Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

**Лабораторная работа №9**

**По дисциплине ОАиП за 2 семестр**  
**Тема: “** **Пользовательские функции”**

**Выполнил:**

Студент группы ПО-6(1)  
 1-го курса

Мартынович Даниил

**Проверила:**

Хацкевич М.В.

Брест 2021

**Цель работы:** изучить основные принципы написания пользовательских функций, ознакомиться с возможностями передачи данных в функции и получения результата по итогам работы функции. Реализовать собственные функции для обработки данных составных и простых типов.

**Ход работы**

**Задача 1 «массивы»**

**Вариант 18**

Описать процедуру Chessboard(M, N, A), формирующую по целым положительным числам M и N матрицу A размера M × N, которая содержит числа 0 и 1, расположенные в «шахматном» порядке, причем A1,1 = 0. Двумерный целочисленный массив A является выходным параметром. С помощью этой процедуры по данным целым числам M и N сформировать матрицу A размера M × N.

#include <iostream>

using namespace std;

void Chessboard( int M, int N, int \*\*A);////Объявление функции

void OutPut\_Arr(int M, int N, int \*\*A);

int main(int M, int N, int \*\*arr)

{

setlocale(LC\_ALL, "RU");

cout << "Введите количество строк = ";

cin >> M;

cout << "Введите количество колонок = ";

cin >> N;

Chessboard(M, N, arr);

OutPut\_Arr(M, N, arr);

system("pause");

return 0;

}

void Chessboard(int M, int N, int\*\* A)

{

/////Заполнение массива

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

{

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

{

A[i][j] = (i + j) % 2;

}

}

}

void OutPut\_Arr(int M, int N, int\*\* A)

{ //////Вывод массива

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

{

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

{

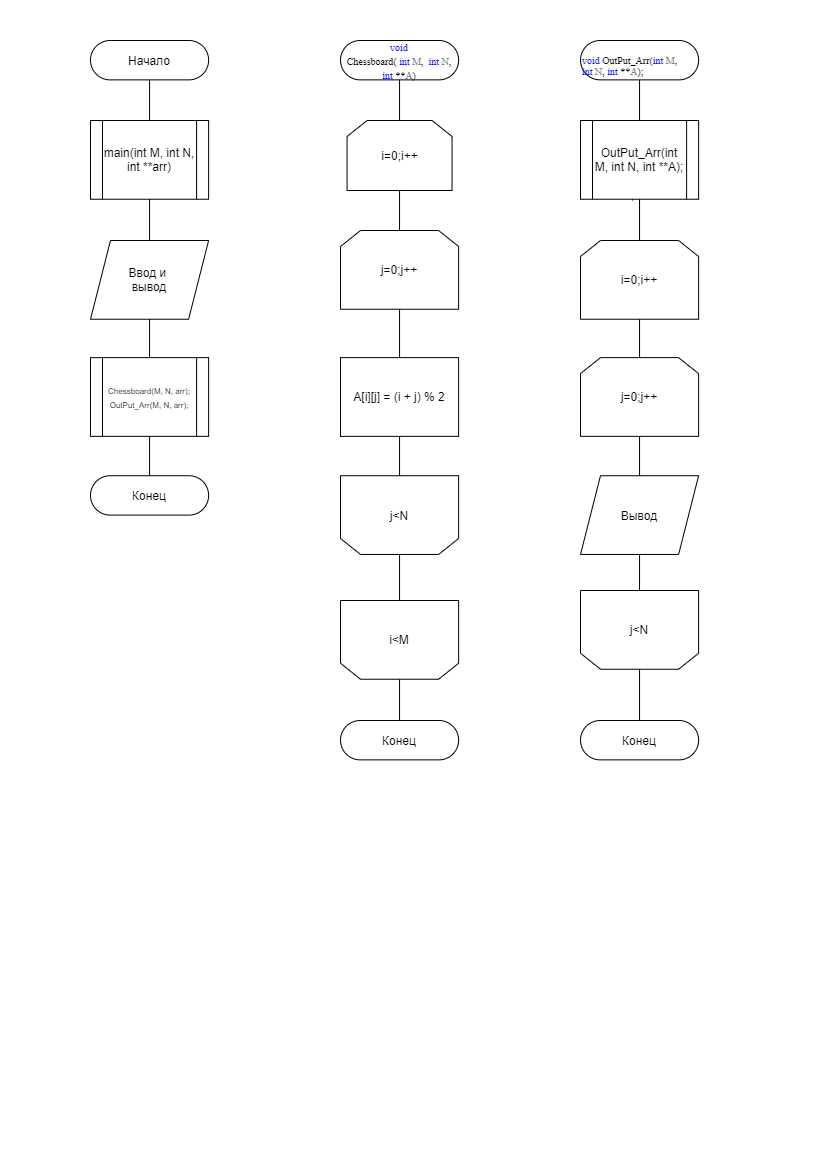
cout << A[i][j] << "\t";

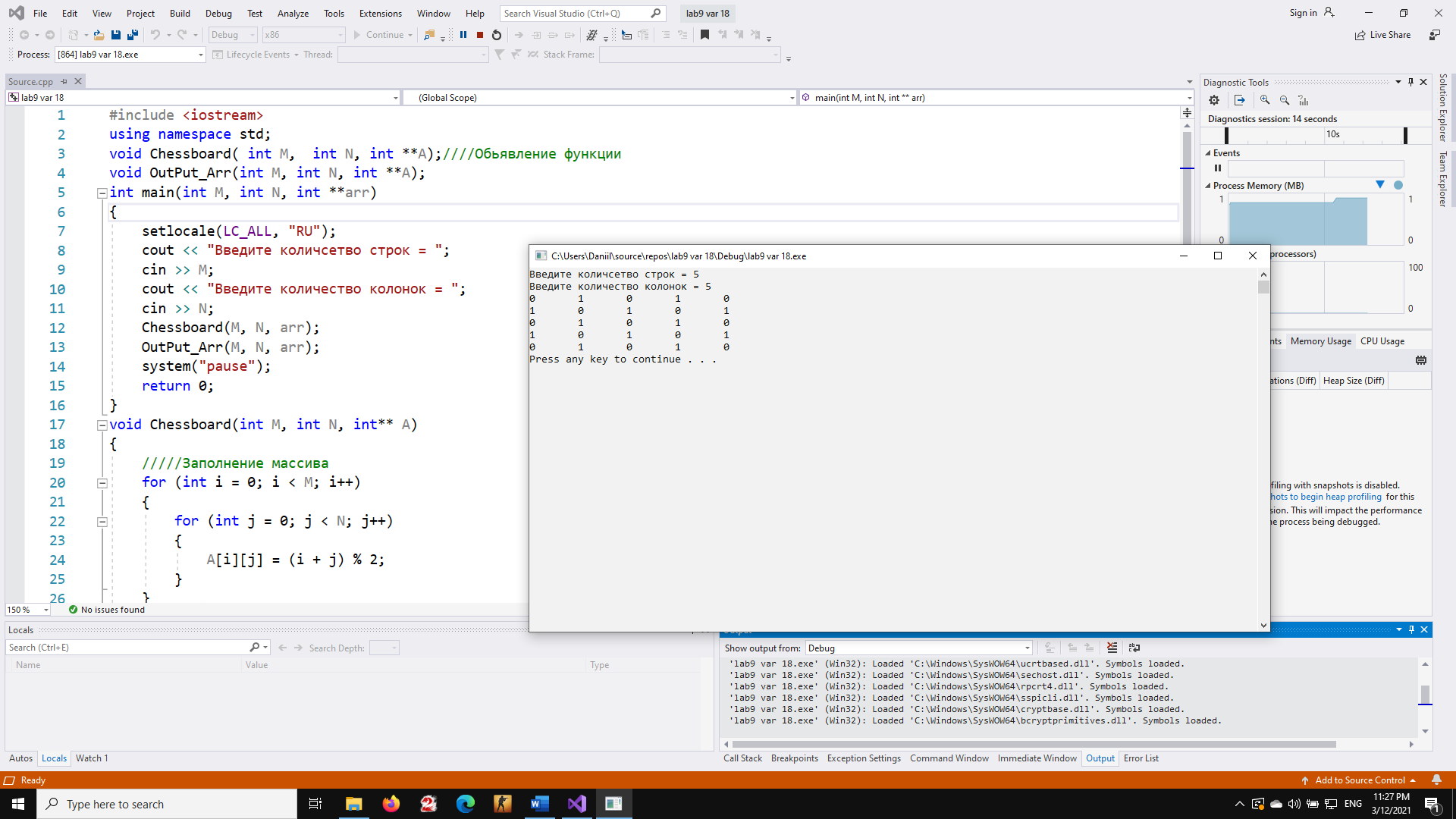
}

cout << endl;

}

}





**Задача 2 «строки»**

**Вариант 14**

Описать функцию DecompressStr(S) строкового типа, восстанавливающую строку, сжатую процедурой CompressStr (см. задание 13). Параметр S содержит сжатую строку; восстановленная строка является возвращаемым значением функции. С помощью функции DecompressStr восстановить пять данных сжатых строк.

#include <iostream>

#include <stdlib.h>

using namespace std;

void DecompressStr(char\* S);

int n = 0;

int main()

{

setlocale(LC\_ALL, "RU");

char STroka1[20];

cout << "Введите строку 1: ";

cin >> STroka1;

char STroka2[20];

cout << "Введите строку 2: ";

cin >> STroka2;

char STroka3[20];

cout << "Введите строку 3: ";

cin >> STroka3;

char STroka4[20];

cout << "Введите строку 4: ";

cin >> STroka4;

char STroka5[20];

cout << "Введите строку 5: ";

cin >> STroka5;

cout << "После преобразования " << endl;

cout << "Строка 1: ";

DecompressStr(STroka1);

cout << endl;

cout << "Строка 2: ";

DecompressStr(STroka2);

cout << endl;

cout << "Строка 3: ";

DecompressStr(STroka3);

cout << endl;

cout << "Строка 4: ";

DecompressStr(STroka4);

cout << endl;

cout << "Строка 5: ";

DecompressStr(STroka5);

cout << endl;

system("pause");

return 0;

}

void DecompressStr(char\* S)

{

for (const char\* symbol = S; \*symbol; symbol++)

{

if (\*symbol == 'c')

{

symbol++;

if (\*symbol == '{')

{

symbol++;

if (isdigit(\*symbol))

{

symbol++;

if (\*symbol == '}')

{

symbol--;

n = atoi(symbol);

}

}

}

}

}

for (const char\* sym = S; \*sym; sym++)

{

if (\*sym == 'c')

{

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

{

cout << "c";

}

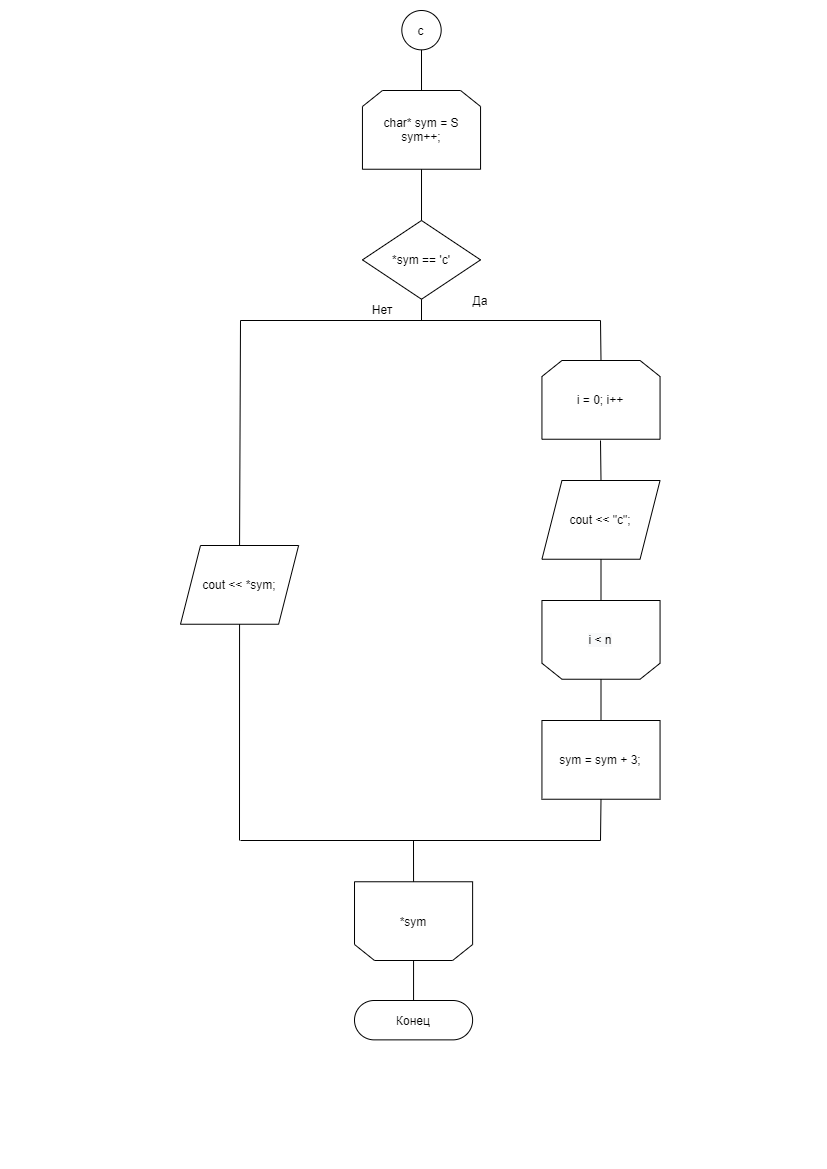
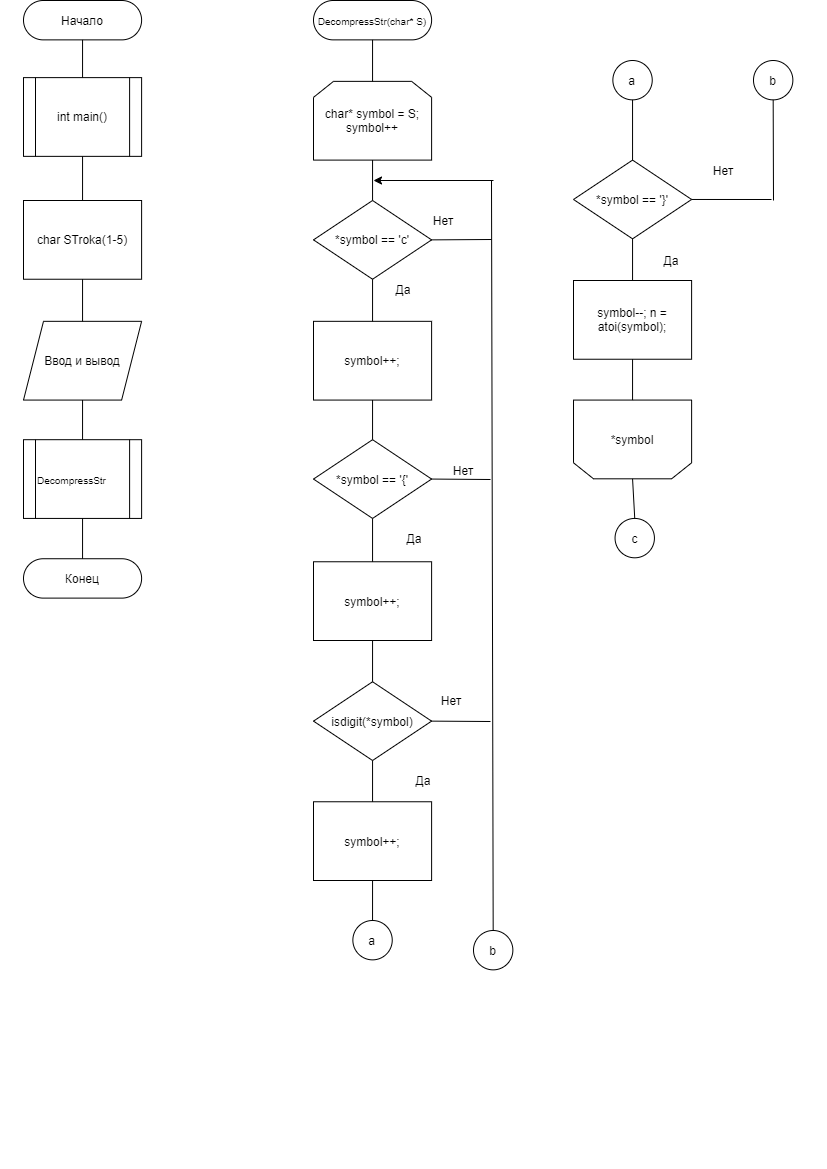
sym = sym + 3;

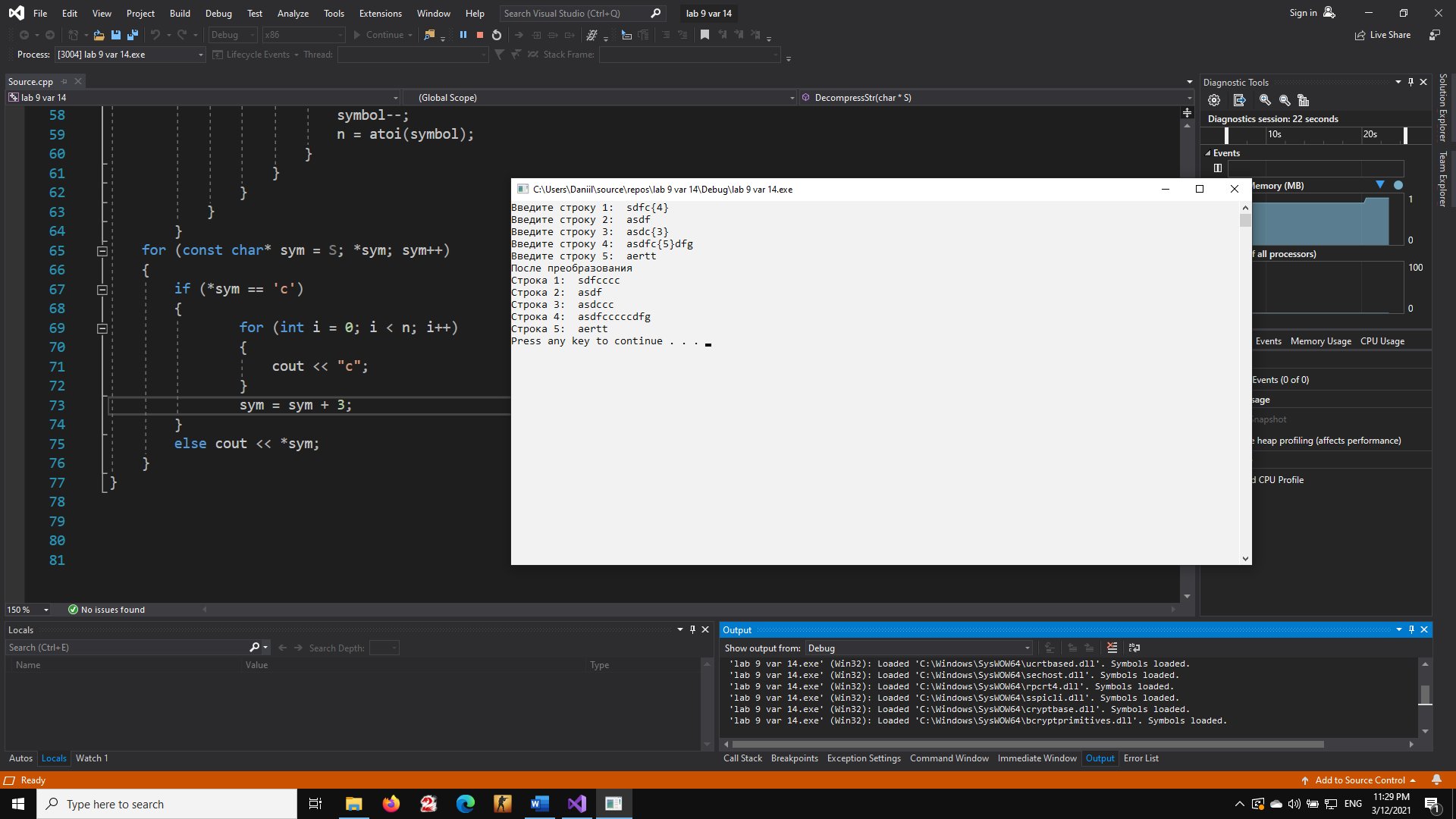
}

else cout << \*sym;

}

}





**Задача 3 «рекурсия»**

**Вариант 11**

Описать рекурсивную функцию MaxInt(A, N) целого типа, которая находит максимальный элемент целочисленного массива A размера N (1 ≤ N ≤ 10), не используя оператор цикла. С помощью этой функции найти максимальные элементы массивов A, B, C размера NA, NB, NC соответственно.

#include <iostream>

#include <ctime>

using namespace std;

int max;

int MaxInt(int A[], int N);//Обьявление функции.

int main()

{

srand(time(0));

setlocale(LC\_ALL, "RU");

int size;

cout << "Введите количество элеметнов первого массива от 1 до 10 : " ;

cin >> size;

if (size < 1 || size>10)

{

return 0;

}

int sizeB;

cout << "Введите количество элеметнов второго массива от 1 до 10 : ";

cin >> sizeB;

if (sizeB < 1 || sizeB>10)

{

return 0;

}

int sizeC;

cout << "Введите количество элеметнов третьего массива от 1 до 10 : ";

cin >> sizeC;

if (sizeC < 1 || sizeC>10)

{

return 0;

}

int Array\_A[10];

int Array\_B[10];

int Array\_C[10];

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

{

Array\_A[i] = rand() % 100;

}

cout << "Массив 1" << endl;

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

{

cout << Array\_A[i] << "\t";

}

cout << endl;

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

{

Array\_B[i] = rand() % 100;

}

cout << "Массив 2" << endl;

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

{

cout << Array\_B[i] << "\t";

}

cout << endl;

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

{

Array\_C[i] = rand() % 100;

}

cout << "Массив 3" << endl;

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

{

cout << Array\_C[i] << "\t";

}

MaxInt(Array\_A, size);

cout << endl;

cout << "\nMax 1 = " << MaxInt(Array\_A, size) << endl;

max = 0;

MaxInt(Array\_B, sizeB);

cout << "\nMax 2= " << MaxInt(Array\_B, size) << endl;

max = 0;

MaxInt(Array\_C, sizeC);

cout << "\nMax 3= " << MaxInt(Array\_C, size) << endl;

cout << endl;

system("pause");

return 0;

}

int MaxInt(int A[], int N)

{

if (N > 0)

{

if (A[N - 1] > max)

{

max = A[N - 1];

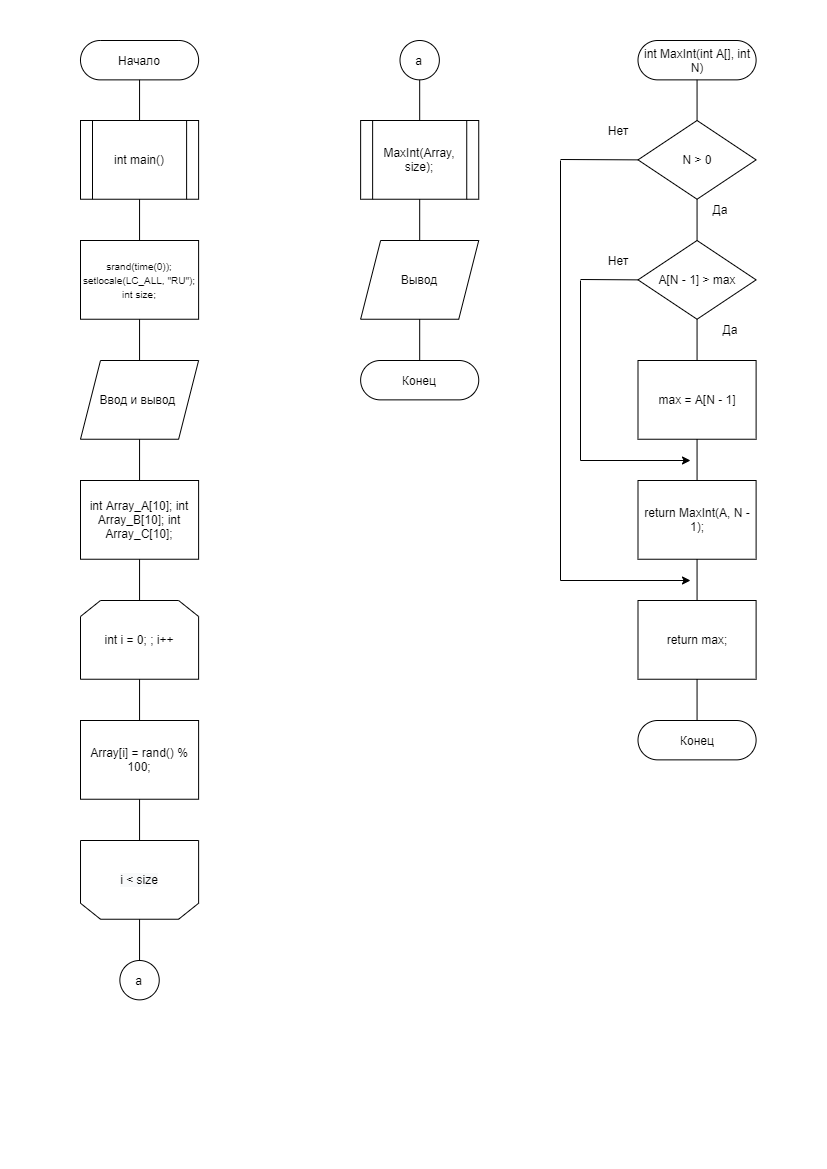
}

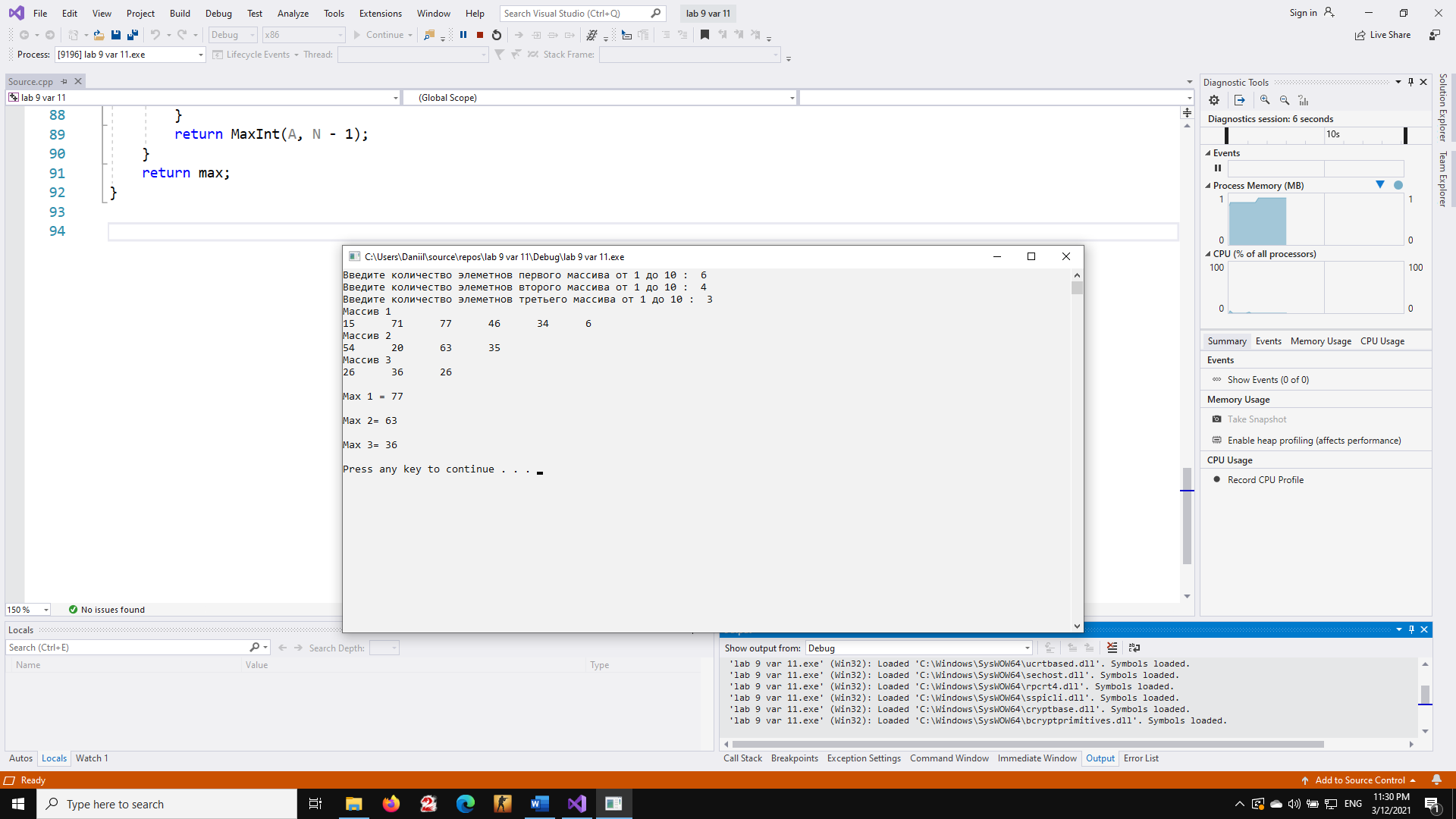
return MaxInt(A, N - 1);

}

return max;

}





**Вывод**: изучил основные принципы написания пользовательских функций, ознакомился с возможностями передачи данных в функции и получения результата по итогам работы функции. Реализовал собственные функции для обработки данных составных и простых типов.