# 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: Одномерные статические массивы

|  |  |
| --- | --- |
| Студентка гр. 3372 | Шорсткая А.А. |
| 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.

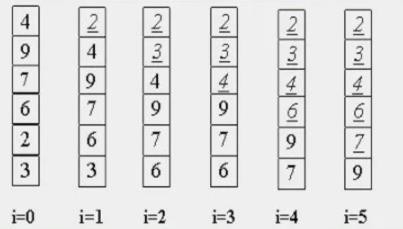


Рисунок 2 – Сортировка пузырьком В данной работе представлены следующие сортировки:

* 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.

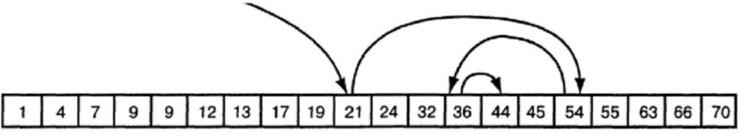


Рисунок 3 – Бинарный поиск

# 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).

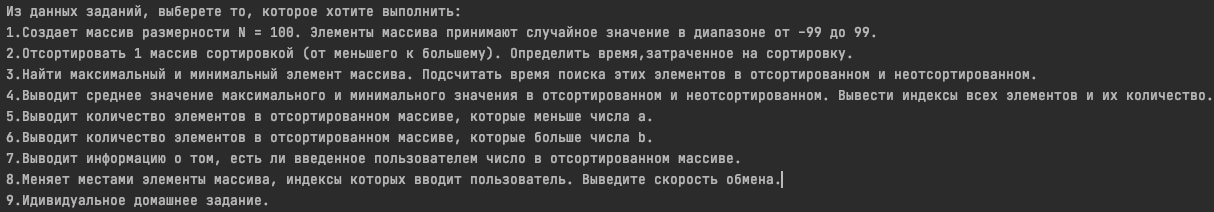


Рисунок 1 – Выбор номер поведения программы

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

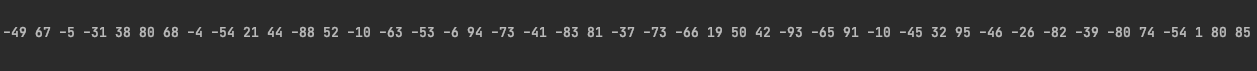


Рисунок 2 – Вывод массива

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

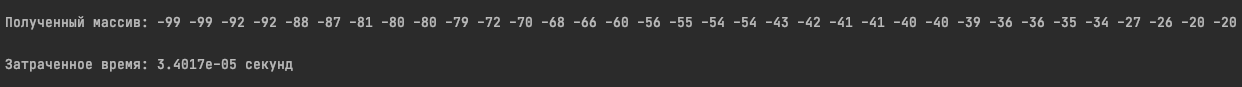


Рисунок 3 – Сортировка «пузырьком»

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

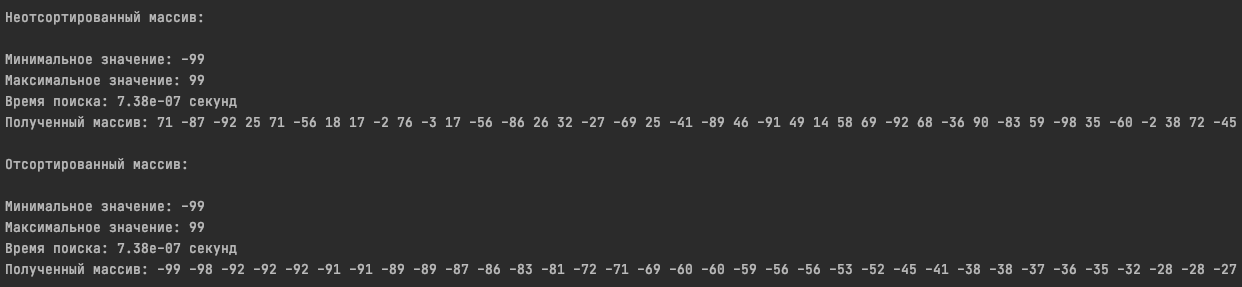


Рисунок 4 – Сортировка «слиянием»

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)

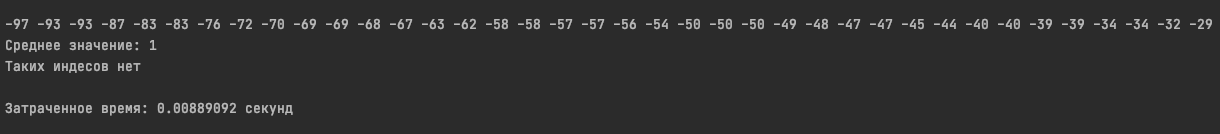


Рисунок 5 – Подсчет и поиск среднего в массиве в неотсортированном и отсортированном массиве

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

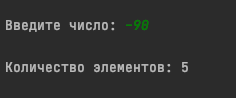


Рисунок 6 – Отображение количества элементов, меньше введённого значения

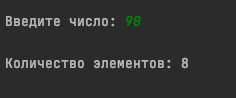


Рисунок 7 – Отображение количества элементов, больше введённого значения

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)

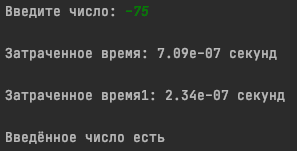


Рисунок 8 – Отображение информации о том, что есть введённое число в массиве и сравнение скорости работы бинарного поиска с обычным перебором

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

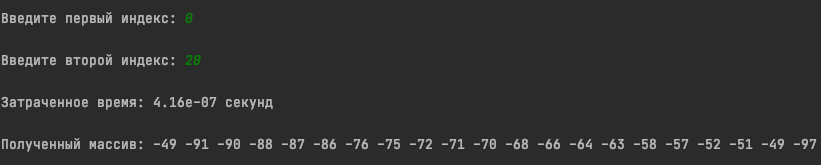


Рисунок 9 – Смена элементов

# ПРИЛОЖЕНИЕ А РАБОЧИЙ КОД

Листинг А.1 – Рабочий код программы

#include <iostream>  
using namespace std;  
  
int main() {  
 setlocale(LC\_ALL, "ru");  
 srand(time(NULL));  
 bool task = true;  
 int choose = 0, idz = 0, idzChoose = -1;  
  
 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 пузырьком.\n"  
 "4.Программа, которая уменьшает, увеличивает, умножает или делит все элементы матрицы на введенное пользователем число.\n"  
 "5.Идивидуальное домашнее задание.\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 0: {  
 cout << "Пока \n\n";  
 choose = -1;  
 break;  
 }  
  
 case 1: {  
 int N;  
 cout << "Выберете размер матрицы(6,8,10): ";  
 cin >> N;  
 int arr[N][N];  
  
 srand(time(NULL));  
  
 int \*ptr = &arr[0][0];  
  
 for (int i = 0; i < N \* N; i++) {  
 //\*(ptr + i) = 0;  
 \*(ptr + i) = rand() % (N \* N) + 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";  
  
 int Xstr = 0, Xfin = 0, Ystr = 0, Yfin = 0;  
  
 int k = 1;  
 int i = 0;  
 int j = 0;  
  
 while (k <= N \* N) {  
 \*(ptr + (i \* N + j)) = rand() % (N \* N) + 1;  
 if (i == Xstr && j < N - Yfin - 1)  
 ++j;  
 else if (j == N - Yfin - 1 && i < N - Xfin - 1)  
 ++i;  
 else if (i == N - Xfin - 1 && j > Ystr)  
 --j;  
 else  
 --i;  
  
 if ((i == Xstr + 1) && (j == Ystr) && (Ystr != N - Yfin - 1)) {  
 ++Xstr;  
 ++Xfin;  
 ++Ystr;  
 ++Yfin;  
 }  
 ++k;  
 }  
  
 for (int x = 0; x < N; ++x) {  
 for (int y = 0; y < N; ++y)  
 cout << arr[x][y] << " ";  
 cout << "\n";  
 }  
 cout << "\n" << "\n";  
  
  
 for (int i = 0; i < N \* N; i++) {  
 //\*(ptr + i) = 0;  
 \*(ptr + i) = rand() % (N \* N) + 1;  
 }  
 for (int x = 0; x < N; ++x) {  
 for (int y = 0; y < N; ++y)  
 cout << arr[x][y] << " ";  
 cout << "\n";  
 }  
 cout << "\n";  
 for (int ik = 0; ik < N; ik++) {  
 for (int jk = 0; jk < N; jk++) {  
 int value = ik \* N + jk + 1;  
  
 Настя, [15.12.2023, 20:57:48]:  
 if (ik % 2 == 0) {  
 \*(ptr + jk \* N + ik) = value;  
 } else {  
 \*(ptr + ((N - jk - 1) \* N + ik)) = value;  
 }  
 }  
 }  
 for (int x = 0; x < N; ++x) {  
 for (int y = 0; y < N; ++y)  
 cout << arr[x][y] << " ";  
 cout << "\n";  
 }  
 break;  
 }  
  
 case 2: {  
 int N;  
 cout << "Выберете размер матрицы(6,8,10): ";  
 cin >> N;  
 int arr[N][N];  
  
 srand(time(NULL));  
  
 int \*ptr = &arr[0][0];  
  
 for (int i = 0; i < N \* N; i++) {  
 //\*(ptr + i) = i+1;  
 \*(ptr + i) = rand() % (N \* N) + 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';  
  
 int copya[N][N];  
 int \*ptra = &copya[0][0];  
 for (int i = 0; i < N / 2; i++) {  
 for (int j = 0; j < N / 2; j++) {  
 \*(ptra + i \* N + (j + N / 2)) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = 0; i < N / 2; i++) {  
 for (int j = N / 2; j < N; j++) {  
 \*(ptra + (i + N / 2) \* N + j) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = N / 2; i < N; i++) {  
 for (int j = N / 2; j < N; j++) {  
 \*(ptra + i \* N + (j - N / 2)) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = N / 2; i < N; i++) {  
 for (int j = 0; j < N / 2; j++) {  
 \*(ptra + (i - N / 2) \* N + j) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = 0; i < N; i++) {  
 for (int j = 0; j < N; j++) {  
 std::cout << copya[i][j] << " ";  
 }  
 std::cout << std::endl;  
 }  
 cout << '\n';  
  
  
 int copyb[N][N];  
 int \*ptrb = &copyb[0][0];  
 for (int i = 0; i < N / 2; i++) {  
 for (int j = 0; j < N / 2; j++) {  
 \*(ptrb + (i + N / 2) \* N + (j + N / 2)) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = N / 2; i < N; i++) {  
 for (int j = N / 2; j < N; j++) {  
 \*(ptrb + (i - N / 2) \* N + (j - N / 2)) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = 0; i < N/2; i++) {  
 for (int j = N / 2; j < N; j++) {  
 \*(ptrb + (i + N / 2) \* N + (j - N / 2)) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = N / 2; i < N; i++) {  
 for (int j = 0; j < N/2; j++) {  
 \*(ptrb + (i - N / 2) \* N + (j + N / 2)) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = 0; i < N; i++) {  
 for (int j = 0; j < N; j++) {  
 std::cout << copyb[i][j] << " ";  
 }  
 std::cout << std::endl;  
 }  
 cout << '\n';  
  
   
 int copyc[N][N];  
 int \*ptrc = &copyc[0][0];  
 for (int i = 0; i < N / 2; i++) {  
 for (int j = N / 2; j < N; j++) {  
 \*(ptrc + (i + N / 2) \* N + j) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = N / 2; i < N; i++) {  
 for (int j = N / 2; j < N; j++) {  
 \*(ptrc + (i - N / 2) \* N + j) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = N / 2; i < N; i++) {  
 for (int j = 0; j < N / 2; j++) {  
 \*(ptrc + (i - N / 2) \* N + j) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = 0; i < N / 2; i++) {  
 for (int j = 0; j < N / 2; j++) {  
 \*(ptrc + (i + N / 2) \* N + j) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = 0; i < N; i++) {  
 for (int j = 0; j < N; j++) {  
 std::cout << copyc[i][j] << " ";  
 }  
 std::cout << std::endl;  
 }  
 cout << '\n';  
  
  
 int copyd[N][N];  
 int \*ptrd = &copyd[0][0];  
 for (int i = 0; i < N / 2; i++) {  
 for (int j = 0; j < N / 2; j++) {  
 \*(ptrd + i \* N + (j + N / 2)) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = 0; i < N / 2; i++) {  
 for (int j = N / 2; j < N; j++) {  
 \*(ptrd + i \* N + (j - N / 2)) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = N / 2; i < N; i++) {  
 for (int j = N / 2; j < N; j++) {  
 \*(ptrd + i \* N + (j - N / 2)) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = N / 2; i < N; i++) {  
 for (int j = 0; j < N / 2; j++) {  
 \*(ptrd + i \* N + (j + N / 2)) = \*(ptr + i \* N + j);  
 }  
 }  
 for (int i = 0; i < N; i++) {  
 for (int j = 0; j < N; j++) {  
 std::cout << copyd[i][j] << " ";  
 }  
 std::cout << std::endl;  
 }  
 break;  
 }  
  
 case 3: {  
 int N;  
 cout << "Выберете размер матрицы(6,8,10): ";  
 cin >> N;  
 int arr[N][N];  
  
 srand(time(NULL));  
  
 int \*ptr = &arr[0][0];  
  
 for (int i = 0; i < N \* N; i++) {  
 //\*(ptr + i) = i+1;  
 \*(ptr + i) = rand() % (N \* N) + 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';  
 for (int i = 0; i < N \* N; i++) {  
 for (int j = 0; j < N \* N - 1 - i; j++) {  
 if (\*(ptr + j) > \*(ptr + j + 1)) {  
 int arr1 = \*(ptr + j);  
 \*(ptr + j) = \*(ptr + j + 1);  
 \*(ptr + j + 1) = arr1;  
 }  
 }  
 }  
 for (int i = 0; i < N; i++) {  
 for (int j = 0; j < N; j++) {  
 std::cout << arr[i][j] << " ";  
 }  
 std::cout << std::endl;  
 }  
 break;  
 }  
  
 case 4: {  
 int N = 0;  
 cout << "Выберете размер матрицы(6,8,10): ";  
 cin >> N;  
 int my\_number = 0;  
 char sign = 0;  
 cout << "Введите число: ";  
 std::cin >> my\_number;  
 cout << "Введите знак операции: ";  
 std::cin >> sign;  
 int arr[N][N];  
  
 srand(time(NULL));  
  
 int \*ptr = &arr[0][0];  
  
 for (int i = 0; i < N \* N; i++) {  
 //\*(ptr + i) = i + 1;  
 \*(ptr + i) = rand() % (N \* N) + 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";  
  
 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 arr[3][3];  
 srand(time(0));  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 3; j++) {  
 arr[i][j] = rand() % 61 - 30;  
 }  
 }  
 cout << "Полученная матрица:" << endl;  
 for(int i=0; i<3; i++){  
 for(int j=0; j<3; j++){  
 cout << arr[i][j] << " ";  
 }  
 cout << endl;  
 }  
 int det = arr[0][0] \* arr[1][1] \* arr[2][2] - arr[0][0] \* arr[1][2] \* arr[2][1]  
 - arr[0][1] \* arr[1][0] \* arr[2][2] + arr[0][1] \* arr[1][2] \* arr[2][0]  
 + arr[0][2] \* arr[1][0] \* arr[2][1] - arr[0][2] \* arr[1][1] \* arr[2][0];  
 cout << "Определитель матрицы: " << det << endl;  
 break;  
 }  
 }  
 }  
 return 0;  
}