**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.

|  |  |  |
| --- | --- | --- |
| Студент гр. 3372 |  | Беляев К. В. |
| 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 <cmath>  
  
using namespace std;  
using namespace chrono;  
using time\_mngr = time\_point<steady\_clock, duration<\_\_int64\_t, ratio<1, 1000000000>>>;  
  
  
void clearStream(){  
 cin.sync();  
}  
  
  
void fillArray(int arr[], const int len){  
 for (int i = 0; i < len; ++i){  
 arr[i] = rand() % 200 - 99;  
 }  
}  
  
  
int findPivot(int arr[], int min, int max) {  
 int pivot = min -1;  
 for (int i = min; i < max; ++i){  
 if(arr[i] < arr[max]) {  
 pivot++;  
 swap(arr[i], arr[pivot]);  
 }  
 }  
 pivot++;  
 swap(arr[pivot], arr[max]);  
 return pivot;  
}  
  
  
void quickSort(int arr[], int min, int max) {  
 if (min >= max) {  
 return;  
 }  
 int pivot = findPivot(arr, min, max);  
 quickSort(arr, min, pivot - 1);  
 quickSort(arr, pivot + 1, max);  
}  
  
  
int binSearch(int arr[], int value, const int len){  
 int low = 0;  
 int high = len - 1;  
 int mid;  
 while (low <= high){  
 mid = (low + high) / 2;  
 if (value == arr[mid]){  
 return mid;  
 }  
 else if (value > arr[mid]){  
 low = mid + 1;  
 }  
 else{  
 high = mid - 1;  
 }  
 }  
 return -1;  
}  
  
  
void printArray(int arr[], const int len){  
 cout << "{" << arr[0];  
 for (int i = 1; i < len; ++i){  
 cout << ", " << arr[i];  
 }  
 cout << "}";  
}  
  
  
  
  
  
void findExtremums(int arr[], const int len){  
  
 int mx = arr[0];  
 int mn = arr[0];  
  
 for (int i = 1; i < len; ++i){  
 mx = max(mx, arr[i]);  
 mn = min(mn, arr[i]);  
 }  
}  
  
  
int findRight(int arr[], int value, int low, int mid, int high){  
 while ((high - low) > 1){  
 mid = (low + high) / 2;  
 if (value >= arr[mid]){  
 low = mid;  
 }else{  
 high = mid;  
 }  
 }  
 return low;  
}  
  
  
int findLeft(int arr[], int value, int low, int mid, int high){  
 while ((high - low) > 1){  
 mid = (low + high) / 2;  
 if (value <= arr[mid]){  
 high = mid;  
 }else{  
 low = mid;  
 }  
 }  
 return high;  
}  
  
  
int findElement(int arr[], int len, int value){  
 for (int i = 0; i < len; ++i){  
 if (value == arr[i]){  
 return 1;  
 }  
 }  
 return 0;  
  
}  
  
  
void idz9(int arr[], int len, int num) {  
 for (int i = 1; i < len; i += 2) {  
 if (i % 2) {  
 arr[i] -= num;  
 arr[i] \*= rand() % 9 + 1;  
 }  
 }  
 for (int j = 1; j < 10; ++j) {  
 cout << "Элементы, кратные " << j << ": ";  
 for (int i = 0; i < len; ++i) {  
 if (!(arr[i] % j)){  
 cout << arr[i] << " ";  
 }  
 }  
 cout << "\n";  
 }  
  
}  
  
  
int main(){  
 setlocale(LC\_ALL, "ru");  
  
 const int n = 100;  
 int arr[n];  
 int problem, mx, mn;  
 int a, b;  
 int avg;  
 time\_mngr start, end;  
 nanoseconds result;  
 int low, mid, high, l\_border, r\_border;  
 int index;  
 int index1, index2;  
 int idz\_num;  
  
  
 for (int i = 0; i < 10; ++i) {  
 if (!i){  
 cout << "Задание 1\n";  
 cout << "Заполненный массив\n";  
 fillArray(arr, n);  
 printArray(arr, n);  
 cout << "\n";  
 };  
  
 cout << "\nВведите номер задачи: ";  
 cin >> problem;  
 clearStream();  
 switch (problem) {  
  
 case 1:  
 fillArray(arr, n);  
 printArray(arr, n);  
 cout << "\n";  
 break;  
  
 case 2:  
 start = steady\_clock::now();  
 quickSort(arr, 0, n - 1);  
 end = steady\_clock::now();  
 result = end - start;  
 printArray(arr, n);  
 cout << "\n";  
 cout << "Сортировка прошла за " << result.count() << " наносек.";  
 cout << "\n";  
 break;  
  
 case 3:  
 fillArray(arr, n);  
 start = steady\_clock::now();  
 findExtremums(arr, n);  
 end = steady\_clock::now();  
 result = end - start;  
 cout << "Поиск max и min значений несортированного массива прошел за " << result.count() << " наносек.\n";  
  
 quickSort(arr, 0,n - 1);  
 start = steady\_clock::now();  
 mx = arr[n -1];  
 mn = arr[0];  
 end = steady\_clock::now();  
 result = end - start;  
 cout << "Поиск max и min значений отсортированного массива прошел за " << result.count() << " наносек.\n";  
 cout << "min: " << mn << ", max: " << mx;  
 cout << "\n";  
 break;  
  
 case 4:  
 avg = round((mx + mn) / 2.0);  
 printArray(arr, n);  
 cout << "\nmx: " << mx << ", mn: " << mn;  
 cout <<"\navg: " << avg << "\n";  
 if (binSearch(arr, avg, n) == -1) {  
 cout << "Такого элемента нет в массиве";  
 } else {  
 high = n - 1;  
 l\_border = findLeft(arr, avg, low, mid, high);  
 r\_border = findRight(arr, avg, low, mid, high);  
 cout << "В массиве " << r\_border - l\_border + 1 << " элементов " << avg << "\n";  
 cout << "Индексы элементов: ";  
 for (int i = l\_border; i <= r\_border; ++i){  
 cout << i << " ";  
 }  
 }  
 cout << "\n";  
 break;  
  
 case 5:  
 cout << "Введите число a: ";  
 cin >> a;  
 clearStream();  
 low = 0;  
 high = n - 1;  
 l\_border = findLeft(arr, a, low, mid, high);  
 cout << l\_border << " элементов, меньших " << a << "\n";  
 break;  
  
 case 6:  
 cout << "Введите число b: ";  
 cin >> b;  
 clearStream();  
 low = 0;  
 high = n - 1;  
 r\_border = findRight(arr, b, low, mid, high);  
 cout << n - 1 - r\_border << " элементов, больших " << b << "\n";  
 break;  
  
 case 7:  
 cout << "Введите число a: ";  
 cin >> a;  
 clearStream();  
 fillArray(arr, n);  
 quickSort(arr, 0, n - 1);  
 printArray(arr, n);  
 start = steady\_clock::now();  
 findElement(arr, n, a);  
 end = steady\_clock::now();  
 result = end - start;  
 cout << "Обычный перебор прошел за " << result.count() << " наносек\n";  
  
 start = steady\_clock::now();  
 index = binSearch(arr, a, n);  
 end = steady\_clock::now();  
 result = end - start;  
 if (index == -1){  
 cout << "Числа " << a << " нет в массиве";  
 }else{  
 cout << "Число " << arr[index] << " Находится под индексом " << index;  
 }  
 cout << "\nБинпоиск прошел за " << result.count() << " наносек\n";  
 break;  
  
 case 8:  
 cout << "Вводите индексы элементов, которые хотите поменять местами (Вводить попарно и через пробел, чтобы закончить введите: -1 -1):\n";  
 cin >> index1 >> index2;  
 clearStream();  
 while (index1 != -1 && index2 != -1){  
 start = steady\_clock::now();  
 swap(arr[index1], arr[index2]);  
 end = steady\_clock::now();  
 cout << "Свап прошел за " << (end - start).count() << " наносек\n";  
 cin >> index1 >> index2;  
 clearStream();  
 }  
 printArray(arr, n);  
 break;  
  
 case 9:  
 cout << "Введите число: ";  
 cin >> idz\_num;  
 clearStream();  
 idz9(arr, n, idz\_num);  
 break;  
  
  
 }  
  
 }  
 return 0;  
}