# МИНОБРНАУКИ РОССИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

# «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

**Кафедра информационных систем**

# ОТЧЕТ

**по практической работе №3**

# по дисциплине «Программирование» Тема:  "**ДВУМЕРНЫЕ СТАТИЧЕСКИЕ МАССИВЫ. УКАЗАТЕЛИ"**

|  |  |
| --- | --- |
| Студентка гр. 3372 | Козина П.С. |
| Преподаватель | Глущенко А. Г. |

Санкт-Петербург 2023

# Цель работы.

Разработать программу, позволяющая выполнять различные операции с массивами, включая сортировку и поиск.

# Основные теоретические положения.

Массив – структура данных, хранящая набор значений, идентифицируемых по индексу или набору индексов, принимающих целые значения из некоторого заданного непрерывного диапазона. Одномерный массив можно рассматривать как реализацию абстрактного типа данных. Подробная информация о массиве представлена на рисунке 1.



Рисунок 1 – Одномерный массив

Сортировка массива – это расположение элементов массива в некотором заданном порядке (по возрастанию или убыванию). Например, на рисунке 2 представлена сортировка пузырьком.

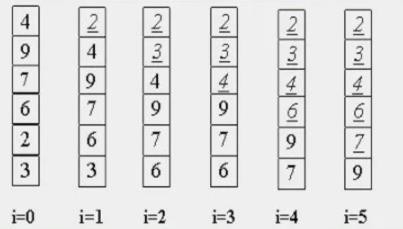


Рисунок 2 – Сортировка пузырьком В данной работе представлены следующие сортировки:

* сортировка полным перебором (от меньше к большему) – это сортировка выполняющая полный перебор всех элементов массива сравнивая и переставляя их при удовлетворенности условий;
* «быстрая сортировка» – один из самых популярных алгоритмов, в котором используется принцип «разделяй и властвуй»;
* сортировка «слиянием» – это сортировка, где мы делится массив пополам, каждый из них сортируется слиянием и потом соединяются оба массива.

Поиск элемента в массиве дает возможность удостоверится о наличии данного элемента в части массива.

В неотсортированном массиве поиск элемента необходимо просматривать весь массив (линейный поиск).

В отсортированном массиве поиск элемента можно осуществлять, не просматривая весь массив. Например, бинарный поиск представлен на рисунке 3.

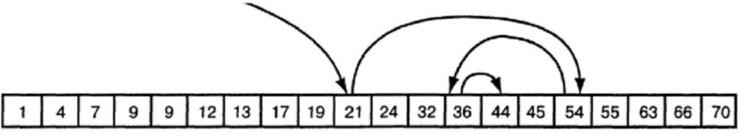


Рисунок 3 – Бинарный поиск

# Постановка задачи.

Разработать алгоритм и написать программу, которая позволяет:

1. Создать целочисленный массив размерности N = 100. Элементы массивы должны принимать случайное значение в диапазоне от -99 до 99.
2. Отсортировать заданный в пункте 1 массив […] сортировкой (от меньшего к большему). Определить время, затраченное на сортировку, используя библиотеку chrono.
3. Найти максимальный и минимальный элемент массива. Подсчитать время поиска этих элементов в отсортированном массиве и неотсортированном, используя библиотеку chrono.
4. Выводить среднее значение (если необходимо, число нужно округлить) максимального и минимального значения в отсортированном и неотсортированном. Выводить индексы всех элементов, которые равны этому значению, и их количество. Подсчитать время поиска.
5. Выводить количество элементов в отсортированном массиве, которые меньше числа a, которое инициализируется пользователем.
6. Выводить количество элементов в отсортированном массиве, которые больше числа b, которое инициализируется пользователем.
7. Выводит информацию о том, есть ли введенное пользователем число в отсортированном массиве. Реализуйте алгоритм бинарного поиска. Сравните скорость его работы с обычным перебором.
8. Менять местами элементы массива, индексы которых вводит пользователь. Выводить скорость обмена, используя библиотеку chrono.

# Выполнение работы.

Код программы представлен в приложении А.

# Выводы.

Данная практическая работа позволяет сформировать навыки и умения работы с массивами, усвоить принцип сортировки и поиска, сравнить время работы различных функций и методов.

# Ход работы

1. При запуске программы необходимо выбрать номер действия (рисунок 1).

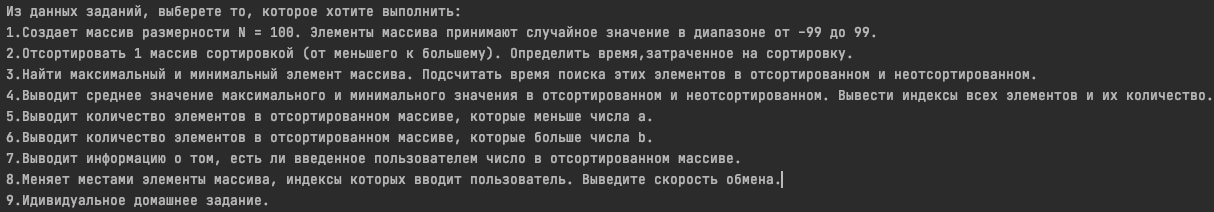


Рисунок 1 – Выбор номер поведения программы

1. При выборе пункта «1» выводится массив размерности N = 100. Элементы массива принимают случайное значение в диапазоне от -99 до 99 (рисунок 2).

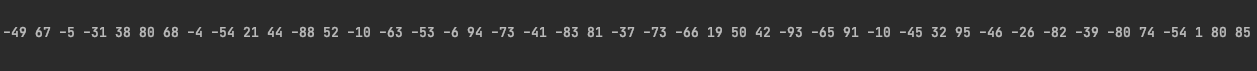


Рисунок 2 – Вывод массива

1. При выборе пункта «2» осуществляется сортировка «пузырьком» и отображение время выполнения сортировки (рисунок 3).

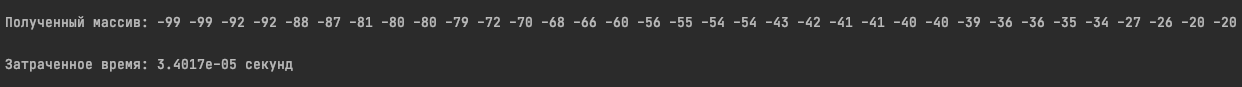


Рисунок 3 – Сортировка «пузырьком»

1. При выборе пункта «3» осуществляется поиск минимального и максимального элемента в неотсортированном и отсортированном массиве и расчет времени этого поиска (рисунок 4).

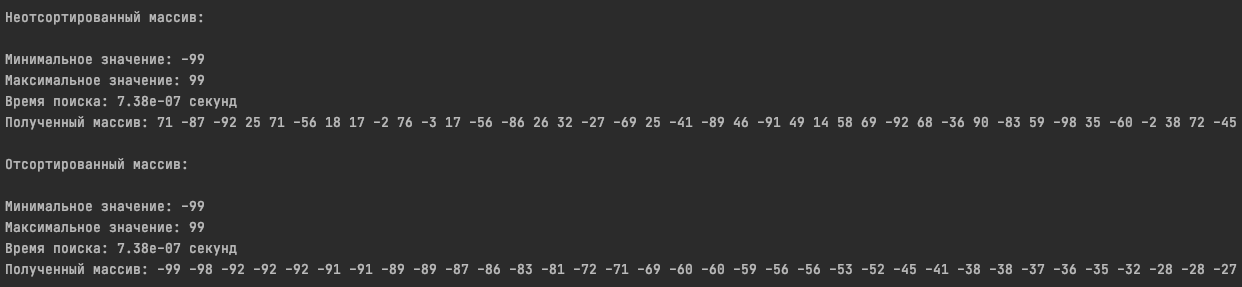


Рисунок 4 – Сортировка «слиянием»

1. При выборе пункта «4» отображается среднее минимального и максимального элемента в неотсортированном и отсортированном массиве, также выводятся индексы всех элементов и их количество и время поиска (рисунок 5)

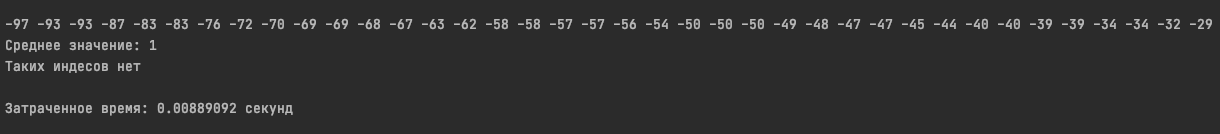


Рисунок 5 – Подсчет и поиск среднего в массиве в неотсортированном и отсортированном массиве

6. При выборе пункта «5» и «6» отображается количество элементов массива, меньше и больше введенного значения пользователя соответственно (рисунок 6,7).

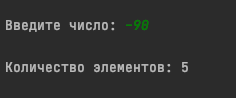


Рисунок 6 – Отображение количества элементов, меньше введённого значения

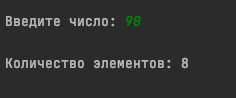


Рисунок 7 – Отображение количества элементов, больше введённого значения

1. При выборе пункта «7» выводится информацию о том, есть ли введенное пользователем число в отсортированном массиве, также сравнивает скорость работы бинарного поиска с обычным перебором (рисунок 8)

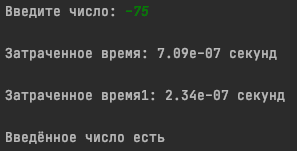


Рисунок 8 – Отображение информации о том, что есть введённое число в массиве и сравнение скорости работы бинарного поиска с обычным перебором

1. При выборе пункта «8» осуществляется смена элементов массива при помощи индексов, введенных пользователем и выводится скорость обмена (рисунок 9)

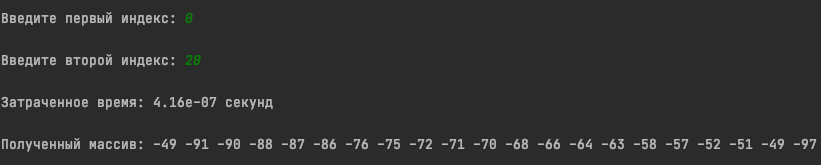


Рисунок 9 – Смена элементов

# ПРИЛОЖЕНИЕ А РАБОЧИЙ КОД

Листинг А.1 – Рабочий код программы

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "ru");

srand(time(NULL));

int choose = 0, idz = 0;

while (choose != -1) {

cout << "\n\n\nИз данных заданий, выберете то, которое хотите выполнить: \n"

"1.Программа, которая используя арифметику указателей, заполняет квадратичную целочисленную матрицу порядка N (6,8,10) случайными числами от 1 до N\*N согласно схемам.\n"

"2.Программа, которая получает новую матрицу, из матрицы п. 1, переставляя ее блоки в соответствии со схемами.\n"

"3.Программа, которая используя арифметику указателей, сортирует элементы(Insert sort).\n"

"4.Программа, которая уменьшает, увеличивает, умножает или делит все элементы матрицы на введенное пользователем число.\n"

"5.ИДЗ.\n\n\n";

cout << "Задание: ";

cin >> choose;

cout << "\n";

if (choose != 1 and choose != 2 and choose != 3 and choose != 4 and choose != 5 and choose != 0)

cout << "Такого варианта ответа нет \n";

else

switch (choose) {

case 1: {

int N;

cout << "Выберете размер матрицы(6,8,10): ";

cin >> N;

int\*\* arr = new int\* [N];

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

arr[i] = new int[N];

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

arr[i][j] = 0;

}

}

srand(time(NULL));

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

arr[i][0] = rand() % (N \* N) + 1;

}

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

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

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

cout << "\n";

}

int dir = 0; //0 - вправо 1 - влево

int i = N - 1;

int j = 0;

for (int step = N - 1; step > 0; step--) {

if (dir == 0) {

for (int st = 0; st < step; st++) {

j += 1;

arr[i][j] = rand() % (N \* N) + 1;

}

for (int st = 0; st < step; st++) {

i -= 1;

arr[i][j] = rand() % (N \* N) + 1;

}

dir = 1;

}

else{

for (int st = 0; st < step; st++) {

j -= 1;

arr[i][j] = rand() % (N \* N) + 1;

}

for (int st = 0; st < step; st++) {

i += 1;

arr[i][j] = rand() % (N \* N) + 1;

}

dir = 0;

}

cout << "\n";

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

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

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

cout << "\n";

}

cout << "\n";

}

arr = new int\* [N];

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

arr[i] = new int[N];

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

arr[i][j] = 0;

}

}

srand(time(NULL));

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

arr[i][0] = rand() % (N \* N) + 1;

}

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

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

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

cout << "\n";

}

dir = 0; //0 - вверх 1 - вниз

i = N - 1;

j = 0;

for (int k = 0; k < N - 1; k ++) {

if (dir == 0) {

j += 1;

arr[i][j] = rand() % (N \* N) + 1;

for (int st = 0; st < N - 1; st++) {

i -= 1;

arr[i][j] = rand() % (N \* N) + 1;

}

dir = 1;

}

else {

j += 1;

arr[i][j] = rand() % (N \* N) + 1;

for (int st = 0; st < N - 1; st++) {

i += 1;

arr[i][j] = rand() % (N \* N) + 1;

}

dir = 0;

}

cout << "\n";

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

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

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

cout << "\n";

}

cout << "\n";

}

break;

}

case 2: {

int N;

cout << "Выберете размер матрицы(6,8,10): ";

cin >> N;

int\*\* arr = new int\* [N];

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

arr[i] = new int[N];

}

srand(time(NULL));

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

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

\*(\*(arr + i) + j) = i \* N + j + 1;

}

}

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

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

std::cout << arr[i][j] << " ";

}

std::cout << std::endl;

}

cout << '\n';

// 1 2

// 3 4

for (int i = 0; i < N / 2; i++) {//1 2

for (int j = 0; j < N / 2; j++) {

swap(\*(\*(arr + i) + j + N/2), \*(\*(arr + i) + j));

}

}

for (int i = N / 2; i < N; i++) { //1 3

for (int j = 0; j < N / 2; j++) {

swap(\*(\*(arr + i - N / 2) + j), \*(\*(arr + i) + j));

}

}

for (int i = N / 2; i < N; i++) { // 3 4

for (int j = N / 2; j < N; j++) {

swap(\*(\*(arr + i) + j - N/2), \*(\*(arr + i) + j));

}

}

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

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

std::cout << arr[i][j] << " ";

}

std::cout << std::endl;

}

cout << '\n';

for (int i = 0; i < N / 2; i++) { // 1 4

for (int j = 0; j < N / 2; j++) {

swap(\*(\*(arr + i) + j), \*(\*(arr + i + N/2) + j + N/2));

}

}

for (int i = 0; i < N / 2; i++) { // 2 3

for (int j = N/2; j < N; j++) {

swap(\*(\*(arr + i) + j), \*(\*(arr + i + N / 2) + j - N / 2));

}

}

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

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

std::cout << arr[i][j] << " ";

}

std::cout << std::endl;

}

cout << '\n';

for (int i = N / 2; i < N; i++) { //1 3 и 2 4

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

swap(\*(\*(arr + i - N / 2) + j), \*(\*(arr + i) + j));

}

}

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

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

std::cout << arr[i][j] << " ";

}

std::cout << std::endl;

}

cout << '\n';

for (int i = 0; i < N; i++) {//1 2 и 3 4

for (int j = 0; j < N / 2; j++) {

swap(\*(\*(arr + i) + j + N / 2), \*(\*(arr + i) + j));

}

}

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

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

std::cout << arr[i][j] << " ";

}

std::cout << std::endl;

}

cout << '\n';

break;

}

case 3: {

int N;

cout << "Выберете размер матрицы(6,8,10): ";

cin >> N;

int\*\* arr = new int\* [N];

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

arr[i] = new int[N];

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

arr[i][j] = 0;

}

}

srand(time(NULL));

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

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

arr[i][j] = rand() % (N \* N) + 1;

}

}

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

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

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

cout << "\nn";

}

int row1, row2, col1, col2;

for (int i = 0; i < N\*N; i++) {

for (int j = i - 1; j > 0; j--) {

row2 = j / N;

col2 = j % N;

row1 = (j - 1) / N;

col1 = (j - 1) % N;

if (\*(\*(arr + row1) + col1) > \*(\*(arr + row2) + col2)) {

swap(\*(\*(arr + row1) + col1), \*(\*(arr + row2) + col2));

}

else {

break;

}

}

}

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

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

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

cout << "\n";

}

}

case 4: {

int N = 0;

cout << "Выберете размер матрицы(6,8,10): ";

cin >> N;

int my\_number = 0;

char sign = 0;

cout << "Введите число:\n";

cin >> my\_number;

cout << "Введите знак операции:\n";

cin >> sign;

int\*\* arr = new int\*[N];

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

arr[i] = new int[N];

}

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

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

\*(\*(arr + i) + j) = i \* N + j + 1;

}

}

cout << "\n";

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

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

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

cout << "\n";

}

cout << "\n";

switch (sign) {

case '+': {

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

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

\*(\*(arr + i) + j) += my\_number;

std::cout << arr[i][j] << " ";

}

std::cout << std::endl;

}

break;

}

case '-': {

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

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

\*(\*(arr + i) + j) -= my\_number;

std::cout << arr[i][j] << " ";

}

std::cout << std::endl;

}

break;

}

case '/': {

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

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

\*(\*(arr + i) + j) /= my\_number;

std::cout << arr[i][j] << " ";

}

std::cout << std::endl;

}

break;

}

case '\*': {

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

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

\*(\*(arr + i) + j) \*= my\_number;

std::cout << arr[i][j] << " ";

}

std::cout << std::endl;

}

break;

}

}

break;

}

case 5: {

int N;

cout << "Выберете размер матрицы: ";

cin >> N;

int\*\* arr1 = new int\* [N];

int\*\* arr2 = new int\* [N];

int\*\* resArr = new int\* [N];

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

arr1[i] = new int[N];

arr2[i] = new int[N];

resArr[i] = new int[N];

}

srand(time(NULL));

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

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

\*(\*(arr1 + i) + j) = rand() % (N \* N) + 1;

\*(\*(arr2 + i) + j) = rand() % (N \* N) + 1;

}

}

cout << "\n\nПервая матрица:\n\n";

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

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

cout << arr1[i][j] << " ";

cout << "\n";

}

cout << "\n\nВторая матрица:\n\n";

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

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

cout << arr2[i][j] << " ";

cout << "\n";

}

cout << "\n\nСумма:\n\n";

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

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

resArr[i][j] = arr1[i][j] + arr2[i][j];

cout << arr1[i][j] << " + " << arr2[i][j] << " = " << resArr[i][j] << " ";

}

cout << "\n";

}

cout << "\n\nРезультат:\n\n";

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

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

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

cout << "\n";

}

break;

}

}

}

return 0;

}