**Пермский филиал федерального государственного автономного образовательного учреждения высшего**

**“Национальный исследовательский университет**

**“Высшая школа экономики”**

Факультет экономики, менеджмента и бизнес-информатики

Дмитриев Арсений Алексеевич

**Лабораторная работа №4. «Работа с одномерными массивами»**

Отчет по практической работе

**студента образовательной программы «Программная инженерия»**

**по направлению подготовки *09.03.04 Программная инженерия***

**руководитель**

**к. тех. н., доцент каф.**

**инф. техн. в биз.**

**Викентьева О.Л.**

**Пермь, 2019 г.**

1. **Постановка задачи:**

**Общая**

1) Сформировать массив из n элементов с помощью датчика случайных чисел (n задается

пользователем с клавиатуры).

2) Распечатать массив.

3) Выполнить удаление указанных элементов из массива.

4) Выполнить добавление указанных элементов в массив.

5) Выполнить перестановку элементов в массиве.

6) Выполнить поиск указанных в массиве элементов и подсчитать количество сравнений, необходимых для поиска нужного элемента.

7) Выполнить сортировку массива указанным методом.

8) Выполнить поиск указанных элементов в отсортированном массиве и подсчитать количество сравнений, необходимых для поиска нужного элемента.

**Вариант №7**

1) Сформировать массив из n элементов с помощью датчика случайных чисел (n задается пользователем с клавиатуры).

2) Распечатать массив.

3) Выполнить удаление нечетных элементов из массива.

4) Выполнить добавление N элементов, начиная с номера K.

5) Выполнить перестановку элементов: положительные элементы переставить в начало, отрицательные – в конец.

6) Выполнить поиск элемента с заданным ключом (значением) в массиве и подсчитать количество сравнений, необходимых для поиска нужного элемента.

7) Выполнить сортировку массива методом простых обменов.

8) Выполнить поиск элемента с заданным ключом (значением)в отсортированном массиве и подсчитать количество сравнений, необходимых для поиска нужного элемента.

1. **Анализ задач:**

Распределение задач по классам:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Функция | Класс | Пояснение |
| 1 | Печать массива | 1 | Однотипная обработка элементов (печать) 1 проходом |
|
| 2 | Удаление нечетных элементов | 3 | Асинхронная обработка 2х массивов (старого и нового) |
|
| 3 | Добавление новых элементов | 3 | Асинхронная обработка 2х массивов (старого и нового) |
|
| 4 | Перестановка элементов | 2 | Обмены элементов массива |
|
| 5 | Поиск в несортированном массиве | 4 | Задача поиска |
|
| 6 | Сортировка массива | 2 | Обмены элементов массива |
|
| 7 | Поиск в сортированном массиве | 4 | Задача поиска |
|

Классы входных и выходных данных:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Функция | Входные данные | Классы входных данных | Выходные данные |
| Основные функции | | | | |
| 1.1 | Формирование массива с помощью ДСЧ | * Количество элементов в массиве | Целое положительное число (>0) | Массив |
| Целое неположительное (<=0) | Сообщение об ошибке |
| Вещественное число |
| Нечисло |
| 1.2 | Формирование массива с клавиатуры | * Количество элементов в массиве | Целое положительное число (>0) | Массив |
| Целое неположительное (<=0) | Сообщение об ошибке |
| Вещественное число |
| Нечисло |
| * Число | Целое число | Массив |
| Не целое число или нечисло | Сообщение об ошибке |
| 2 | Печать массива | * Массив * Размер массива | Не пустой массив | Печать массива |
| Пустой массив | Сообщение об ошибке |
| 3 | Удаление нечетных элементов из массива | * Массив * Размер массива | Не пустой массив | Измененный массив |
| Пустой массив | Сообщение об ошибке |
| 4 | Добавление новых элементов в массив | * Массив * Номер элемента,   с которого добавляем   * Количество добавляемых элементов | * Не пустой массив * Номер (1 <= K <= размер массива) * Количество >0 | Измененный массив |
|
|
|
|
| * Номер, с которого добавляем (K<=0) или (K > размера массива) * или Номер, с которого добавляем - нечисло * или Количество добавляемых - нечисло * или Количество добавляемых - неположительное число (<=0) | * Сообщение об ошибке * Массив без изменений |
|
|
|
|
|
|
|
| 5 | Перестановка элементов | * Массив * Размер массива | Не пустой массив | Измененный массив |
| Пустой массив | Сообщение об ошибке |
| 6 | Поиск элемента в неотсортированном массиве | * Массив * Размер массива * Искомый элемент | * Не пустой массив * Искомый элемент есть в массиве | * Индекс искомого элемента * Количество потребовавшихся итераций |
|
|
|
|
| * Не пустой массив * Искомого элемента в массиве нет | Сообщение об отсутствии элемента |
|
|
|
|
| Пустой массив | Сообщение об ошибке |
| Искомый элемент - нечисло |
| 7 | Сортировка массива | * Массив * Размер массива | * Не пустой массив * Массив не отсортирован | Отсортированный массив |
|
| * Не пустой массив * Массив отсортирован | * Сообщение о сортированности * Массив без изменений |
|
|
| Пустой массив | Сообщение об ошибке |
|
| 8 | Поиск в отсортированном массиве | * Массив * Размер массива * Искомый элемент | * Не пустой массив * Искомый элемент есть в массиве | * Индекс искомого элемента * Количество потребовавшихся итераций |
|
|
|
|
| * Не пустой массив * Искомого элемента в массиве нет | Сообщение об отсутствии элемента |
|
|
|
|
| Пустой массив | Сообщение об ошибке |
| Искомый элемент - нечисло |
| Массив не отсортирован |
|
| Вспомогательные функции | | | | |
| 9 | Ввод целого числа | * Строка, введенная пользователем * Нижняя граница допустимого диапазона * Верхняя граница диапазона | Введенная строка – число, попадающее в допустимый диапазон от нижней до верхней границы | Целое число |
|
|
|
| Строка - не целое число | Сообщение об ошибке |
| Строка - нечисло |
| Строка - целое число, не попадающее в допустимый диапазон |
|
|
| Строка - целое число, >2^31-1 или < -2^31 |
|
| 10 | Проверка массива на сортированность | * Массив * Размер массива | Массив отсортирован | True |
|
| Массив не отсортирован | False |
|
| Пустой массив | Сообщение об ошибке |
| 11 | Вывод меню для пользователя |  |  |  |
|
|

1. **Проектирование:**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Функция | Входные данные | Выходные данные |
| Основные функции | | | |
| 1 | ReadArray  (формирование массива с клавиатуры) | int sizeOfArray - размер массива | int[] arr - массив Сообщение об ошибке |
|
|
|
| 2 | CreateRandomArray  (формирование массива с помощь ДСЧ) | int sizeOfArray - размер массива | int[] arr - массив |
|
|
|
| 3 | ArrayPrint  (Печать массива) | int[] arr - массив | Печать массива |
|
|
|
| 4 | ArrayRemoveOddElements  (Удаление нечетных элементов) | int[] arr - массив | int[] newArr - измененный массив |
|
|
|
| 5 | ArrayAddNewElements  (Добавление элементов в массив) | int[] arr - массив int pos - позиция,с которой добавляются элементы | int[] newArr - измененный массив |
|
|
|
| 6 | ArrayPosNegShuffle  (Перестановка элементов массива) | int[] arr - массив | int[] newArr - измененный массив |
|
|
|
| 7 | ArraySearchNoSort  (Поиск элемента в несортированном массиве) | int[] arr - массив int key - искомый элемент | int pos - позиция искомого элемента int it - количество итераций |
|
|
|
| 8 | ArraySort  (Сортировка массива) | int[] arr - массив | int[] newArr - измененный массив |
|
|
|
| 9 | ArrayBinarySearch  (Поиск в отсортированном массиве) | int[] arr - массив int key - искомый элемент | int pos - позиция искомого элемента int it - количество итераций |
|
|
|
| Вспомогательные функции | | | |
| 10 | ReadInteger  (Ввод целого числа) | int lowerBound - нижняя граница диапазона int upperBound - верхняя граница диапазона | int number - целое число |
|
|
|
| 11 | IsArraySorted  (Проверка массива на сортированность) | int[] arr - массив | bool sorted |
|
|
|
| 12 | MainMenu  (Вывод меню для пользователя) |  |  |
|
|
|

1. **Алгоритм:**

Основа:





1. Формирование массива:
2. Печать массива:

1. Удаление элементов 4) Добавление элементов



5) Перестановка 6) Поиск по ключу в несортированном





7) Сортировка простым обменом:

8) поиск в сортированном массиве (бинпоиск)



1. **Программа**

using System;

namespace Lab4

{

class Program

{

static int ReadInteger(string userInstruction, int lowerBound, int upperBound)

{

int number = 0;

bool ok = false;

do

{

try

{

Console.Write(userInstruction);

string input = Console.ReadLine();

number = int.Parse(input);

if (number >= lowerBound && number <= upperBound) ok = true;

else

{

Console.WriteLine($"Ошибка: введите число из промежутка от {lowerBound} до {upperBound}!");

}

}

catch (FormatException)

{

Console.WriteLine("Ошибка: введите целое число!");

ok = false;

}

catch (OverflowException)

{

Console.WriteLine("Ошибка: введено слишком большое (маленькое) число!");

ok = false;

}

} while (!ok);

return number;

}

static int[] ReadArray(int sizeOfArray)

{

int[] arr = new int[sizeOfArray];

for (int i = 0; i < sizeOfArray; i++) arr[i] = ReadInteger($"Введите значение {i+1} элемента: ", -10000, 10000);

return arr;

}

static int[] CreateRandomArray(int sizeOfArray)

{

int[] arr = new int[sizeOfArray];

Random rand = new Random();

for (int i = 0; i < sizeOfArray; i++) arr[i] = rand.Next(20000) - 10000;

return arr;

}

static int[] CreateNewArray(string userInstruction)

{

Console.WriteLine(userInstruction);

int sizeOfArray = ReadInteger("Введите размер массива: ", 1, 1000);

Console.WriteLine("Выберите как должен быть сформирован массив:");

Console.WriteLine(" 1: Датчиком случайных чисел");

Console.WriteLine(" 2: Ввод с клавиатуры");

int userChoise = ReadInteger("Нужная опция: ", 1, 2);

int[] arr = null;

switch (userChoise)

{

case 1:

{

arr = CreateRandomArray(sizeOfArray);

break;

}

case 2:

{

arr = ReadArray(sizeOfArray);

break;

}

}

return arr;

}

static bool EmptyArraySituation(string cause, out int[] arr)

{

Console.WriteLine($"После {cause} массив стал пустым");

Console.WriteLine("Выберите вариант продолжения работы:");

Console.WriteLine("1: Сформировать новый массив");

Console.WriteLine("0: Прекратить работу программы");

bool ok = false;

arr = null;

int userChoise = ReadInteger("Ваш выбор: ", 0, 1);

if(userChoise == 1)

{

arr = CreateNewArray("Формирование нового массива:");

ok = true;

}

return !ok;

}

static void MainMenu()

{

Console.WriteLine();

Console.WriteLine("Меню доступных опций:");

Console.WriteLine(" 1: Сформировать новый массив");

Console.WriteLine(" 2: Напечатать массив");

Console.WriteLine(" 3: Удалить все нечетные элементы");

Console.WriteLine(" 4: Добавить новые элементы в массив");

Console.WriteLine(" 5: Переставить положительные элементы в начало, отрицательные - в конец");

Console.WriteLine(" 6: Выполнить поиск элемента в несортированном массиве");

Console.WriteLine(" 7: Отсортировать массив по возрастанию");

Console.WriteLine(" 8: Выполнить поиск элемента в отсортированном массиве");

Console.WriteLine(" 0: Прекратить работу программы");

}

static void ArrayPrintUser(int[] arr)

{

Console.WriteLine("Печать массива:");

int sizeOfArray = arr.Length;

Console.WriteLine($"Размер массива : {sizeOfArray}");

Console.Write("Элементы массива: ");

for (int i = 0; i < sizeOfArray; i++) Console.Write($"{arr[i]} ");

Console.WriteLine();

}

static void ArrayPrintFunc(int[] arr)

{

if (arr.Length == 0) Console.WriteLine("Пустой массив");

for (int i = 0; i < arr.Length; i++) Console.Write(arr[i] + " ");

Console.WriteLine();

}

static int[] ArrayRemoveOddElements(int[] arr)

{

Console.WriteLine("Удаление нечетных элементов:");

int sizeOfArray = arr.Length;

int sizeNewArray = 0;

for (int i = 0; i < sizeOfArray; i++) if (arr[i] % 2 == 0) sizeNewArray++;

int[] newArr = new int[sizeNewArray];

int k = 0;

for (int i = 0; i < sizeOfArray && sizeNewArray > 0; i++)

{

if (arr[i] % 2 == 0)

{

newArr[k] = arr[i];

k++;

sizeNewArray--;

}

}

ArrayPrintFunc(newArr);

return newArr;

}

static int[] ArrayAddNewElements(int[] arr, int pos)

{

Console.WriteLine("Добавление элементов:");

int[] addElem = CreateNewArray("Сформируем массив добавляемых элементов:");

int sizeNewArray = arr.Length + addElem.Length;

int[] newArr = new int[sizeNewArray];

int arrInd = 0;

int addInd = 0;

for (int i = 0; i < pos; i++, arrInd++) newArr[i] = arr[i];

for (int i = pos; i < pos + addElem.Length; i++, addInd++) newArr[i] = addElem[addInd];

for (int i = pos + addElem.Length; i < sizeNewArray; i++, arrInd++) newArr[i] = arr[arrInd];

ArrayPrintFunc(newArr);

return newArr;

}

static void ArrayPosNegShuffle(ref int[] arr)

{

Console.WriteLine("Перестановка элементов");

int[] addArr = new int[arr.Length];

int posI = 0, negI = arr.Length - 1;

for(int i = 0; i<arr.Length; i++)

{

if(arr[i] < 0)

{

addArr[negI] = arr[i];

negI--;

} else

{

addArr[posI] = arr[i];

posI++;

}

}

arr = addArr;

ArrayPrintFunc(arr);

}

static bool ArraySearchNoSort(int[] arr, int key, out int pos, out int it)

{

Console.WriteLine("Поиск элемента в несортированном массиве:");

bool find = false;

pos = -1; it = 0;

for(int i = 0; i < arr.Length && !find; i++)

{

if(arr[i] == key)

{

find = true;

pos = i+1;

}

it++;

}

ArrayPrintFunc(arr);

return find;

}

static void ArraySort(ref int[] arr)

{

Console.WriteLine("Сортировка массива");

int sizeOfArray = arr.Length;

for(int i = 1; i<sizeOfArray; i++)

{

for(int j = sizeOfArray-1; j>=i; j--)

{

if(arr[j] < arr[j - 1])

{

int tmp = arr[j];

arr[j] = arr[j - 1];

arr[j - 1] = tmp;

}

}

}

ArrayPrintFunc(arr);

}

static bool IsArraySorted(int[] arr)

{

for(int i = 0; i<arr.Length-1; i++)

{

if (arr[i + 1] < arr[i]) return false;

}

return true;

}

static bool BinarySearch(int[] arr, int key, out int pos, out int it)

{

Console.WriteLine("Поиск в сортированном массиве");

int left = 0, right = arr.Length - 1, mid;

it = 0; pos = -1;

do

{

mid = (left + right) / 2;

if (arr[mid] < key) left = mid + 1;

else right = mid;

it++;

} while (left != right);

ArrayPrintFunc(arr);

if (arr[left] == key)

{

pos = left+1;

return true;

}

else return false;

}

static void Main(string[] args)

{

Console.WriteLine("Программа для работы с массивом размером до 1000 и элементами не превышающими по модулю 10000");

int[] arr = CreateNewArray("Для начала сформируем массив:");

int sizeOfArray = arr.Length;

bool finish = false;

do

{

MainMenu();

int userChoise = ReadInteger("Выбранная опция: ", 0, 8);

Console.WriteLine();

switch (userChoise)

{

case 0:

{

Console.WriteLine("Завершаю работу");

finish = true;

break;

}

case 1:

{

arr = CreateNewArray("Формирование нового массива:");

break;

}

case 2:

{

ArrayPrintUser(arr);

break;

}

case 3:

{

arr = ArrayRemoveOddElements(arr);

if (arr.Length == 0)

{

finish = EmptyArraySituation("удаления нечетных элементов", out arr);

}

break;

}

case 4:

{

int addPos = ReadInteger("Введите позицию с которой нужно добавить новые элементы: ", 1, sizeOfArray);

arr = ArrayAddNewElements(arr, addPos - 1); //Позицию уменьшаем на 1, тк пользователь считает от 1

break;

}

case 5:

{

ArrayPosNegShuffle(ref arr);

break;

}

case 6: {

int key = ReadInteger("Введите искомый элемент: ", -10000, 10000);

int posOfKey, iterNum;

bool find = ArraySearchNoSort(arr, key, out posOfKey, out iterNum);

if (find)

{

Console.WriteLine($"Искомый элемент находится в массиве на позиции: {posOfKey}");

Console.WriteLine($"Для нахождения потребовалось итераций: {iterNum}");

}

else Console.WriteLine("Искомый элемент не найден");

break;

}

case 7:

{

if (IsArraySorted(arr)) Console.WriteLine("Массив уже отсортирован");

else ArraySort(ref arr);

break;

}

case 8:

{

bool ok = IsArraySorted(arr);

if(!ok)

{

Console.WriteLine("Поиск невозможен, так как массив не отсортирован");

Console.WriteLine("Отсортировать массив? (1 - Да, 0 - Нет): ");

int toSort = ReadInteger("", 0, 1);

if (toSort == 1) ArraySort(ref arr);

else break;

}

int key = ReadInteger("Введите искомый элемент: ", -10000, 10000);

int posOfKey, iterNum;

bool find = BinarySearch(arr, key, out posOfKey, out iterNum);

if (find)

{

Console.WriteLine($"Искомый элемент находится в массиве на позиции: {posOfKey}");

Console.WriteLine($"Для нахождения потребовалось итераций: {iterNum}");

}

else Console.WriteLine("Искомый элемент не найден");

break;

}

}

} while (!finish);

}

}

}

1. **Тестирование**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерии\тесты | Т1 | Т2 | Т3 | Т4 |
| Критерии входных данных: размер массива | | | | |
| Целое число >=1 | + | + | + | + |
| Целое число <1 |  | + |  |  |
| Вещественное число |  |  | + |  |
| Нечисло | + |  |  |  |
| Критерии входных данных: элементы массива | | | | |
| Целое число <=|10000| | + | + | + | + |
| Целое число >|10000| и <|2^31-1| |  | + |  |  |
| Вещественное число |  | + |  |  |
| Нечисло |  |  | + |  |
| Критерии длины набора | | | | |
| Единичный набор |  |  | + |  |
| Длина >1 | + | + |  | + |
| Критерии выходных данных | | | | |
| Массив | + | + | + | + |
| Сообщение об ошибке | + | + | + |  |

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ввод | Ожидаемый результат | Результат работы |
| Т1 | а 2 15 3 | Сообщение об ошибке  Массив | Сообщение об ошибке  Массив |
| Т2 | -4 3 1,5 11 2^32 11 5 | Сообщение об ошибке  Массив | Сообщение об ошибке  Массив |
| Т3 | 100,1 1 ft 3 | Сообщение об ошибке  Массив | Сообщение об ошибке  Массив |
| Т4 | 4 2 12 -8 13 | Массив | Массив |

1. Формирование массива
2. Печать массива

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ввод | Ожидаемый результат | Результат работы |
| Т1 | 4 2 12 -8 13 | 2 12 -8 13 | 2 12 -8 13 |

|  |  |
| --- | --- |
| Критерии\тесты | Т1 |
| Критерии входных данных: массив | |
| Не пустой | + |
| Критерии выходных данных: | |
| Напечатанный массив | + |

1. Удаление элементов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ввод | Ожидаемый результат | Результат работы |
| Т1 | 4 2 12 -8 13 | 2 12 -8 | 2 12 -8 |
| Т2 | 3 -10 32 14 | -10 32 14 | -10 32 14 |
| Т3 | 5 1 3 5 7 9 | Пустой массив | Пустой массив |

|  |  |  |  |
| --- | --- | --- | --- |
| Критерии\тесты | Т1 | Т2 | Т3 |
| Критерии входных данных: массив | | | |
| Не пустой | + | + | + |
| Критерии длины набора: нечетные элементы | | | |
| Нечетных =0 |  | + |  |
| Нечетных >=1 | + |  |  |
| Нечетных = размер массива |  |  | + |
| Критерии выходных данных | | | |
| Массив без изменений |  | + |  |
| Измененный массив | + |  |  |
| "Массив стал пустым" |  |  | + |

1. Добавление элементов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ввод | Ожидаемый результат | Результат работы |
| Т1 | f 1 d 3 1 2 3 | Сообщение об ошибке Измененный массив | Сообщение об ошибке Измененный массив |
| Т2 | 1,5 4 -2 в 1 15 | Сообщение об ошибке Измененный массив | Сообщение об ошибке Измененный массив |
| Т3 | -100 6 1,1 1 12 | Сообщение об ошибке Измененный массив | Сообщение об ошибке Измененный массив |
| Т4 | 3 5 1 3 5 7 9 | Измененный массив | Измененный массив |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерии\тесты | Т1 | Т2 | Т3 | Т4 |
| Вход: позиция, с которой добавляем элементы | | | | |
| Целое число >=1 и <=размер | + | + | + | + |
| Целое число <1 или >размер |  |  | + |  |
| Вещестевенное число |  | + |  |  |
| Нечисло | + |  |  |  |
| Критерии входных данных: количество добавляемых элементов | | | | |
| Целое число >=1 и <= 1000 | + | + | + | + |
| Целое число <1 или >1000 |  | + |  |  |
| Вещестевенное число |  |  | + |  |
| Нечисло | + |  |  |  |
| Критерии области допустимых значений: позиция добавления | | | | |
| В начало массива (=1) | + |  |  |  |
| В середину массива |  | + |  | + |
| В конец массива (=размер) |  |  | + |  |
| Критерии выходных данных | | | | |
| Измененный массив | + | + | + | + |
| Сообщение об ошибке | + | + | + |  |

1. Перестановка

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ввод | Ожидаемый результат | Результат работы |
| Т1 | 2 -23 -8 13 | 2 13 -8 -23 | 2 13 -8 -23 |

|  |  |
| --- | --- |
| Критерии\тесты | Т1 |
| Критерии входных данных: массив | |
| Не пустой массив | + |
| Классы выходных данных | |
| Измененный массив | + |

1. Поиск по ключу в несортированном массиве

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ввод | Ожидаемый результат | Результат работы |
| Т1 | ап 12| 12 1 3 5 | Сообщение об ошибке Позиция - 1  Итераций – 1 | Сообщение об ошибке Позиция – 1  Итераций – 1 |
| Т2 | 1,8 1| 4 9 22 13 7 1 | Сообщение об ошибке Позиция – 6  Итераций -6 | Сообщение об ошибке Позиция - 6  Итераций -6 |
| Т3 | 10001 18| 15 2 14 8 | Сообщение об ошибке Сообщение о ненахождении | Сообщение об ошибке Сообщение о ненахождении |
| Т4 | 40| 5 8 9 40 50 | Позиция - 4  Итераций - 4 | Позиция - 4  Итераций - 4 |
| Т5 | 12| 1 5 12 3 5 12 | Позиция - 3  Итераций - 3 | Позиция - 3  Итераций - 3 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерии\тесты | Т1 | Т2 | Т3 | Т4 | Т5 |
| Критерии входных данных: искомое значение | | | | | |
| Целое число <=|10000| | + | + | + | + | + |
| Целое число >|10000| |  |  | + |  |  |
| Вещественное число |  | + |  |  |  |
| Нечисло | + |  |  |  |  |
| Критерии входных данных: массив | | | | | |
| В массиве есть искомый элемент | + | + |  | + | + |
| В массиве нет искомого элемента |  |  | + |  |  |
| Критерии задачи поиска: искомый элемент | | | | | |
| Встречается =1 раз | + | + |  | + |  |
| Встречается >1 раза |  |  |  |  | + |
| В начале массива | + |  |  |  |  |
| В середине массива |  |  |  | + |  |
| В конце массива |  | + |  |  |  |
| Критерии выходных данных | | | | | |
| Позиция искомого элемента | + | + |  | + | + |
| Количество итераций | + | + |  | + | + |
| Сообщение о ненахождении |  |  | + |  |  |
| Сообщение об ошибке | + | + | + |  |  |

1. Сортировка

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ввод | Ожидаемый результат | Результат работы |
| Т1 | 2 -1 8 10 -9 | -9 -1 2 8 10 | -9 -1 2 8 10 |
| Т2 | 1 2 3 4 5 6 | 1 2 3 4 5 6 | 1 2 3 4 5 6 |
| Т3 | 10 8 2 0 -10 | -10 0 2 8 10 | -10 0 2 8 10 |

|  |  |  |  |
| --- | --- | --- | --- |
| Критерии\тесты | Т1 | Т2 | Т3 |
| Критерии упорядоченности | | | |
| Массив неупорядочен | + |  |  |
| Массив упорядочен в прямом порядке |  | + |  |
| Массив упорядочен в обратном порядке |  |  | + |
| Критерии выходных данных | | | |
| Массив, упорядоченнный в прямом порядке | + | + | + |

1. Бинарный поиск

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ввод | Ожидаемый результат | Результат работы |
| Т1 | ап 1| 1 2 3 5 | Сообщение об ошибке Позиция - 1  Итераций - 2 | Сообщение об ошибке Позиция - 1  Итераций - 2 |
| Т2 | 1,8 1| 4 9 22 13 7 1 | Сообщение об ошибке | Сообщение об ошибке |
| Т3 | 10001 18| 2 8 14 15 | Сообщение об ошибке Сообщение о ненахождении | Сообщение об ошибке Сообщение о ненахождении |
| Т4 | 40| 5 8 9 40 50 | Позиция - 4  Итераций - 2 | Позиция - 4  Итераций - 2 |
| Т5 | 12| 1 5 12 12 23 71 | Позиция - 3  Итераций - 2 | Позиция - 3  Итераций - 2 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерии\тесты | Т1 | Т2 | Т3 | Т4 | Т5 |
| Критерии входных данных: массив | | | | | |
| В массиве есть искомый элемент | + | + |  | + | + |
| В массиве нет искомого элемента |  |  | + |  |  |
| Критерии входных данных: искомое значение | | | | | |
| Целое число <=|10000| | + | + | + | + | + |
| Целое число >|10000| |  |  | + |  |  |
| Вещественное число |  | + |  |  |  |
| Нечисло | + |  |  |  |  |
| Критерии упорядоченности | | | | | |
| Массив упорядочен в прямом порядке | + |  | + | + | + |
| Массив неупорядочен |  | + |  |  |  |
| Критерии задачи поиска: искомый элемент | | | | | |
| Встречается =1 раз | + | + |  | + |  |
| Встречается >1 раза |  |  |  |  | + |
| В начале массива | + |  |  |  |  |
| В середине массива |  |  |  | + |  |
| В конце массива |  | + |  |  |  |
| Критерии выходных данных | | | | | |
| Позиция искомого элемента | + |  |  | + | + |
| Количество итераций | + |  |  | + | + |
| Сообщение о ненахождении |  |  | + |  |  |
| Сообщение об ошибке | + | + | + |  |  |