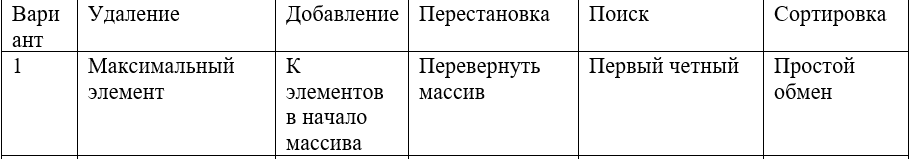
|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет экономики, менеджмента и бизнес-информатики* |
|  |
| Полежаев Яков Вячеславович  **ЛАБОРАТОРНАЯ РАБОТА №4**  по направлению подготовки *38.03.05 Бизнес-информатика*  образовательная программа «Бизнес-информатика»   |  |  | | --- | --- | |  | Руководитель  Преподаватель кафедры ИТБ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Марквирер В.Д. |   Пермь, 2022 |

**Задача №1**

**1. Постановка задачи**

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.



**2. Анализ**

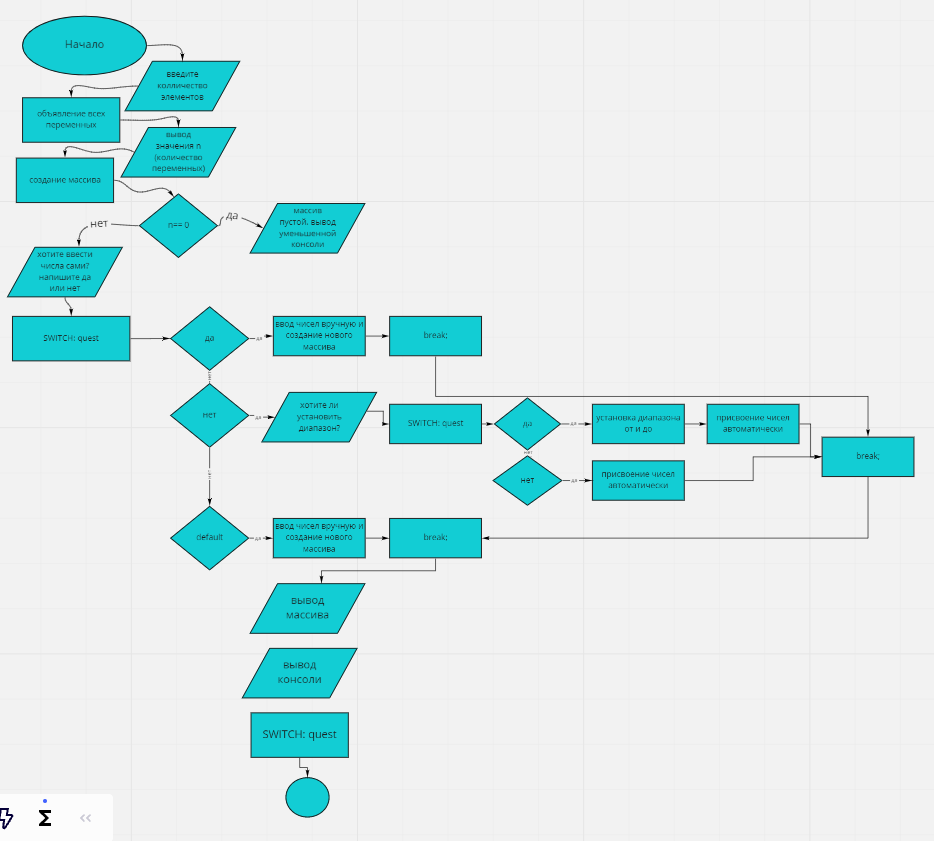
Исходные данные:

* n – натуральное положительное число(int).
* a (он же nums[i]) – натуральное число (int)

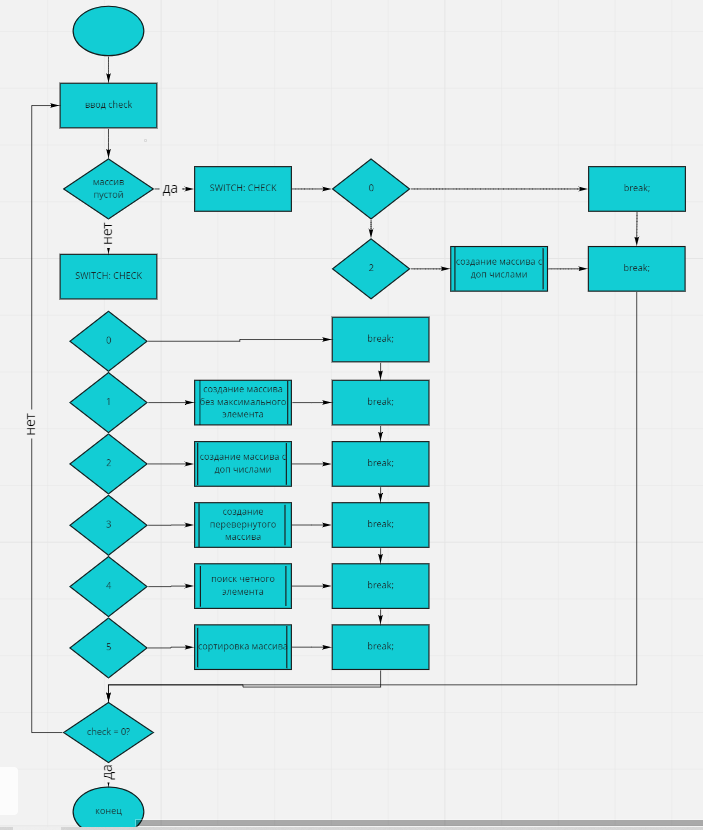
Выходные данные:

* nums – массив (int)
* ***Табл. 1. Анализ классов исходных данных.***

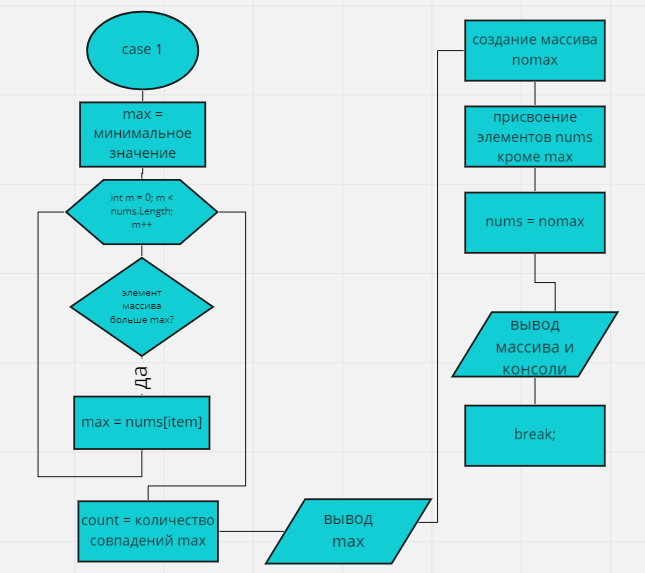
|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Класс входных данных** | **Выходные данные** |
| n | Не целое или натуральное число | Сообщение об ошибке ввода, повторный ввод |
| Целое натуральное положительное число | N = {n} |
| а\ nums[i] | Целые числа | Вывод массива nums |
| Не число или не целое число | Сообщение об ошибке ввода, повторный ввод |

**3. Алгоритм (блок-схема) **

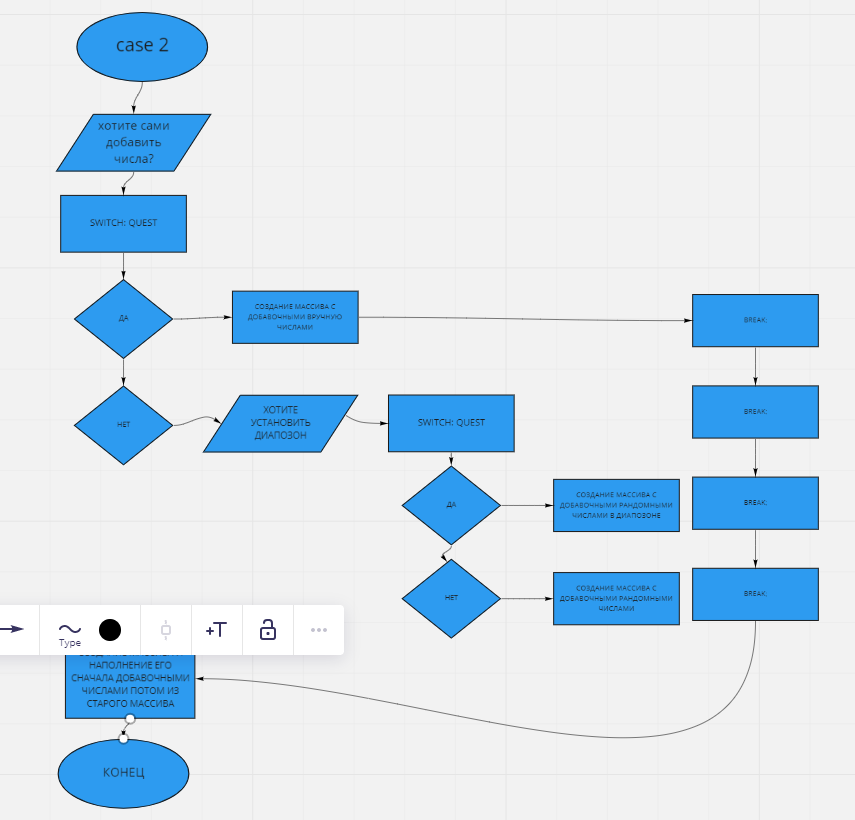
***Рис. 1. Блок-схема создания изначального массива.***



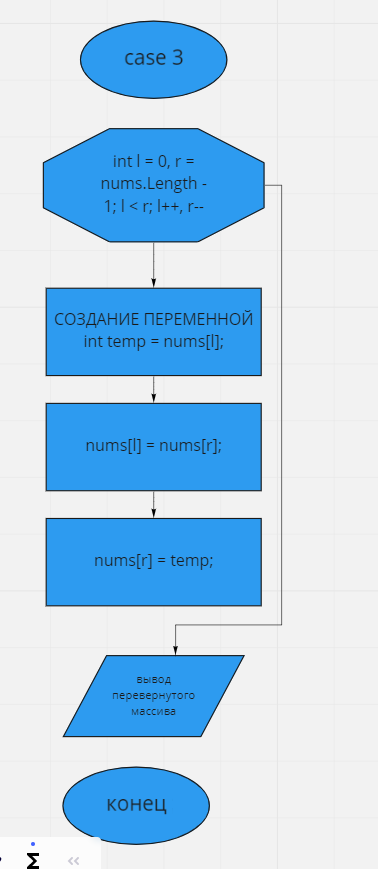
***Рис. 2. Блок-схема работы с массивом***



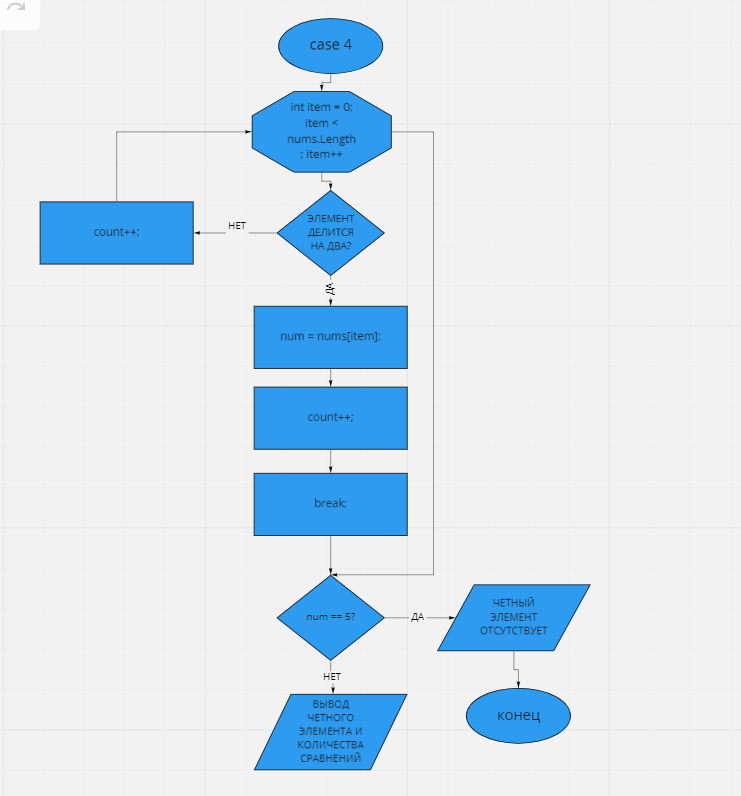
***Рис. 2. Блок-схема case 1 удаление максимального значения***



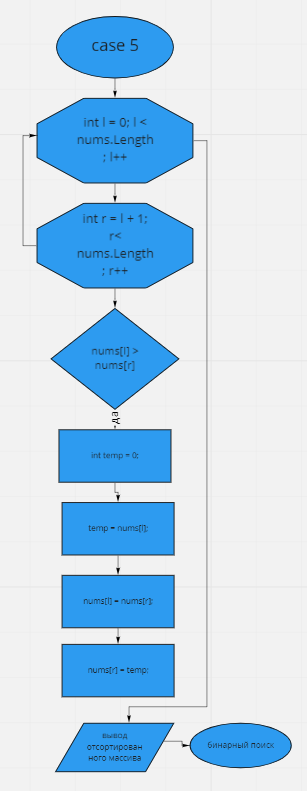
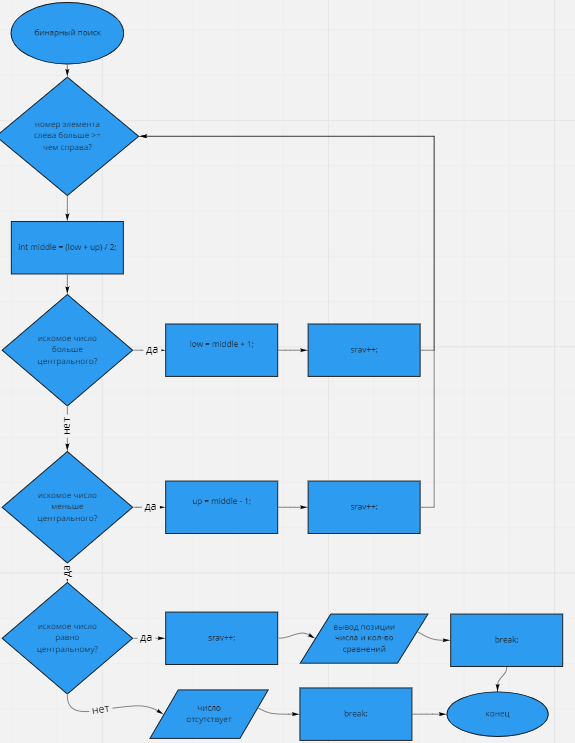
***Рис. 3. Блок-схема подпрограммы (КЕЙС 2 ДОБАВЛЕНИЕ ЧИСЕЛ В НАЧАЛО МАССИВА)***



***Рис. 3. Блок-схема подпрограммы (КЕЙС 3ПЕРЕВЕРНУТЬ МАССИВ)***



***Рис. 4. Блок-схема подпрограммы (КЕЙС 4 ПОИСК ПРЕВОГО ЧЕТНОГО ЭЛЕМЕНТА)***

****** ******

***Рис. 5. Блок-схема подпрограммы (КЕЙС 5 СОРТИРОВКА + БИНАРНЫЙ ПОИСК В ОТСОРТИРОВАННОМ МАССИВЕ)***

**4. Программа (листинг)**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_\_\_4

{

internal class Program

{

static int InputNumberOfElements(ref int CheckNumber)//проверка кол-ва элементов

{

CheckNumber = 0;

bool flag = false;//проверка

do

{

flag = int.TryParse(Console.ReadLine(), out CheckNumber);

if (CheckNumber < 0 || flag == false)

{

Console.WriteLine("Ошибка ввода");

flag = false;

}

} while (flag == false);

return CheckNumber;

}

static int InputNum(ref int Elements)//проверка вводимого числа

{

bool flag = false;

do

{

try

{

Elements = Convert.ToInt32(Console.ReadLine());

flag = true;

}

catch

{

Console.WriteLine("Ошибка ввода");

flag = false;

}

} while (flag == false);

return Elements;

}

static void ArrayIteration(int[] Array)//перебор массива

{

Console.Write("nums = ");

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

{

Console.Write($"{Array[i]} ");

}

}

public static void CreateArrayWithRand(int Elements,Random Randomiser,int a,int b,ref int[] Array)//наполнение массива с рандомом

{

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

{

Array[i] = Randomiser.Next(a,b);

}

}

public static void CreateArrayWithManual(ref int[] Array)//наполнение массива вручную

{

int a = 0;

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

{

InputNum(ref a);

Array[i] = a;

}

}

public static void DeleteMax(ref int[] nums,string console,string SmallConsole)//удаление макс элемента(кейс1)

{

int max = int.MinValue;//НАЧАЛО посика максимального значения

for (int m = 0; m < nums.Length; m++)

{

if (nums[m] > max)

{

max = nums[m];

}

}

int count = 0;

for (int item = 0; item < nums.Length; item++)

{

if (nums[item] == max)

{

count++;

}

}

Console.Clear();

Console.WriteLine($"\nmax = {max}");

int[] nomax = new int[nums.Length - count]; //НАЧАЛО создание массива без максимума

for (int item1 = 0, item2 = 0; item1 < nums.Length && item2 < nomax.Length; item1++, item2++)

{

if (nums[item1] != max) //не должен превышать размер номакс

{

nomax[item2] = nums[item1];

}

else // елси равен максимуму и не превышает размер

{

while (nums[item1] == max)

{

nomax[item2] = nums[item1 + 1];

item1++;

} //на случай если несколько макс подряд

}

}

Console.Write("массив без максимального элемента: ");

if (nomax.Length != 0)

{

ArrayIteration(nomax);

Console.WriteLine("\nчто-нибудь еще?(введите цифру)");

Console.WriteLine(console);

}

else

{

Console.WriteLine(SmallConsole);

}

nums = nomax;

//КОНЕЦ

}

public static void PlusSomeNums(int count,Random random,ref int [] nums)//добавление чисел(кейс2)

{

string quest = "";

int a = 0, b = 0;

int[] plusnums = new int[nums.Length + count];//НАЧАЛО создания массива с доб числами

int[] plus = new int[count];

Console.Clear();

Console.WriteLine("хотите сами добавить числа? да или нет");

do

{

quest = Console.ReadLine();

switch (quest)

{

case ("да"):

Console.WriteLine("прибавим числа: ");

CreateArrayWithManual(ref plus);

break;

case ("нет"):

Console.WriteLine("хотите установить диапозон выбора элемента массива? напишите да или нет");

do

{

quest = Console.ReadLine();

switch (quest)

{

case ("да"):

bool flag;

Console.Clear();

Console.WriteLine("Вводите целые числа. ");

do

{

flag = false;

Console.Write("от ");

InputNum(ref a);

Console.Write("до ");

InputNum(ref b);

if (a > b)

{

Console.WriteLine("ошибка ввода");

}

else flag = true;

} while (flag == false);

CreateArrayWithRand(count, random, a, b, ref plus);

Console.Clear();

break;

case ("нет"):

Console.Clear();

CreateArrayWithRand(count, random, int.MinValue, int.MaxValue, ref plus);

break;

default:

Console.WriteLine("ошибка ввода");

break;

}

} while (quest != "да" && quest != "нет");

Console.Write("прибавим числа: ");

ArrayIteration(plus);

break;

default:

Console.WriteLine("ошибка ввода");

break;

}

} while (quest != "да" && quest != "нет");

for (int item1 = 0, item2 = 0, item3 = 0; item1 < plusnums.Length; item1++, item2++)

{

if (item2 < count)

{

plusnums[item1] = plus[item2]; //заполнение новых чисел

}

else

{

plusnums[item1] = nums[item3];//заполнение старых чисел

item3++;

}

}

Console.Clear();

Console.WriteLine("\nНовый массив: ");

ArrayIteration(plusnums);

nums = plusnums;

}

public static void ReverseArray(int [] nums)//перевернуть массив(кейс3)

{

for (int l = 0, r = nums.Length - 1; l < r; l++, r--)

{

int temp = nums[l];

nums[l] = nums[r];

nums[r] = temp;

}

Console.Clear();

Console.Write("перевернутый массив:");

ArrayIteration(nums);

}

public static void FirstEvenElement(int[] nums, int num = 5, int count = 0)//первый четный(кейс4)

{

for (int item = 0; item < nums.Length; item++)

{

if (nums[item] % 2 == 0)

{

num = nums[item];

count++;

break;

}

else

{

count++;

}

}

Console.Clear();

ArrayIteration(nums);

if (num == 5)

{

Console.WriteLine($"\nв массиве отсутсвуют четные числа");

}

else

{

Console.WriteLine($"\nПервый четный элемент: {num}");

Console.WriteLine($"Для его нахождения было проведено {count} сравнений");

}

}

public static void SortArray(int[] nums)//сортировка(кейс5)

{

for (int l = 0; l < nums.Length; l++)

{

for (int r = l + 1; r < nums.Length; r++)

{

if (nums[l] > nums[r])

{

int temp = 0;

temp = nums[l];

nums[l] = nums[r];

nums[r] = temp;

}

}

}

Console.Clear();

Console.Write("отсортированный массив: ");

ArrayIteration(nums);

Console.WriteLine("какое число будем искать?");

int a = 0;

InputNum(ref a);

int low = 0;

int up = nums.Length;

int srav = 0;

int middle = 0;

while (low <= up)

{

middle = (low + up) / 2;

if (a > nums[middle])

{

low = middle + 1;

srav++;

}

else if (a < nums[middle])

{

up = middle - 1;

srav++;

}

if (a == nums[middle])

{

srav++;

string positions = "";

for (int i = middle, n = 1; middle + n < nums.Length && a == nums[middle + n]; n++)

{

positions += $" {middle + n + 1}";

}

for (int i = middle, n = 1; middle - n >= 0 && a == nums[middle - n] ; n++)

{

positions += $" {middle - n+1}";

}

Console.WriteLine("ваше число " + a + " нашли после " + srav + " сравнений " + "на " + (middle+1) + positions + " позиции");

break;

}

else if (low > up)

{

Console.WriteLine($"число {a} отсутсвует в данном массиве");

break;

}

}

}

static void Main(string[] args)

{

Console.WriteLine("введите колличество элементов");

int n = 0;//колличество элементов

bool flag = false;//проверка

string quest = "", console = " 0. закончить программу" +

"\n 1. Удалить максимальный элемент " +

"\n 2. Добавить К элементов в начало " +

"\n 3. Перевернуть массив " +

"\n 4. Найти первый четный элемент" +

"\n 5. Сортировка простым обменом";

string SmallConsole = "nums = пустой массив" +

"\n\nсписок функций с пустым массивом" +

"\n0. закончить программу" +

"\n2. Добавить К элементов в начало";

int a = 0, b = 0;

Random random = new Random();// рандомайзер

InputNumberOfElements(ref n);

Console.WriteLine($"n = {n}");

int[] nums = new int[n];//создание массива

if (n == 0)

{

Console.WriteLine(SmallConsole);

}

else

{

Console.WriteLine("хотите ввести числа сами? напишите да или нет");

do

{

quest = Console.ReadLine();// вопрос к пользователю

switch (quest)

{

case ("да"):

Console.WriteLine("Вводите числа. ");//ввод чисел вручную

CreateArrayWithManual(ref nums);

Console.Clear();

ArrayIteration(nums);

break;

case ("нет"):

Console.WriteLine("хотите установить диапозон выбора элемента массива? напишите да или нет");//ввод рандомных чисел

quest = Console.ReadLine();

do

{

switch (quest)

{

case ("да"):

Console.Clear();

Console.WriteLine("Вводите целые числа. ");

do

{

flag = false;

Console.Write("от ");

InputNum(ref a);

Console.Write("до ");

InputNum(ref b);

if (a > b)

{

Console.WriteLine("ошибка ввода");

}

else flag = true;

}while (flag == false);

Console.Clear();

CreateArrayWithRand(n, random, a, b, ref nums);

ArrayIteration(nums);

break;

case ("нет"):

Console.Clear();

CreateArrayWithRand(n, random,int.MinValue,int.MaxValue, ref nums);

ArrayIteration(nums);

break;

default:

Console.WriteLine("ошибка ввода");

break;

}

} while (quest != "да" && quest != "нет");

break;

default:

Console.WriteLine("ошибка ввода");

break;

}

} while (quest != "да" && quest != "нет");

Console.WriteLine("\nсписок функций с массивом ");

Console.WriteLine(console);

}

int check = 0;

int count = 0;

do

{

if (nums.Length != 0)

{

InputNum(ref check);

switch (check)

{

case 1:

DeleteMax(ref nums, console,SmallConsole);

break;

case 2:

count = 0;

Console.Clear();

Console.WriteLine("сколько эллементов хотите добавить?");//добавление элементов

InputNumberOfElements(ref count);

if (count == 0)

{

Console.WriteLine("Массив без изменений: ");

ArrayIteration(nums);

Console.WriteLine("\nчто-нибудь еще?(введите цифру)");

Console.WriteLine(console);

break;

}

else

{

PlusSomeNums(count, random, ref nums);

}

Console.WriteLine("\nчто-нибудь еще?(введите цифру)");

Console.WriteLine(console);

break;

case 3:

ReverseArray(nums);

Console.WriteLine("\nчто-нибудь еще?(введите цифру)");

Console.WriteLine(console);

break;

case 4:

int num = 0;

count = 0;

FirstEvenElement(nums);

Console.WriteLine("что-нибудь еще?(введите цифру)");

Console.WriteLine(console);

break;

case 5:

SortArray(nums);

Console.WriteLine("\nчто-нибудь еще?(введите цифру)");

Console.WriteLine(console);

break;

default:

if (check != 0)

{

Console.WriteLine("Ошибка ввода");

}

break;

}

}

else//если нулевой массив

{

do

{

InputNum(ref check);

switch (check)

{

case 2:

count = 0;

Console.Clear();

Console.WriteLine("сколько эллементов хотите добавить?");//добавление элементов

InputNumberOfElements(ref count);

if (count == 0)

{

Console.WriteLine("Массив без изменений: ");

ArrayIteration(nums);

Console.WriteLine("\nчто-нибудь еще?(введите цифру)");

Console.WriteLine(console);

break;

}

else

{

PlusSomeNums(count, random, ref nums);

}

Console.WriteLine("\nчто-нибудь еще?(введите цифру)");

Console.WriteLine(console);

break;

default:

if (check != 0)

{

Console.WriteLine("Ошибка ввода");

}

break;

}

} while (check != 0 && nums.Length == 0) ;

}

} while (check != 0);

}

} }

**5**. **Тесты (с проверкой достаточности по критериям черного ящика) для каждой задачи.**

***Табл. 1. Тестирование методом черного ящика.* Для кейса 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | Nums = {1,3,4} | 1,3 | 1,3 |
| **2** | Nums = {1} | nums = пустой массив | nums = пустой массив |
| **3** | Nums= {} | - | - |

***Табл. 2. Тестирование методом черного ящика.* Для кейса 2*.***

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | Nums = {1,3,4} plus {2} | Nums = {2,1,3,4} | Nums = {2,1,3,4} |
| **2** | Nums = {1,3,4} plus = {} | Массив без изменений: Nums = {1,3,4} | Массив без изменений: Nums = {1,3,4} |
| **3** | Nums= {} plus= {2} | Nums= {2} | Nums= {2} |
|  |  |  |  |
| **4** | Nums= {} plus= {} | Nums= {} | Nums= {} |

***Табл. 3. Тестирование методом черного ящика.* Для кейса 3*.***

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | Nums = {1,3,4} | Nums = {4,3,1} | Nums = {4,3,1} |
| **2** | Nums = {1} | Nums = {1} | Nums = {1} |
| **3** | Nums= {} | - | - |

***Табл. 4. Тестирование методом черного ящика.* Для кейса 4*.***

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | Nums = {1,3,4} | Первый четный элемент: 4  Для его нахождения было проведено 3 сравнений | Первый четный элемент: 4  Для его нахождения было проведено 3 сравнений |
| **2** | Nums = {1} | в массиве отсутсвуют четные числа | в массиве отсутсвуют четные числа |
| **3** | Nums= {} | - | - |

***Табл. 5. Тестирование методом черного ящика.* Для кейса 5*.***

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | Nums = {1,76,-5,2} искомое = 2 | nums = -5 1 2 76  ваше число 2 нашли после 1 сравнений на 3 позиции | nums = -5 1 2 76  ваше число 2 нашли после 1 сравнений на 3 позиции |
| **2** | Nums = {1,5,2,2} искомое = 2 | nums = 1 2 2 5  ваше число 2 нашли после 1 сравнений на 3 2 позиции | nums = 1 2 2 5  ваше число 2 нашли после 1 сравнений на 3 2 позиции |
| **3** | nums = 1 1 1 1 1  искомое = 1 | nums = 1 1 1 1 1  ваше число 1 нашли после 1 сравнений на 3 4 5 2 1 позиции | nums = 1 1 1 1 1  ваше число 1 нашли после 1 сравнений на 3 4 5 2 1 позиции |
| **4** | Nums = {1,5,2,2}  искомое = 3 | nums = 1 2 2 5  число 3 отсутсвует в данном массиве | nums = 1 2 2 5  число 3 отсутсвует в данном массиве |
| **5** | Nums= {} | - | - |