МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Преподаватель

канд. техн. наук, доцент Л.Н. Бариков

Отчёт

по лабораторной работе №10

по дисциплине ОСНОВЫ ПРОГРАММИРОВАНИЯ

на тему: «Методы сортировки»

Работу выполнил

студент гр. 4041 А. Комолкин

Санкт-Петербург

2021

***Цель лабораторной работы:*** *изучение методов сортировки структур данных; совершенствование навыков процедурного программирования на языке C/C++ при решении задач обработки статических и динамических массивов.*

***Задание на программирование:*** *используя технологию процедурного программирования реализовать заданный метод сортировки и применить его для указанных фрагментов числовой матрицы в соответствии с индивидуальным заданием.*

***Порядок выполнения работы:***

1. Получить у преподавателя индивидуальное задание: метод сортировки и вид сортируемых фрагментов матрицы. В зависимости от задания стрелка на рисунке указывает направление возрастания или убывания значений элементов фрагментов матрицы после сортировки.

2. Исходная матрица, содержащая 2\**n* строк и 2\**n* столбцов, не должна содержать элементов с одинаковыми значениями.

3. Разработать математическую модель: описать с помощью формул и рисунков структуру массива и процесс его преобразования. У результирующей матрицы должны быть отсортированы заданные фрагменты, а значения элементов не сортируемых фрагментов должны быть обнулены.

4. Построить схему алгоритма решения задачи.

5. Составить программу на языке *C*/*C*++.

6. Входные данные на этапах тестирования и демонстрации работы преподавателю должны задаваться либо с использованием специально подобранных арифметических формул, либо вводиться с клавиатуры по запросу. **Датчики псевдослучайных чисел использовать запрещается**.

7. Выходные данные должны выводиться на экран с пояснениями. Операторы вывода результатов работы должны находиться либо в функции *main*(), либо в специальной функции вывода (например, преобразованного массива), вызов которой осуществляется из функции *main*().

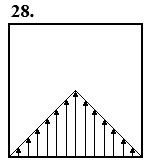
8. Проверить и продемонстрировать преподавателю работу программы на полном наборе тестов, в том числе с ошибочными входными данными. Входные и выходные массивы должны выводиться в одном и том же формате.

9. Использовать стандартные потоковые объекты ввода/вывода ***cin*** и ***cout***.

10. Оформить отчет о лабораторной работе в составе: постановка задачи, математическая модель, схема алгоритма решения, текст программы, контрольные примеры.

**Задание 4/28:**

Сортировка по убыванию методом выбора максимума.



**Математическая модель:**

Метод сортировки: lля сортировки массива методом выбора от наименьшего до наибольшего элемента выполняются следующие шаги:

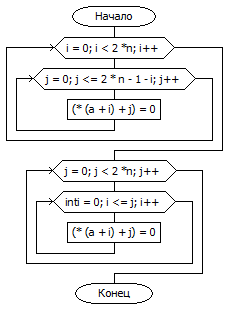
1. Начиная с элемента под индексом 0, ищем в массиве наименьшее значение.
2. Найденное значение меняем местами с нулевым элементом.
3. Повторяем шаги №1 и №2 уже для следующего индекса в массиве (отсортированный элемент больше не трогаем).

Обнуление матрицы происходит по принципу: в начале все выше побочной диагонали, затем выше главной.

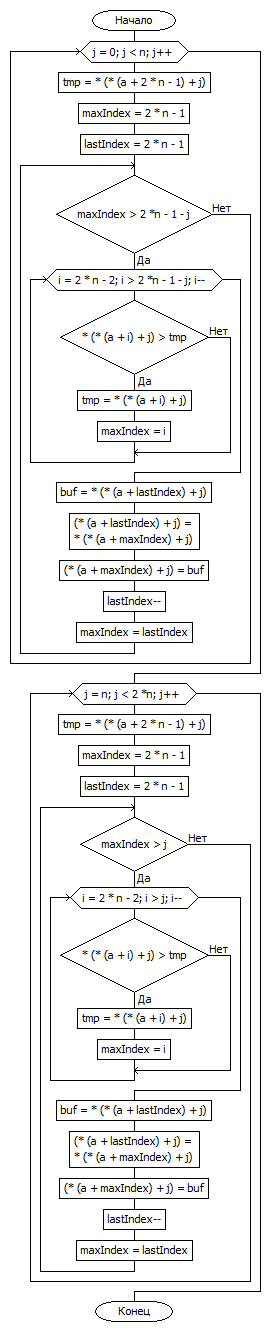
Сортировка проходи по столбцам. До середины берутся элементы с конца строки до побочной диагонали, а после середины до главной. Задача решена.

**Блок-схемы:**

Функция onNull()



Функция Sort



**Текст программы:**

#include <iostream>

#include <math.h>

#include <string>

using namespace std;

typedef int elem; // тип элемент

typedef elem\* mas; //тип "указатель на элемент"

typedef mas\* matr; // тип "указатель на массив"

void Input\_Matr(matr A, int N);

void Output\_Matr(matr A, int N);

void obNull(matr a, int n);

void Sort(matr a, int size);

int main()

{

setlocale(LC\_ALL, "Russian");

cout << " Программа сортировки матрицы по возрастанию методом обмена без флага перестановки\n Введите n, учитывая, что размер матрицы 2n\*2n:\n";

int n;

cin >> n;

cout << "Введите матрицу построчно: \n";

matr A;

A = new mas[2 \* n]; //выделение динамической памяти под массив

//указателей на строки массива

for (int i = 0; i < 2 \* n; i++) //выделение памяти под каждую строку

\*(A + i) = new elem[2 \* n]; //каждому элементу массива указателей

//на строки присваивается адрес начала

//области памяти, выделяемой под строку

Input\_Matr(A, 2 \*n); //ввод матрицы

Output\_Matr(A, 2 \*n);

obNull(A, n);

Sort(A, n); //сортируем

cout << "Отсортированная матрица: \n";

Output\_Matr(A, 2 \* n); //выводим матрицу.

}

void Input\_Matr(matr A, int N)

{

cout << " Значения элементов";

cout << " массива размера " << N << "x" << N;

cout << "\n при тестировании вводятся автоматически:" << endl;

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

{

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

\*(\*(A + i) + j) = -10 \* i + j;

}

}

void Output\_Matr(matr A, int N)

{

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

{

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

{

cout.width(3); //ширина поля выводимого параметра

cout << \*(\*(A + i) + j) << ' ';

}

cout << '\n';

}

cout << '\n';

}

void Sort(matr a, int n) {

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

{

elem tmp = \*(\*(a + 2 \* n - 1) + j);

int maxIndex = 2 \* n - 1;

int lastIndex = 2 \* n - 1;

while (maxIndex > 2 \* n - 1 - j)

{

for (int i = 2 \* n - 2; i > 2 \* n - 1 - j; i--)

{

if (\*(\*(a + i) + j) > tmp)

{

tmp = \*(\*(a + i) + j);

maxIndex = i;

}

}

elem buf = \*(\*(a + lastIndex) + j);

\*(\*(a + lastIndex) + j) = \*(\*(a + maxIndex) + j);

\*(\*(a + maxIndex) + j) = buf;

lastIndex--;

maxIndex = lastIndex;

}

}

for (int j = n; j < 2\*n; j++)

{

elem tmp = \*(\*(a + 2 \* n - 1) + j);

int maxIndex = 2 \* n - 1;

int lastIndex = 2 \* n - 1;

while (maxIndex > j)

{

for (int i = 2 \* n - 2; i > j; i--)

{

if (\*(\*(a + i) + j) > tmp)

{

tmp = \*(\*(a + i) + j);

maxIndex = i;

}

}

elem buf = \*(\*(a + lastIndex) + j);

\*(\*(a + lastIndex) + j) = \*(\*(a + maxIndex) + j);

\*(\*(a + maxIndex) + j) = buf;

lastIndex--;

maxIndex = lastIndex;

}

}

}

void obNull(matr a, int n)

{

for (int i = 0; i < 2 \* n; i++)

{

for (int j = 0; j <= 2 \* n - 1 - i; j++)

{

\*(\*(a + i) + j) = 0;

}

}

for (int j = 0; j < 2 \* n; j++)

{

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

{

\*(\*(a + i) + j) = 0;

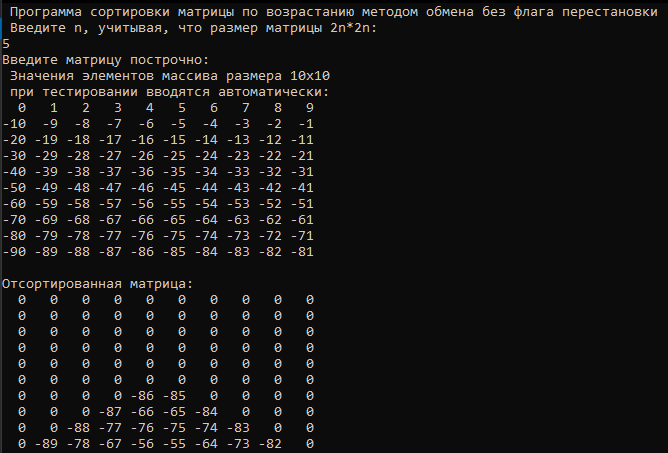
}

}

}

}

**Скриншоты работы программы**



**Вывод:** изучил методы сортировки структур данных; усовершенствовал навыков процедурного программирования на языке C/C++ при решении задач обработки статических и динамических массивов.