**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №2**

**по дисциплине «Программирование»**

Тема: Одномерные статические массивы.

|  |  |  |
| --- | --- | --- |
| Студентка гр. 3372 |  | Емельянова Ю.Б |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2023

**Цель работы.**

Разработать алгоритм и написать программу, работающую с с статичными одномерными массивами.

Приложение А

рабочий код

#include <iostream>

#include <ctime>

#include <chrono>

void fillArray(int arr[], int lenArray, int min, int max) {

srand(time(NULL));

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

arr[i] = rand() % (max - min + 1) + min;

}

}

void printArray(int arr[], int lenArray) {

std::cout << "[";

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

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

}

std::cout << "]\n";

}

void validInput(int &input) {

bool is\_valid;

do {

is\_valid = true;

std::cin >> input;

if (std::cin.fail()) {

std::cin.clear();

std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

std::cerr << "Неверный ввод. Пожалуйста, введите целое число еще раз." << std::endl;

is\_valid = false;

}

} while (!is\_valid);

}

void bubbleSort(int arr[], int lenArr) {

for (int i = 0; i < 100; i++) {

bool sorted = true;

for (int j = 0; j < 100 - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

std::swap(arr[j], arr[j + 1]);

sorted = false;

}

}

}

}

int main() {

setlocale(LC\_ALL, "ru");

std::cout <<

"Правктическая работа №2\n"

"ОДНОМЕРНЫЕ СТАТИЧЕСКИЕ МАССИВЫ\n\n\n"

"Целочисленный массив размера 100, заполненный случайными числами в диапазоне [-99; 99]:\n";

int arr[100];

int sortArr[100];

fillArray(arr, 100, -99, 99);

for (int i = 0; i < 100; i++) {

sortArr[i] = arr[i];

}

bubbleSort(sortArr, 100);

printArray(arr, 100);

std::cout <<

"ДЕЙСТВИЯ:\n"

"1 - Создать новый массив размера 100, заполненный случайными числами в диапазоне [-99; 99].\n"

"2 - Отсортировать массив в порядке возрастания\n"

"3 - Найти минимальный и максимальный элемент массива. Посчитать время поиска в отсортированном и неотсортированном массиве.\n"

"4 - Найти среднее значение максимального и минимального значения в отсортированном и неотсортированном. Вывести"

"индексы всех элементов, которые равны этому значению, и их количество.\n"

"5 - Вывести количество элементов в отсортированном массиве, которые меньше числа а\n"

"6 - Вывести количество элементов в отсортированном массиве, которые больше числа b\n"

"8 - Поменять местами элементы массива, индексы которых вводит пользователь.\n"

"9 - Уменьшить каждый нечетный элемент массива на введенное значение n. Каждый четный элемент умножить на случайную величину в диапазоне [1; 9]. "

"Вывести количество элементов, которое делится на 1, 2, 3, 4, 5, 6, 7, 8, 9.\n"

"0 - Завершить работу программы\n\n\n";

for (int stop = 0; stop < 1; stop) {

std::cout << "Введите номер действия: ";

int task;

validInput(task);

switch (task) {

case 1:

std::cout << "Новый целочисленный массив размера 100, заполненный случайными числами в диапазоне [-99; 99]:\n";

fillArray(arr, 100, -99, 99);

printArray(arr, 100);

for (int i = 0; i < 100; i++) {

sortArr[i] = arr[i];

}

bubbleSort(sortArr, 100);

break;

case 2: {

std::cout << "Отсортированный массив:\n";

int sortArr2[100];

for (int i = 0; i < 100; i++) {

sortArr2[i] = arr[i];

}

auto start = std::chrono::high\_resolution\_clock().now();

bubbleSort(sortArr2, 100);

auto finish = std::chrono::high\_resolution\_clock().now();

std::chrono::duration<float> duration = finish - start;

printArray(sortArr2, 100);

std::cout << "Время сортировки: " << duration.count() << " сек.\n\n";

break;

}

case 3: {

auto start1 = std::chrono::high\_resolution\_clock().now();

int min = arr[0];

int max = arr[0];

for (int i = 0; i < 100; i++) {

min = (min > arr[i] ? arr[i] : min);

max = (max < arr[i] ? arr[i] : max);

}

auto finish1 = std::chrono::high\_resolution\_clock().now();

std::chrono::duration<float> duration1 = finish1 - start1;

auto start2 = std::chrono::high\_resolution\_clock().now();

min = sortArr[0];

max = sortArr[99];

auto finish2 = std::chrono::high\_resolution\_clock().now();

std::chrono::duration<float> duration2 = finish2 - start2;

std::cout <<

"Минимальный элемент массива: " << min << std::endl <<

"Максимальный элемент массива: " << max << std::endl <<

"Время поиска минимального и максимального элемента в неотсортированном массиве: " << std::endl << duration1.count() << " сек.\n"

"Время поиска минимального и максимального элемента в отсортированном массиве массиве: " << duration2.count() << " сек.\n\n";

break;

}

case 4: {

int min = sortArr[0];

int max = sortArr[99];

float average = round((min + max) / 2.0);

int numbOfAverage = 0;

std::cout << "Среднее значение минимального и максимального элемента: " << average << "\n\n";

std::cout << "Индексы переменных, равных среднему значению: ";

auto start1 = std::chrono::high\_resolution\_clock().now();

for (int i = 0; i < 100; i++) {

bool doing = true;

if (sortArr[i] <= average) {

if (average == sortArr[i]) {

++numbOfAverage;

std::cout << i << " ";

}

}

else {

if (numbOfAverage == 0) {

std::cout << "нет\n\n";

}

std::cout << "\n";

doing = false;

}

if (!doing) {

break;

}

}

auto finish1 = std::chrono::high\_resolution\_clock().now();

std::chrono::duration<float> duration1 = finish1 - start1;

auto start2 = std::chrono::high\_resolution\_clock().now();

int numbOfAverage2 = 0;

for (int i = 0; i < 100; i++) {

if (arr[i] == average) {

++numbOfAverage;

}

}

auto finish2 = std::chrono::high\_resolution\_clock().now();

std::chrono::duration<float> duration2 = finish2 - start2;

std::cout << "Количество переменных, равных среднему значению: " << numbOfAverage << "\n";

std::cout << "Время поиска в отсортированном массиве (сек): " << duration1.count() << "\n";

std::cout << "Время поиска в неотсортированном массиве (сек): " << duration2.count() << "\n";

break;

}

case 5: {

int a;

std::cout << "Введите число a: ";

validInput(a);

int numbOfVarr = 0;

for (int i = 0; i < 100; i++) {

if (arr[i] < a) {

++numbOfVarr;

}

}

std::cout << "Количество элеменнтов в массиве, которые меньше " << a << ": " << numbOfVarr << "\n\n";

break;

}

case 6: {

int b;

std::cout << "Введите число b: ";

validInput(b);

int numbOfVarr = 0;

for (int i = 0; i < 100; i++) {

if (arr[i] < b) {

++numbOfVarr;

}

}

std::cout << "Количество элеменнтов в массиве, которые больше b " << b << ": " << numbOfVarr << "\n\n";

break;

}

case 8: {

int index1 = 100;

int index2 = 100;

while (index1 < 0 or index1 > 99) {

std::cout << "Введите первый индекс (от 0 до 99): ";

validInput(index1);

if (index1 < 0 or index1 > 99) {

std::cout << "Вы вышли за диапазон массива.";

}

}

while (index2 < 0 or index2 > 99) {

std::cout << "Введите второй индекс (от 0 до 99): ";

validInput(index2);

if (index2 < 0 or index2 > 99) {

std::cout << "Вы вышли за диапазон массива.";

}

}

std::swap(arr[index1], arr[index2]);

std::cout << "Вы поменяли местами элементы с индексом " << index1 << " и " << index2 << ":\n";

printArray(arr, 100);

break;

}

case 9:{

int n;

std::cout << "Введите целое число n: ";

validInput(n);

srand(time(NULL));

int randNumb = rand() % 9 + 1;

int multiple1 = 100;

int multiple2 = 0;

int multiple3 = 0;

int multiple4 = 0;

int multiple5 = 0;

int multiple6 = 0;

int multiple7 = 0;

int multiple8 = 0;

int multiple9 = 0;

for (int i = 0; i < 100; i++) {

if (i % 2 != 0) {

arr[i] -= n;

}

if (i%2 == 0) {

arr[i] \*= randNumb;

}

multiple2 = (arr[i]%2 == 0) ? ++multiple2: multiple2;

multiple3 = (arr[i] % 3 == 0) ? ++multiple3 : multiple3;

multiple4 = (arr[i] % 4 == 0) ? ++multiple4 : multiple4;

multiple5 = (arr[i] % 5 == 0) ? ++multiple5 : multiple5;

multiple6 = (arr[i] % 6 == 0) ? ++multiple6 : multiple6;

multiple7 = (arr[i] % 7 == 0) ? ++multiple7 : multiple7;

multiple8 = (arr[i] % 8 == 0) ? ++multiple8 : multiple8;

multiple9 = (arr[i] % 9 == 0) ? ++multiple9 : multiple9;

}

std::cout << "Нечетные элемены массива уменишины на n. Четные - умножены на случайное число от 1 до 9:\n";

printArray(arr, 100);

std::cout <<

"Количество элементов, которые делятся на 1: " << multiple1 << " \n"

"Количество элементов, которые делятся на 2: " << multiple2 << " \n"

"Количество элементов, которые делятся на 3: " << multiple3 << " \n"

"Количество элементов, которые делятся на 4: " << multiple4 << " \n"

"Количество элементов, которые делятся на 5: " << multiple5 << " \n"

"Количество элементов, которые делятся на 6: " << multiple6 << " \n"

"Количество элементов, которые делятся на 7: " << multiple7 << " \n"

"Количество элементов, которые делятся на 8: " << multiple8 << " \n"

"Количество элементов, которые делятся на 9: " << multiple9 << " \n";

std::cout << "\n";

}

default:

break;

}

}

return 0;

}