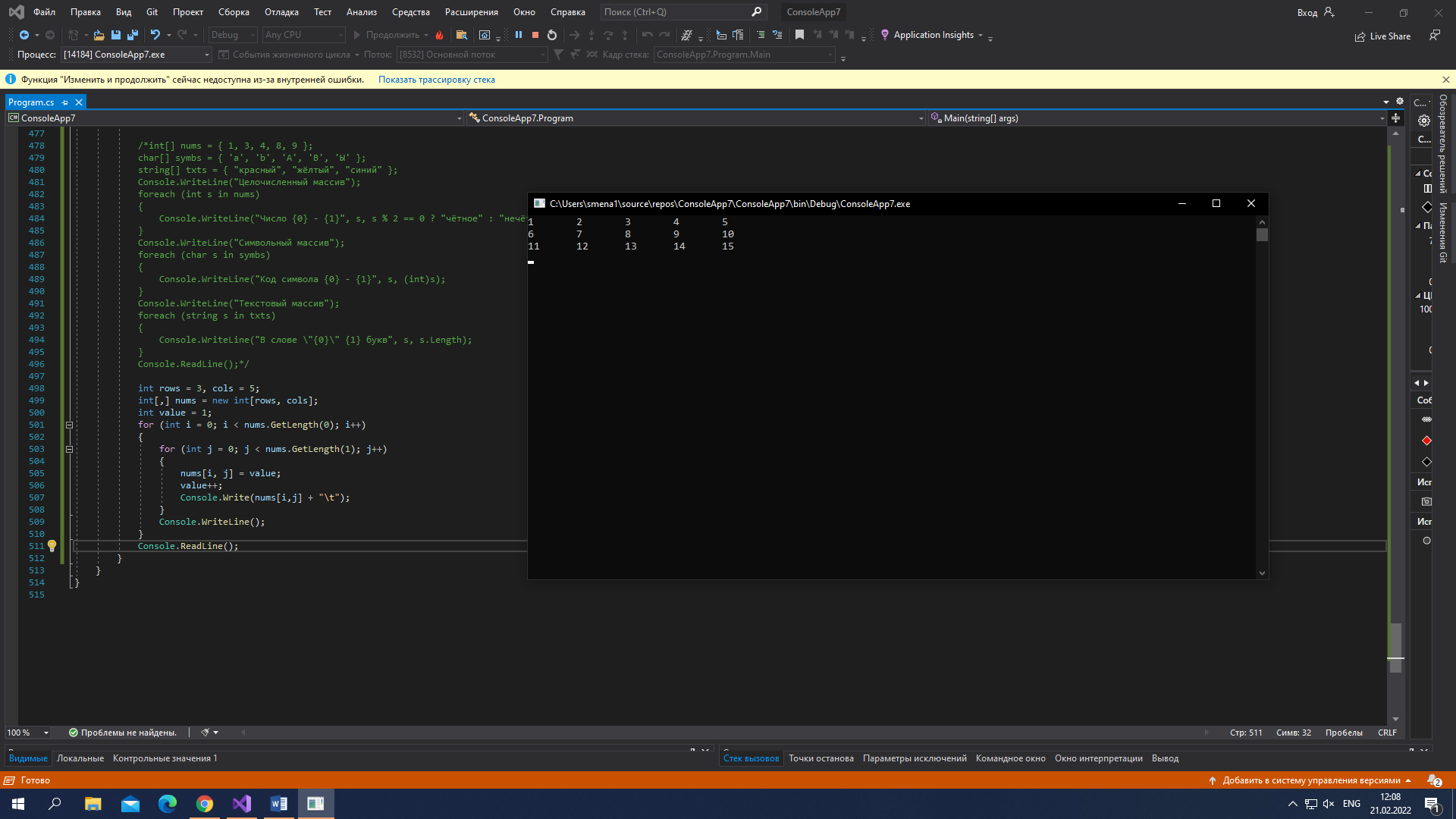
Console.Write(nums[i,j] + "\t");

}

Console.WriteLine();

}

Console.ReadLine();



char[,] symbs = { { 'A', 'B', 'C' }, { 'D', 'E', 'F' } };

Console.WriteLine("Исходный массив:");

for (int i = 0; i < symbs.GetLength(0); i++)

{

for (int j = 0; j < symbs.GetLength(0); j++)

{

Console.Write(symbs[i,j] + " ");

}

Console.WriteLine();

}

Random rnd = new Random();

int row = rnd.Next(symbs.GetLength(0) + 1);

int col = rnd.Next(symbs.GetLength(1) + 1);

Console.WriteLine("Добавляется {0}-я строка и {1}-й столбец", row, col);

char[,] tmp = new char[symbs.GetLength(0) + 1, symbs.GetLength(1) + 1];

int a, b;

char s = 'a';

for (int i = 0; i < symbs.GetLength(0); i++)

{

if (i < row) a = i;

else a = i + 1;

for (int j = 0; j < symbs.GetLength(1); j++)

{

if (j < col) b = j;

else b = j + 1;

tmp[a, b] = symbs[i, j];

}

}

for (int j = 0; j < tmp.GetLength(1); j++)

{

tmp[row, j] = s;

s++;

}

for (int i = 0; i < tmp.GetLength(0); i++)

{

if (i != row)

{

tmp[i, col] = s;

s++;

}

}

symbs = tmp;

Console.WriteLine("Новый массив:");

for (int i = 0; i < symbs.GetLength(0); i++)

{

for (int j = 0; j < symbs.GetLength(1); j++)

{

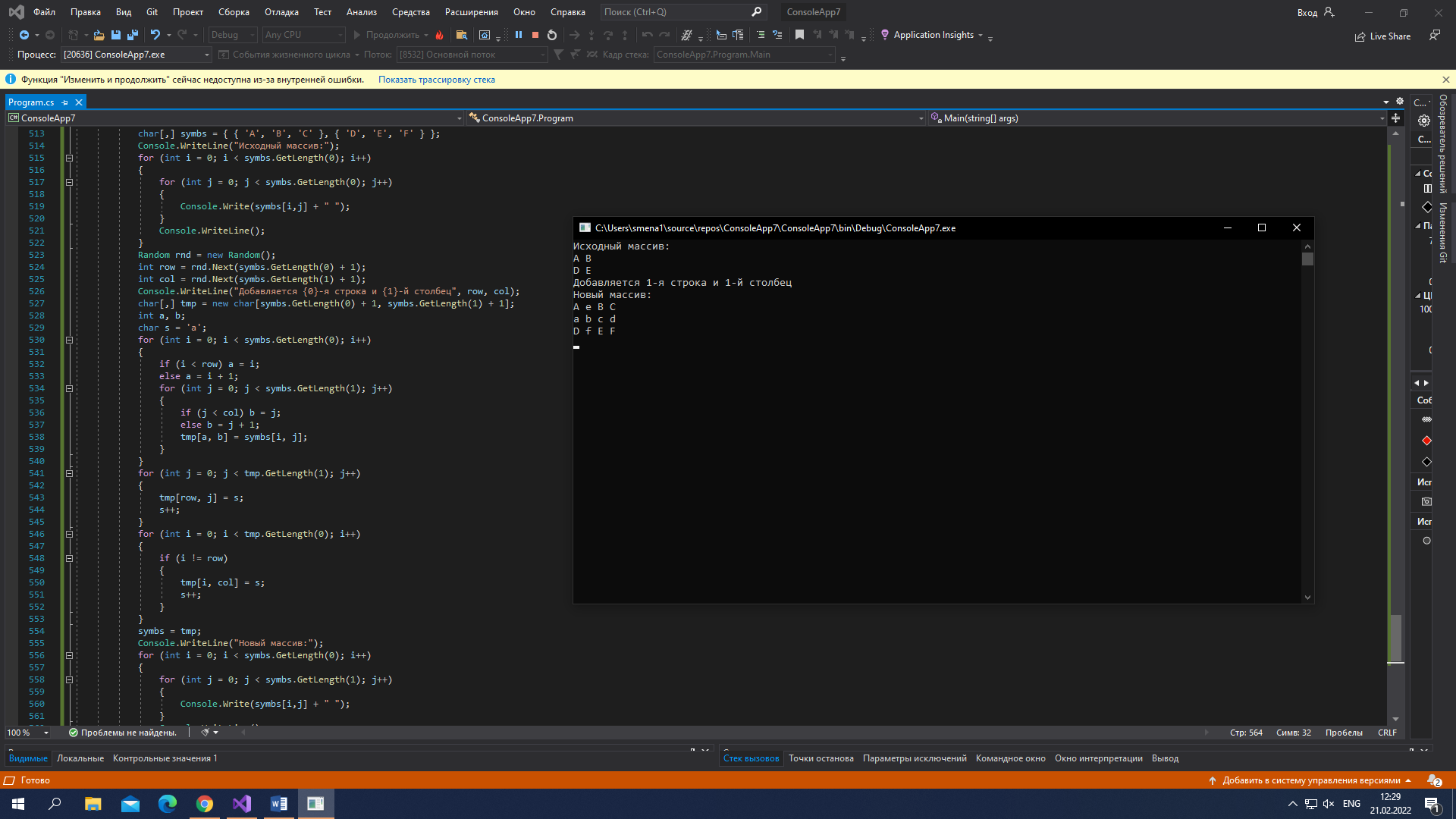
Console.Write(symbs[i,j] + " ");

}

Console.WriteLine();

}

Console.ReadLine();



char[,,] symbs = new char[4, 3, 5];

char[] s = { 'б', 'Б', 'b', 'B' };

for (int i = 0; i < symbs.GetLength(0); i++)

{

Console.WriteLine("Слой №{0}:", i + 1);

for (int j = 0; j < symbs.GetLength(1); j++)

{

for (int k = 0; k < symbs.GetLength(2); k++)

{

symbs[i, j, k] = s[i];

s[i]++;

Console.Write(symbs[i,j,k] + " ");

}

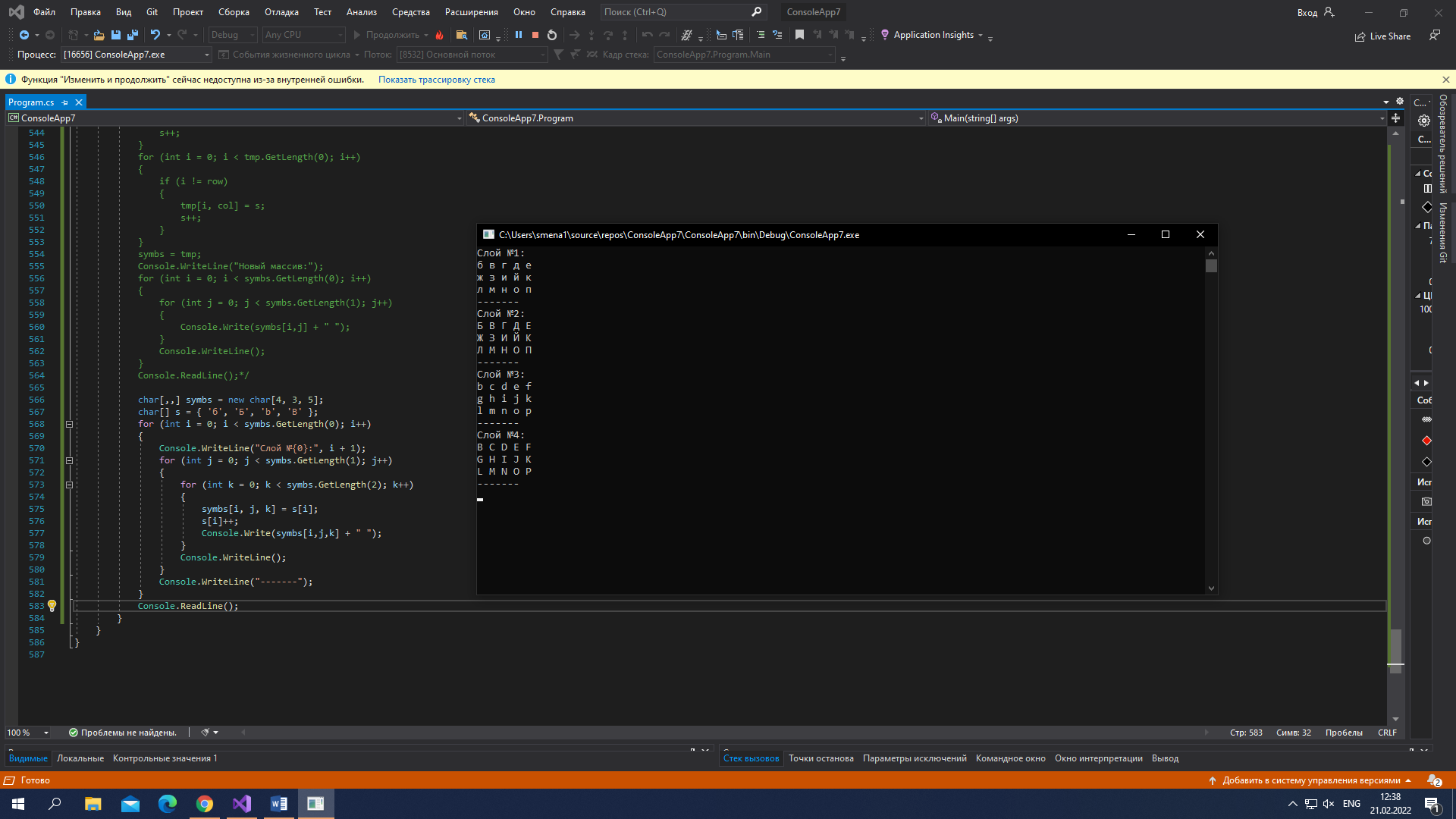
Console.WriteLine();

}

Console.WriteLine("-------");

}

Console.ReadLine();



char[][] symbs = new char[5][];

int[][] nums = { new int[] { 1, 2, 3 }, new int[] { 4, 5 }, new int[] { 6, 7, 8, 9 } };

char s = 'A';

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

{

symbs[i] = new char[i + 3];

for (int j = 0; j < symbs[i].Length; j++)

{

symbs[i][j] = s;

s++;

}

}

Console.WriteLine("Целочисленный массив:");

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

{

for (int j = 0; j < nums[i].Length; j++)

{

Console.Write("{0,3}", nums[i][j]);

}

Console.WriteLine();

}

Console.WriteLine("Символьный массив:");

foreach (char[] q in symbs)

{

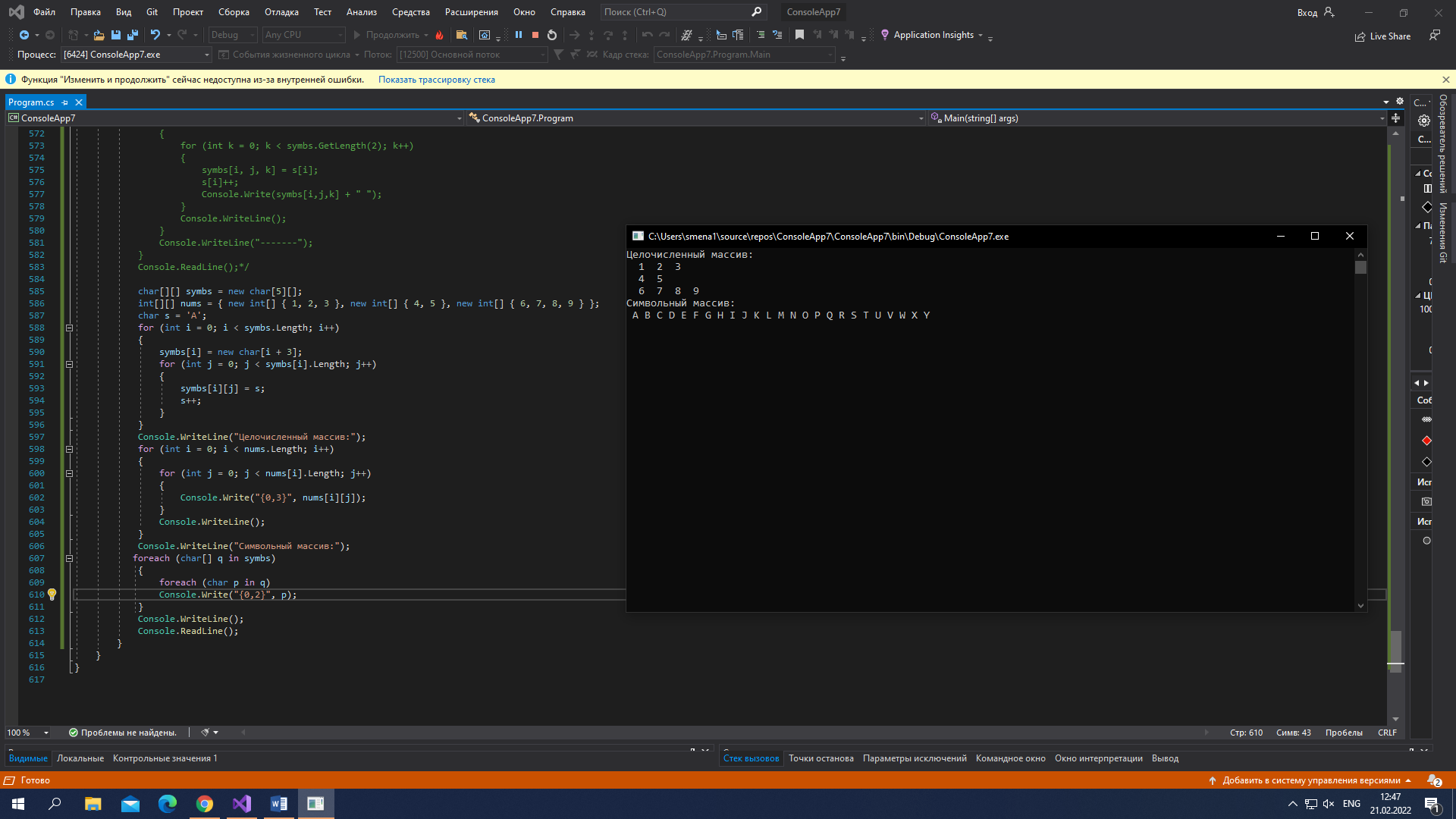
foreach (char p in q)

Console.Write("{0,2}", p);

}

Console.WriteLine();

Console.ReadLine();



Object[] objs = new Object[3];

objs[0] = 123;

objs[1] = 'A';

objs[2] = "Третий элемент";

Console.WriteLine("Создан такой массив: ");

for (int k = 0; k < objs.Length; k++)

{

Console.WriteLine(k + ": " + objs[k]);

}

objs[0] = (int)objs[0] + 111;

objs[1] = "Второй элемент";

objs[2] = 3.141592;

Console.WriteLine("После присваивания значений: ");

for (int k = 0; k < objs.Length; k++)

{

Console.WriteLine(k + ": " + objs[k]);

}

int[] nums = { 10, 20, 30 };

objs[2] = nums;

Console.WriteLine("Целочисленный массив: ");

for (int i = 0; i < ((int[])objs[2]).Length; i++)

{

Console.Write("{0, 3}", ((int[])objs[2])[i]);

}

Console.WriteLine();

((int[])objs[2])[1] = 0;

Console.WriteLine("Этот же массив: ");

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

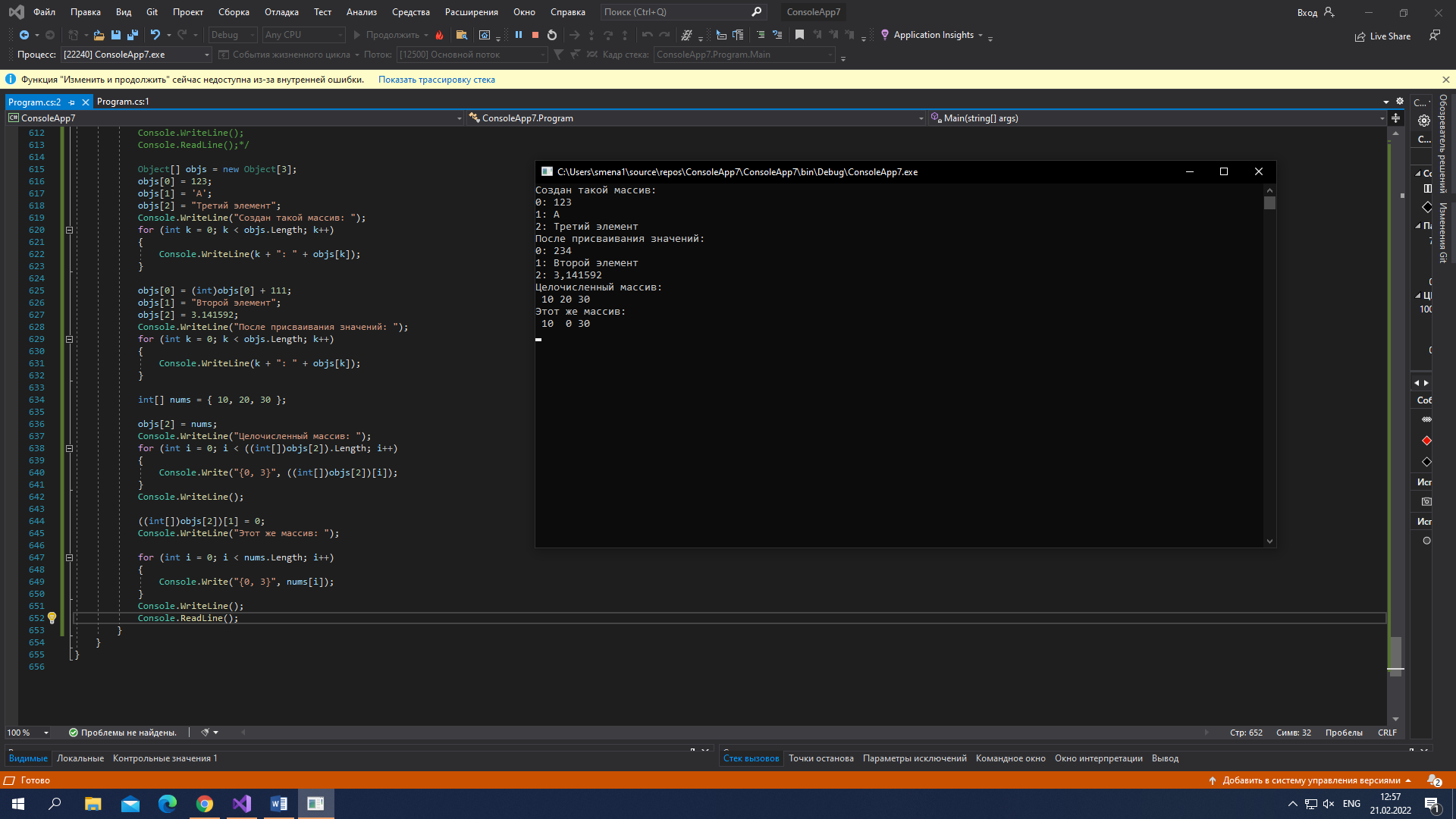
{

Console.Write("{0, 3}", nums[i]);

}

Console.WriteLine();

Console.ReadLine();



Задание 2.

//Напишите программу, в которой создается одномерный числовой массив и заполняется числами, которые при делении на 5 дают в остатке 2(числа 2, 7, 12, 17 и так далее).

//Размер массива вводится пользователем. Предусмотреть обработку ошибки, связанной с вводом некорректного значения.

int n, number = 0;

try

{

Console.WriteLine("Введите число: ");

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

int[] array = new int[n];

Console.WriteLine();

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

{

while (number % 5 != 2)

{

number++;

}

array[i] = number;

Console.Write(array[i] + " ");

number++;

}

}

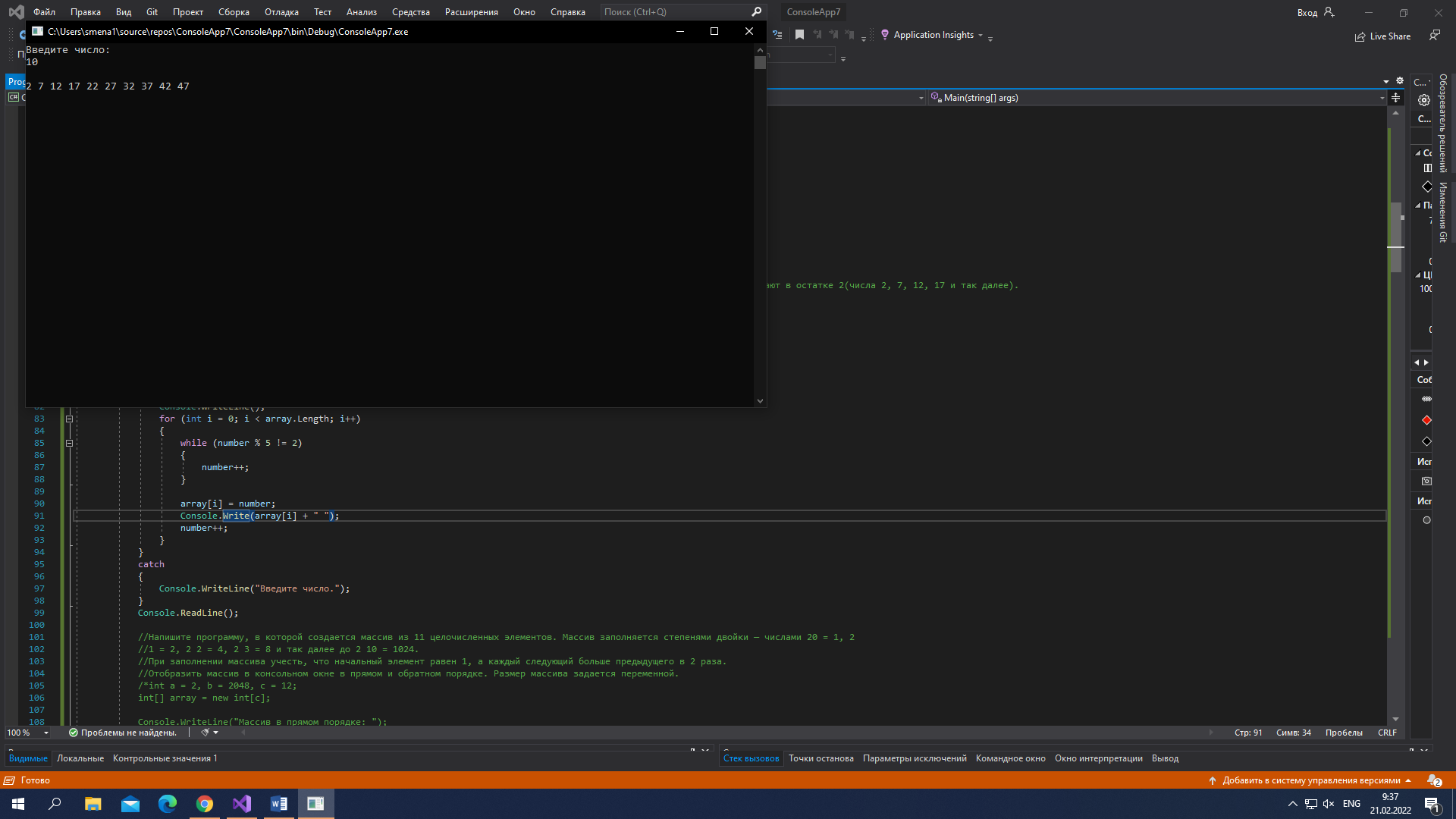
catch

{

Console.WriteLine("Введите число.");

}

Console.ReadLine();



Задание 3.

//Напишите программу, в которой создается массив из 11 целочисленных элементов. Массив заполняется степенями двойки — числами 20 = 1, 2

//1 = 2, 2 2 = 4, 2 3 = 8 и так далее до 2 10 = 1024.

//При заполнении массива учесть, что начальный элемент равен 1, а каждый следующий больше предыдущего в 2 раза.

//Отобразить массив в консольном окне в прямом и обратном порядке. Размер массива задается переменной.

int a = 2, b = 2048, c = 12;

int[] array = new int[c];

Console.WriteLine("Массив в прямом порядке: ");

for (int k = 11; k >= 0; k--)

{

b = b / 2;

array[k] = b;

Console.Write(array[k] + " ");

}

Console.WriteLine("\nМассив в обратном порядке: ");

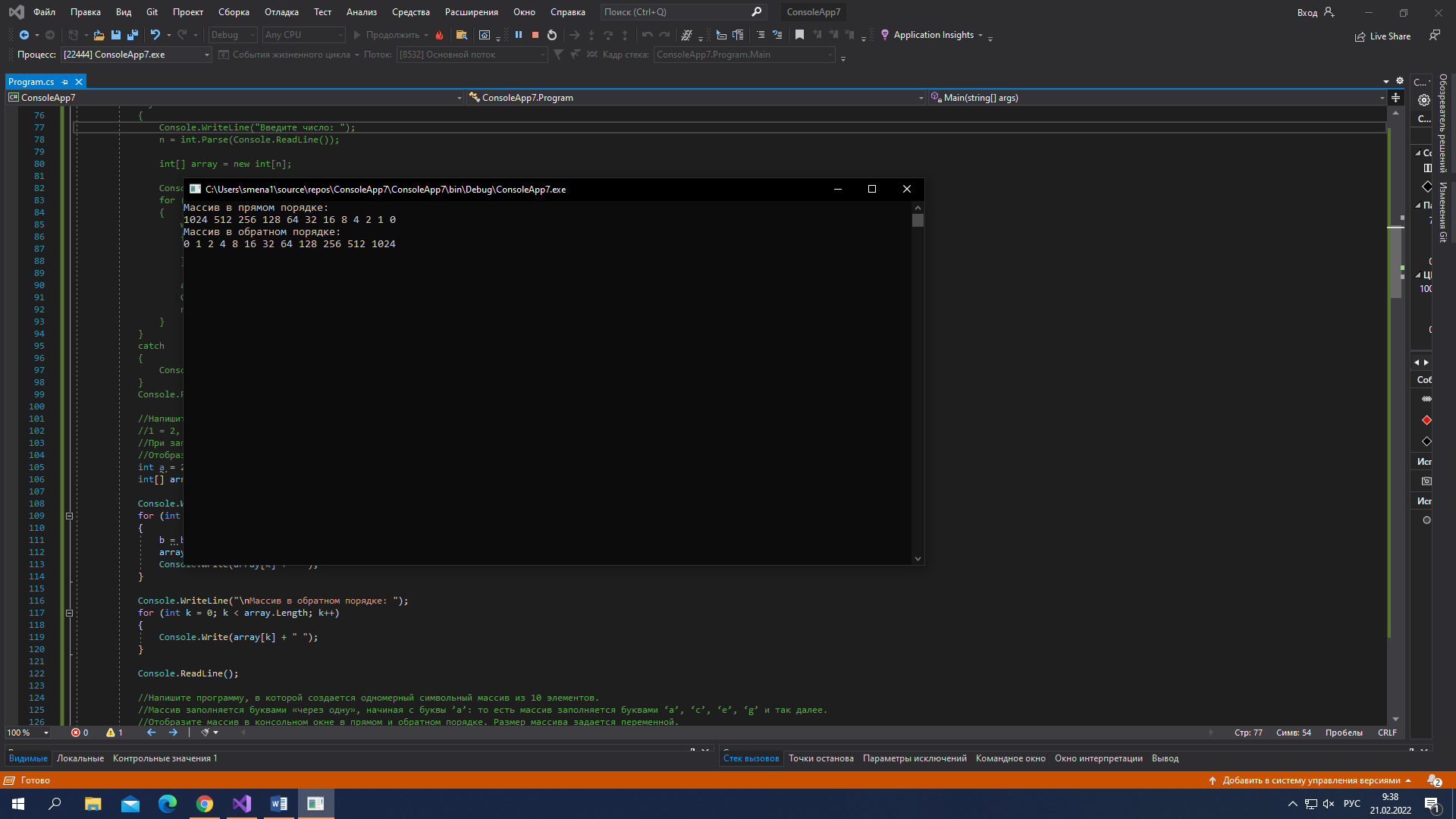
for (int k = 0; k < array.Length; k++)

{

Console.Write(array[k] + " ");

}

Console.ReadLine();



Задание 4.

//Напишите программу, в которой создается одномерный символьный массив из 10 элементов.

//Массив заполняется буквами «через одну», начиная с буквы ’а’: то есть массив заполняется буквами ‘а’, ‘с’, ‘е’, ‘g’ и так далее.

//Отобразите массив в консольном окне в прямом и обратном порядке. Размер массива задается переменной.

char a = 'a';

int n = 10;

char[] array = new char[n];

Console.WriteLine("Массив в прямом порядке: ");

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

{

array[i] = a;

a = (char)(int)(a + 2);

Console.Write(array[i] + " ");

}

Console.WriteLine("\nМассив в обратном порядке: ");

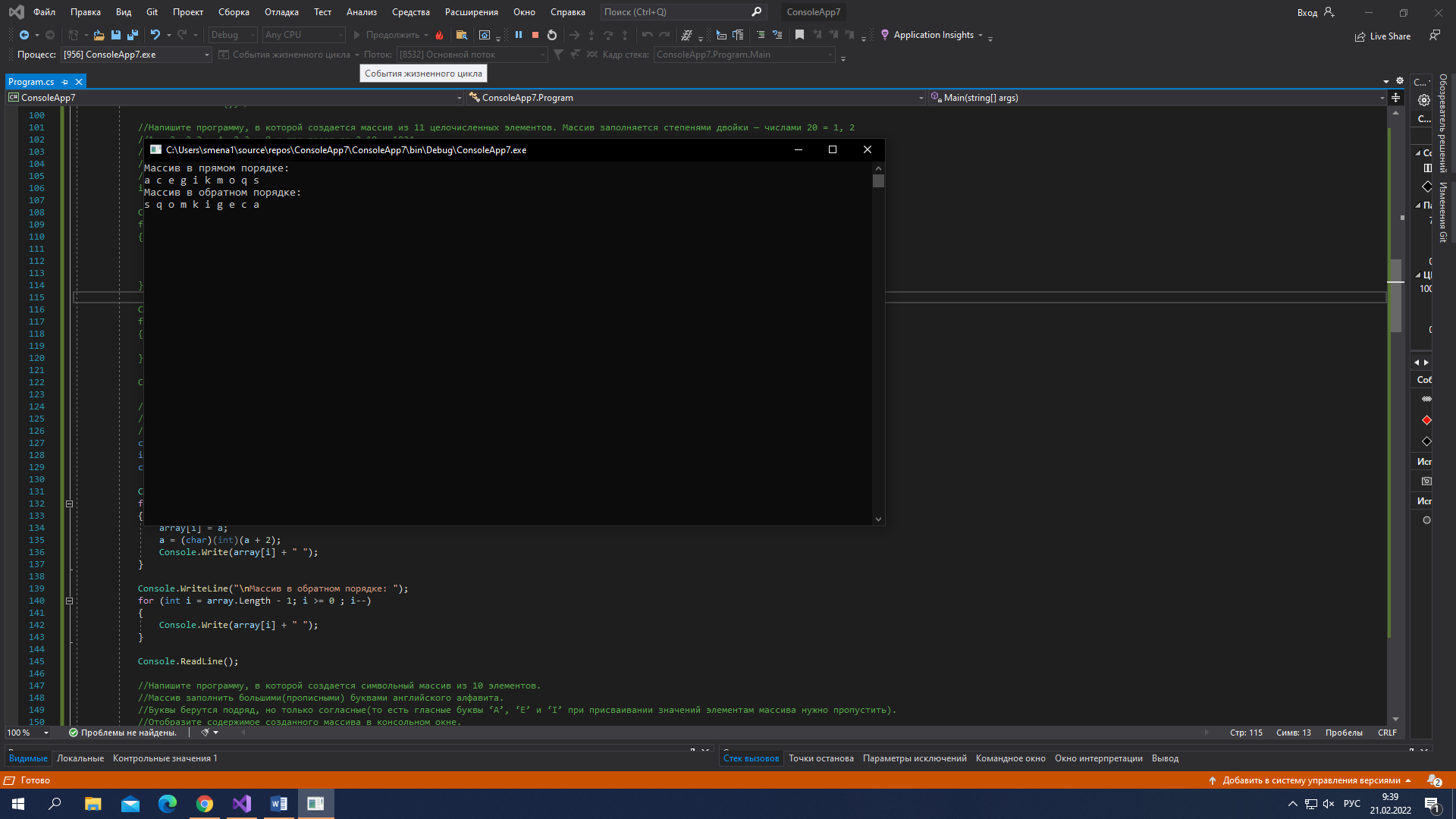
for (int i = array.Length - 1; i >= 0 ; i--)

{

Console.Write(array[i] + " ");

}

Console.ReadLine();



Задание 5.

//Напишите программу, в которой создается символьный массив из 10 элементов.

//Массив заполнить большими(прописными) буквами английского алфавита.

//Буквы берутся подряд, но только согласные(то есть гласные буквы ‘А’, ‘Е’ и ‘I’ при присваивании значений элементам массива нужно пропустить).

//Отобразите содержимое созданного массива в консольном окне.

char a = 'A';

char[] array = new char[10];

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

{

switch (a)

{

case 'A':

a++;

break;

case 'E':

a++;

break;

case 'Y':

a++;

break;

case 'U':

a++;

break;

case 'O':

a++;

break;

case 'I':

a++;

break;

}

array[i] = a;

a++;

}

Console.WriteLine("Массив: ");

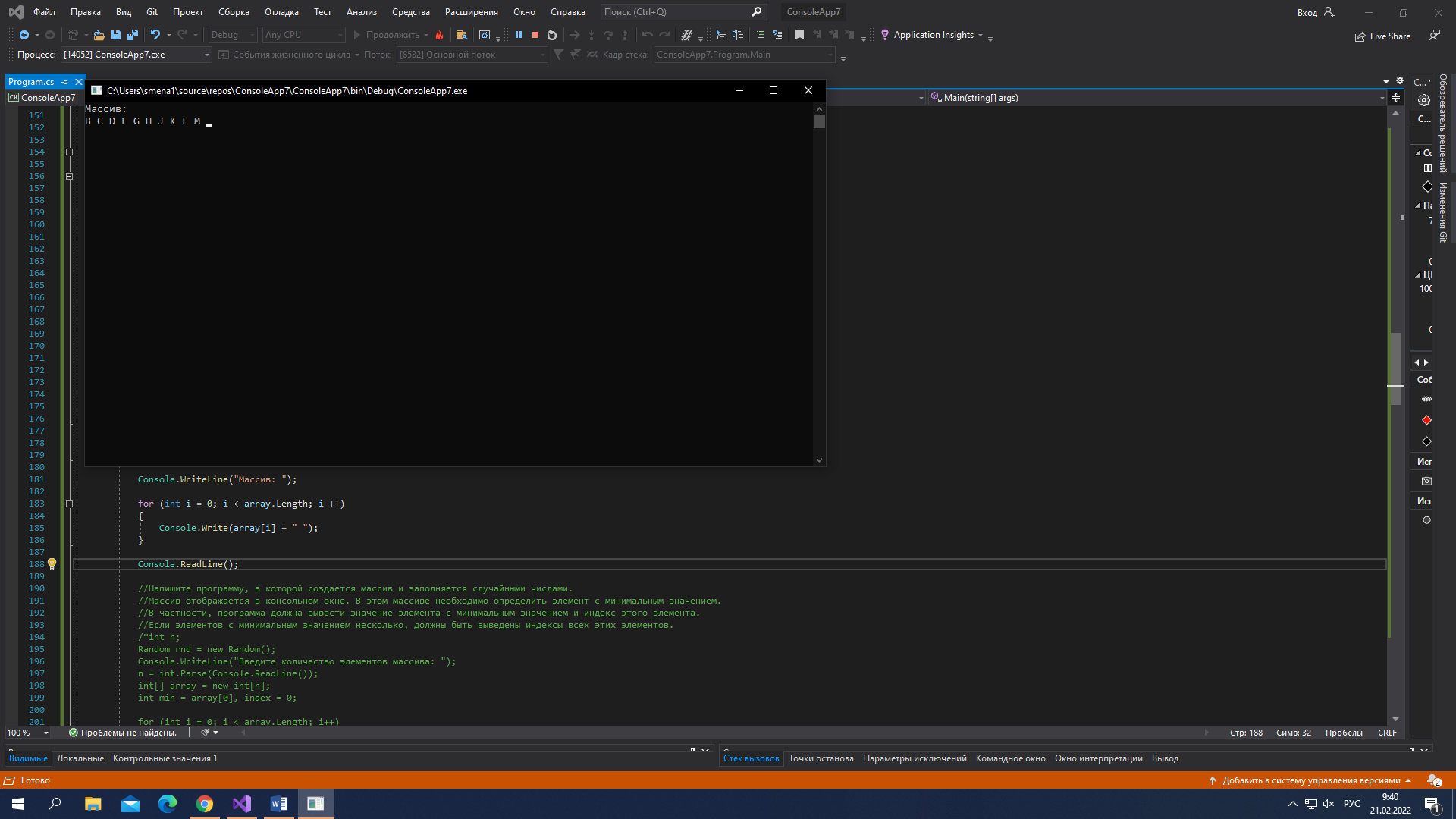
for (int i = 0; i < array.Length; i ++)

{

Console.Write(array[i] + " ");

}

Console.ReadLine();



Задание 6.

//Напишите программу, в которой создается массив и заполняется случайными числами.

//Массив отображается в консольном окне. В этом массиве необходимо определить элемент с минимальным значением.

//В частности, программа должна вывести значение элемента с минимальным значением и индекс этого элемента.

//Если элементов с минимальным значением несколько, должны быть выведены индексы всех этих элементов.

int n;

Random rnd = new Random();

Console.WriteLine("Введите количество элементов массива: ");

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

int[] array = new int[n];

int min = array[0], index = 0;

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

{

array[i] = rnd.Next(-100, 100);

Console.Write(array[i] + " ");

if (min > array[i])

{

min = array[i];

index = i;

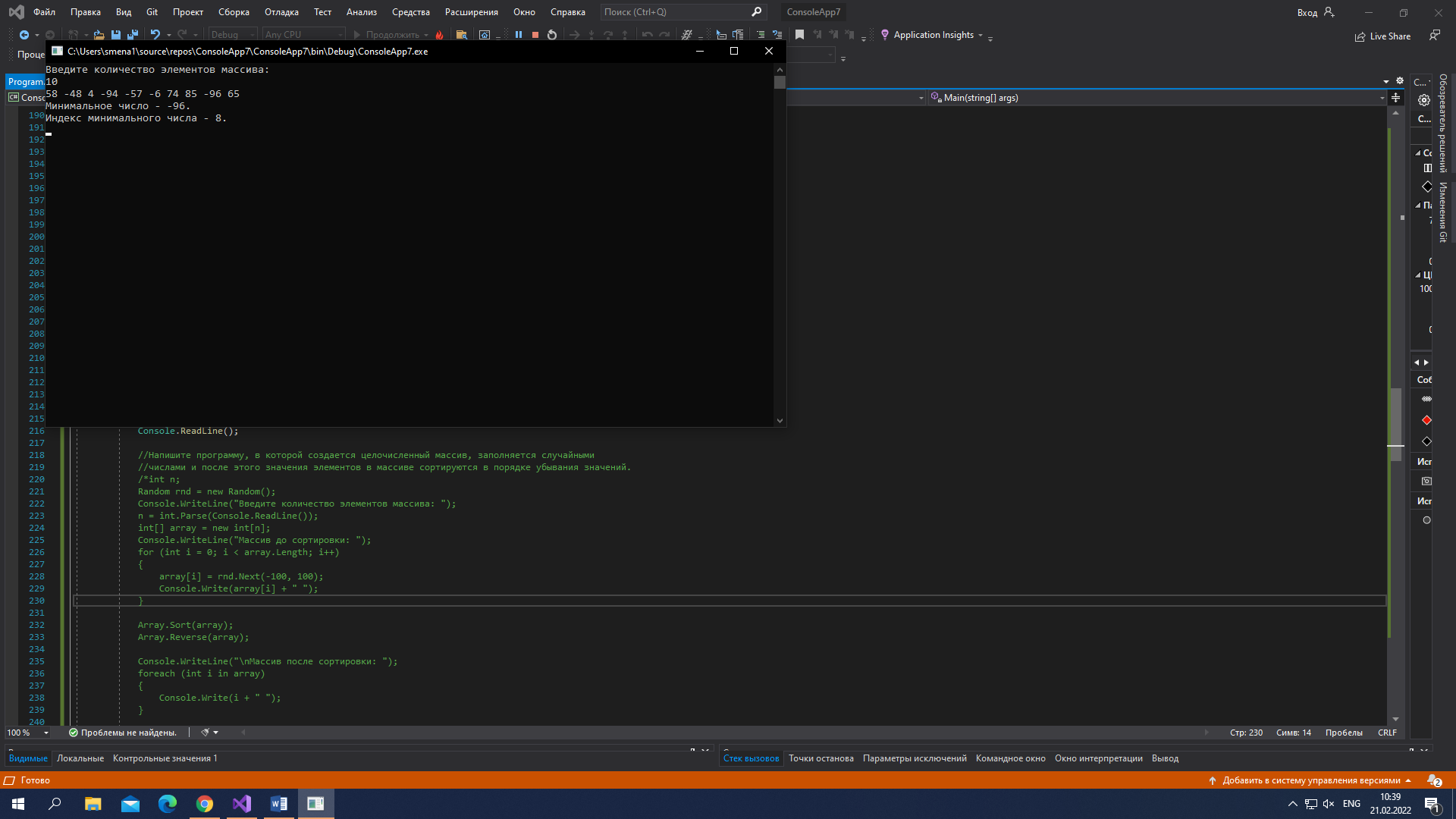
}

}

Console.WriteLine($"Минимальное число - {min}.");

Console.WriteLine($"Индекс минимального числа - {index}.");

Console.ReadLine();



Задание 7.

//Напишите программу, в которой создается целочисленный массив, заполняется случайными

//числами и после этого значения элементов в массиве сортируются в порядке убывания значений.

int n;

Random rnd = new Random();

Console.WriteLine("Введите количество элементов массива: ");

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

int[] array = new int[n];

Console.WriteLine("Массив до сортировки: ");

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

{

array[i] = rnd.Next(-100, 100);

Console.Write(array[i] + " ");

}

Array.Sort(array);

Array.Reverse(array);

Console.WriteLine("\nМассив после сортировки: ");

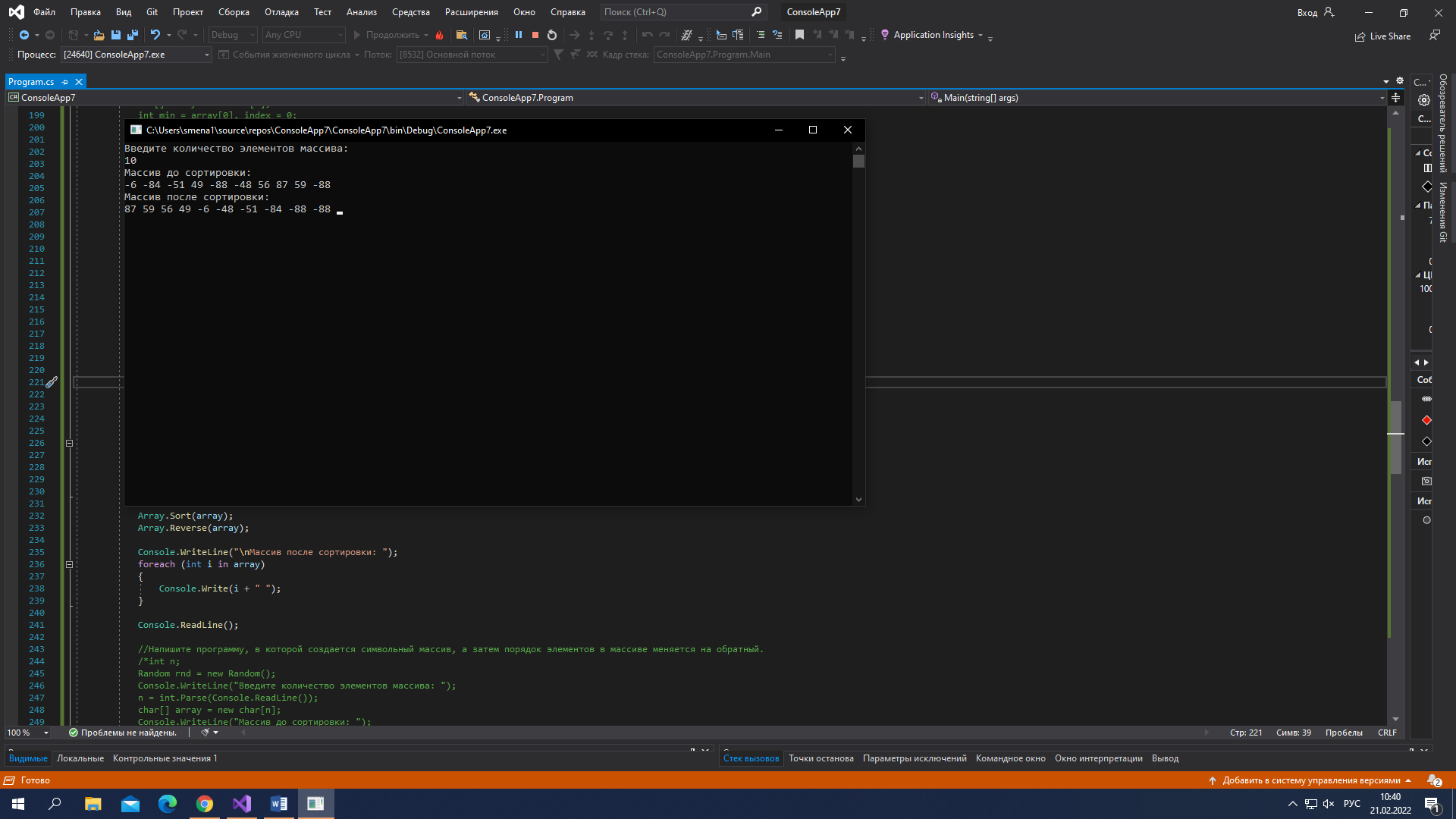
foreach (int i in array)

{

Console.Write(i + " ");

}

Console.ReadLine();



Задание 8.

//Напишите программу, в которой создается символьный массив, а затем порядок элементов в массиве меняется на обратный.

int n;

Random rnd = new Random();

Console.WriteLine("Введите количество элементов массива: ");

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

char[] array = new char[n];

Console.WriteLine("Массив до сортировки: ");

for (int i = 0; i < n; i++)

{

array[i] = (char)rnd.Next(0x0410, 0x44F);

Console.Write(array[i] + " ");

}

Array.Sort(array);

Array.Reverse(array);

Console.WriteLine("\nМассив после сортировки: ");

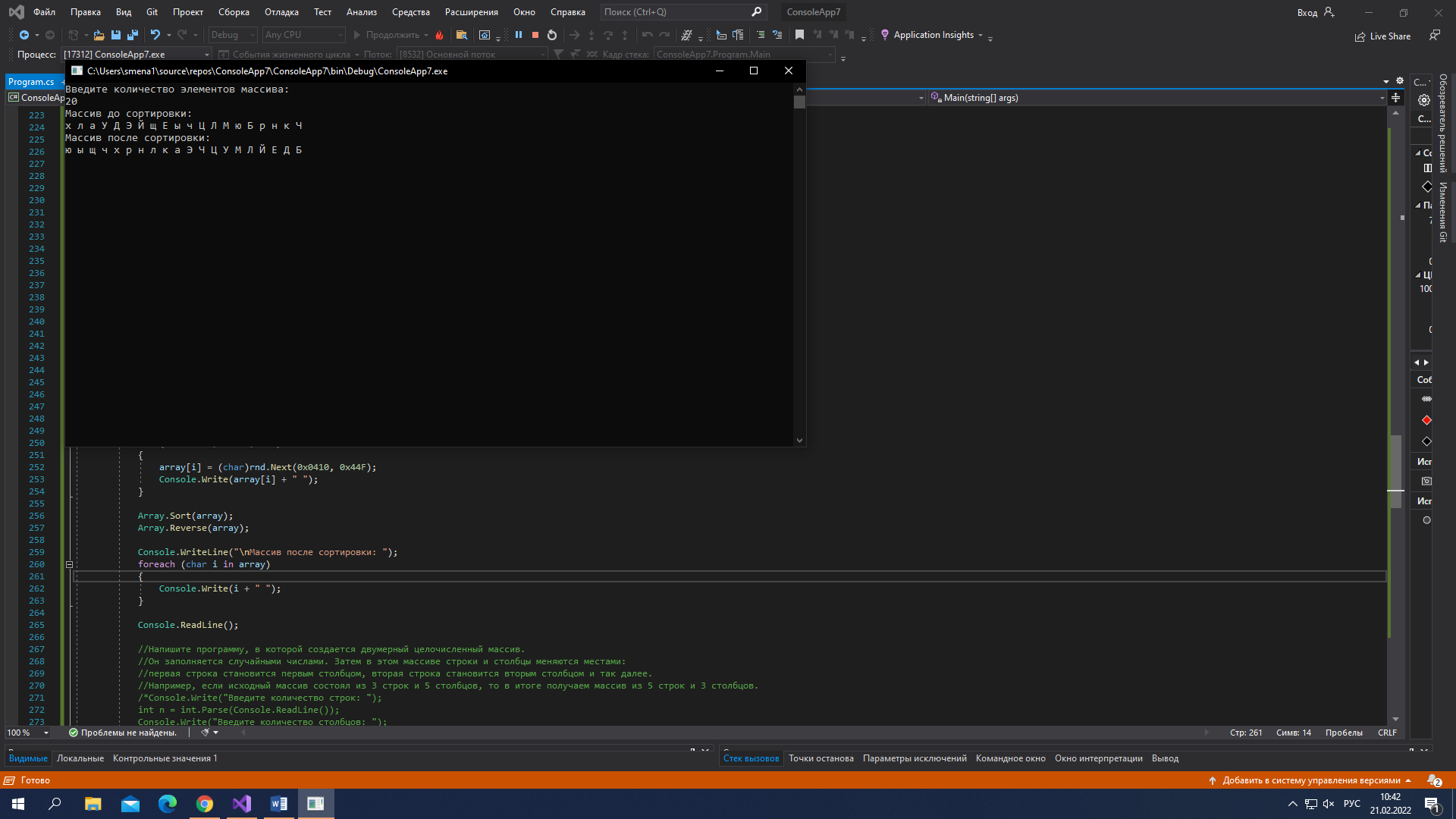
foreach (char i in array)

{

Console.Write(i + " ");

}

Console.ReadLine();



Задание 9.

//Напишите программу, в которой создается двумерный целочисленный массив.

//Он заполняется случайными числами. Затем в этом массиве строки и столбцы меняются местами:

//первая строка становится первым столбцом, вторая строка становится вторым столбцом и так далее.

//Например, если исходный массив состоял из 3 строк и 5 столбцов, то в итоге получаем массив из 5 строк и 3 столбцов.

public static T[,] Transpose<T>(T[,] matrix)

{

var transposeMatrix = new T[matrix.GetLength(1), matrix.GetLength(0)];

for (int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < matrix.GetLength(1); j++)

{

transposeMatrix[j, i] = matrix[i, j];

}

}

return transposeMatrix;

}

Console.Write("Введите количество строк: ");

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

Console.Write("Введите количество столбцов: ");

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

int[,] array = new int[n, m];

Random rnd = new Random();

Console.Write("\nМассив: \n");

for (int i = 0; i < n; i++)

{

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

{

array[i, j] = rnd.Next(-100, 100);

Console.Write(array[i, j] + " ");

}

Console.WriteLine();

}

Console.WriteLine();

Console.Write("\nМассив после транспонирования: \n");

var array\_ = Transpose(array);

for (int i = 0; i < n; i++)

{

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

{

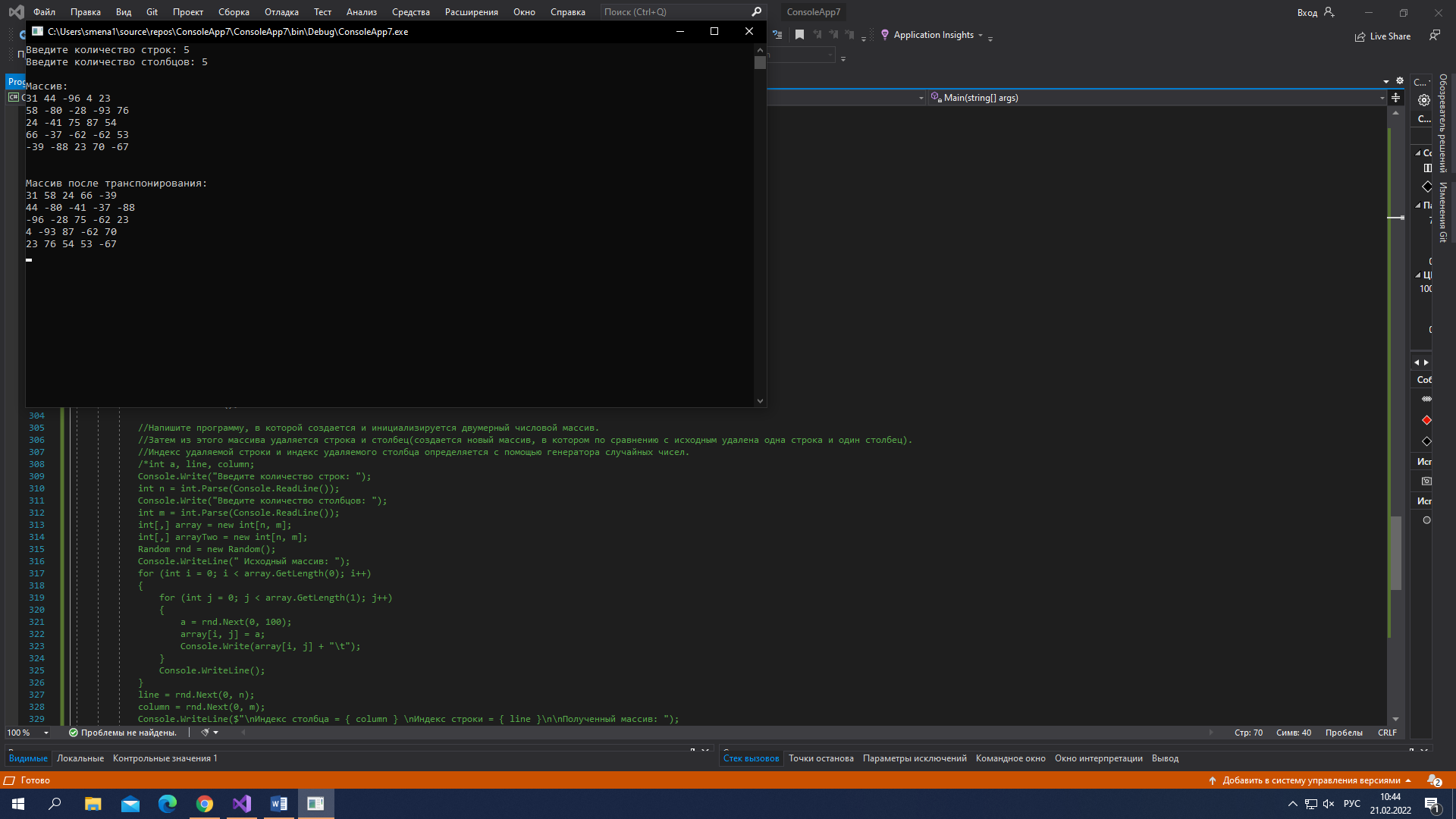
Console.Write(array\_[i, j] + " ");

}

Console.WriteLine();

}

Console.ReadLine();



Задание 10.

//Напишите программу, в которой создается и инициализируется двумерный числовой массив.

//Затем из этого массива удаляется строка и столбец (создается новый массив, в котором по сравнению с исходным удалена одна строка и один столбец).

//Индекс удаляемой строки и индекс удаляемого столбца определяется с помощью генератора случайных чисел.

int a, line, column;

Console.Write("Введите количество строк: ");

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

Console.Write("Введите количество столбцов: ");

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

int[,] array = new int[n, m];

int[,] arrayTwo = new int[n, m];

Random rnd = new Random();

Console.WriteLine(" Исходный массив: ");

for (int i = 0; i < array.GetLength(0); i++)

{

for (int j = 0; j < array.GetLength(1); j++)

{

a = rnd.Next(0, 100);

array[i, j] = a;

Console.Write(array[i, j] + "\t");

}

Console.WriteLine();

}

line = rnd.Next(0, n);

column = rnd.Next(0, m);

Console.WriteLine($"\nИндекс столбца = {column} \nИндекс строки = {line}\n\nПолученный массив: ");

for (int i = 0; i < arrayTwo.GetLength(0); i++)

{

if (i != line)

{

for (int j = 0; j < arrayTwo.GetLength(1); j++)

{

if (j != column)

{

arrayTwo[i, j] = array[i, j];

Console.Write(arrayTwo[i, j] + "\t");

}

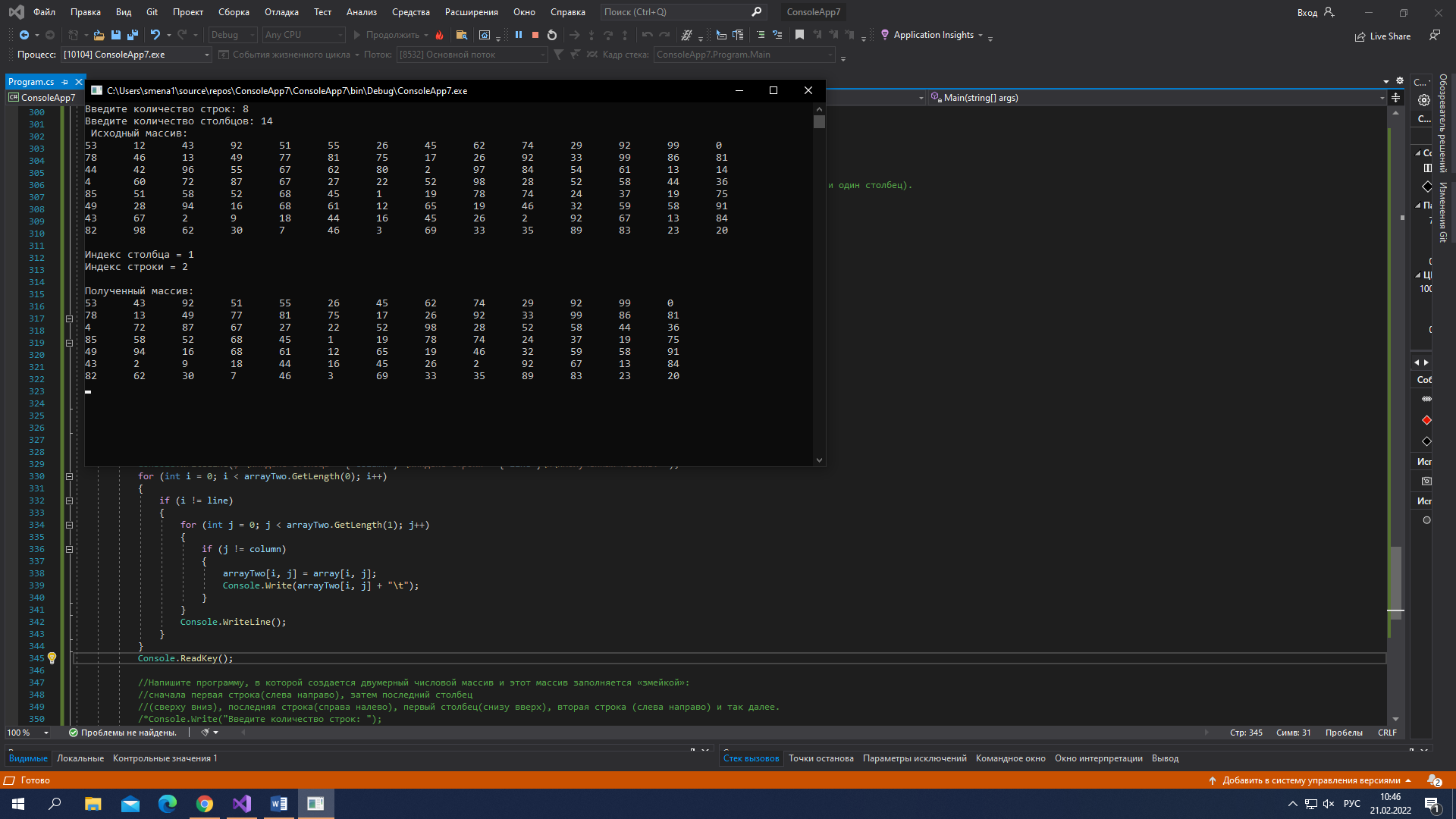
}

Console.WriteLine();

}

}

Console.ReadKey();



Задание 11.

//Напишите программу, в которой создается двумерный числовой массив и этот массив заполняется «змейкой»:

//сначала первая строка (слева направо), затем последний столбец

//(сверху вниз), последняя строка (справа налево), первый столбец (снизу вверх), вторая строка (слева направо) и так далее.

static class Extensions

{

public static void Snake<T>(this T[,] arr, IEnumerable<T> sourceValues)

{

int m = arr.GetLength(0);

int n = arr.GetLength(1);

int maxCount = m \* n;

int mVector = 0;

int nVector = 1;

int mi = 0, ni = 0;

foreach (var val in sourceValues.Take(maxCount))

{

arr[mi, ni] = val;

mi += mVector;

ni += nVector;

if (nVector != 0)

{

if (nVector < 0 && ni < m - mi)

{

mVector = -1;

nVector = 0;

}

if (nVector > 0 && ni >= n - mi - 1)

{

mVector = 1;

nVector = 0;

}

}

else if (mVector != 0)

{

if (mVector < 0 && mi <= ni + 1)

{

mVector = 0;

nVector = 1;

}

if (mVector > 0 && mi > m - (n - ni) - 1)

{

mVector = 0;

nVector = -1;

}

}

}

}

}

Console.Write("Введите количество строк: ");

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

Console.Write("Введите количество столбцов: ");

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

int[,] arr = new int[n, m];

arr.Snake(Enumerable.Range(1, n \* m));

for (int i = 0; i < arr.GetLength(0); i++, Console.WriteLine())

for (int j = 0; j < arr.GetLength(1); j++)

Console.Write("{0,3}", arr[i, j]);

Console.ReadKey();

