Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики

и радиоэлектроники»

Специальность «Программная инженерия»

Кафедра инженерной психологии и эргономики

Учебная дисциплина «Основы алгоритмизации и программирования»

ОТЧЕТ

по лабораторной работе №10

«Алгоритмы сортировки и поиска»

Вариант 14

Подготовил: Зайцев Е. А.

Проверил: Усенко Ф.В.

Минск 2025

**Цель работы:** Сформировать знания и умения по изучению методов внутренних сортировок. Приобретение навыков реализации алгоритмов сортировок.

**Задание:** Отсортировать элементы квадратной вещественной матрицы размерности n, стоящие на побочной диагонали, применив сортировку бинарным включением.

Код программы приведен ниже:

#include <iostream>

using namespace std;

//функция сортировки

void sorting(int\* diag, int n) {

for (int i = 1; i < n; i++) {

int val = diag[i];

int left = 0, right = i - 1;

int pos = i; // по умолчанию вставляем в конец

// Бинарный поиск места для вставки

while (left <= right) {

int mid = (left + right) / 2;

if (val < diag[mid]) {

right = mid - 1;

pos = mid;

}

else {

left = mid + 1;

}

}

// Сдвигаем элементы и вставляем val

for (int j = i; j > pos; j--) {

diag[j] = diag[j - 1];

}

diag[pos] = val;

}

}

int main() {

int n;

cout << "Enter n: \n";

cin >> n;

//Создаем двумерный динамический массив

int\*\* matrix = new int\*[n];

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

matrix[i] = new int[n];

}

//Заполняем массив с клавиатуры

cout << "Enter elements of matrix:\n";

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

for (int j = 0; j < n; j++) {

cin >> matrix[i][j];

}

}

//Извлечем побочную диагональ

int\* diag = new int[n];

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

diag[i] = matrix[n-1-i][i];

}

sorting(diag, n);

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

matrix[n - 1 - i][i] = diag[i];

}

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

cout << endl;

for (int j = 0; j < n; j++) {

cout << matrix[i][j] << " ";

}

}

// Освобождение памяти

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

delete[] matrix[i];

}

delete[] matrix;

delete[] diag;

return 0;

Результат работающей программы представлен на рисунке 1:

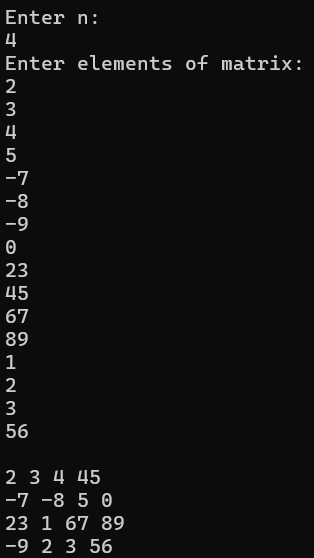


Рисунок 1 – Результат работы программы

Блок-схема работы программы представлена на рисунке 2-3.

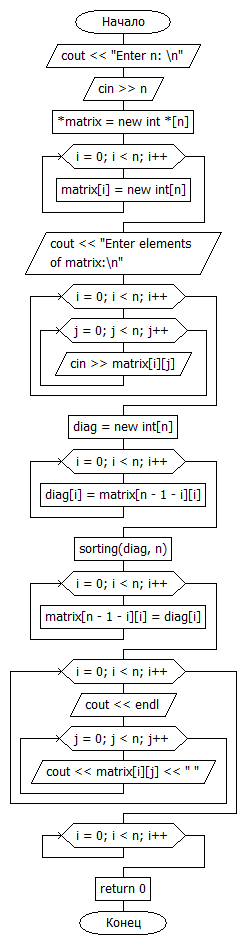
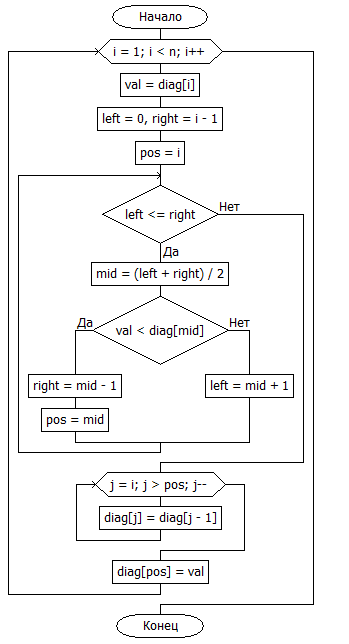
 

Рисунок 2 - main() Рисунок 3 - sorting()

**Вывод:** в ходе выполнения работы была достигнута цель данной лабораторной работы: сформировать знания и умения по изучению методов внутренних сортировок. Приобретение навыков реализации алгоритмов сортировок.