Пермский Национальный Исследовательский Политехнический Университет

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Лабораторная работа №6

по дисциплине

«Объектно-ориентированное программирование»

Выполнил:

студент группы РИС-19-1б

Шеретов М.А.

Проверила:

доцент кафедры ИТАС

Викентьева О. Л.

Пермь 2020

1. **Постановка задачи**

**Постановка задачи 1.**

1. Создать динамический массив (одномерный, двумерный, рваный) из элементов заданного типа. При заполнении массива использовать 2 способа (ручной и с помощью ДСЧ).

2. Массив вывести на печать.

3. Выполнить операции с массивом, указанные в варианте, используя, по возможности, методы класса Array.

4. Результаты обработки вывести на печать.

**Постановка задачи 2.**

1. Ввести строку символов с клавиатуры. Строка состоит из слов, разделенных пробелами (пробелов может быть несколько) и знаками препинания (, ;:). В строке может быть несколько предложений, в конце каждого предложения стоит знак препинания (.!?).

2. Выполнить обработку строки в соответствии с вариантом.

3. Результаты обработки вывести на печать.

**Вариант 1**

**Задача 1**

|  |  |  |
| --- | --- | --- |
| **Тип массива** | **Тип элементов** | **Операция** |
| Одномерный | char | Удалить из массива последнюю гласную букву. |

**Задача 2**

|  |
| --- |
| **Задание** |
| Перевернуть каждое слово, номер которого совпадает с его длиной. |

**2. Анализ задачи**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Описание функции** | **Входные данные** | **Классы входных данных** | **Выходные данные** |
| **Задание 1** | | | | |
| **1** | Формирование одномерного массива char | - Количество элементов массива N  - Элементы массива | - Целое число N > 0  - Элементы массива | Массив char |
| - Целое число N <= 0, вещественное число, строка, целое число | Сообщение об ошибке, повторный ввод |
| - Элементы – не символы | Сообщение об ошибке, повторный ввод |
| **2** | Поиск последней гласной буквы в массиве символов | - Массив | - Массив непустой  - есть гласные элементы | Индекс последней гласной буквы в массиве |
| - Массив непустой  - нет гласных элементов | -1 |
| **3** | Удаление последнего гласного элемента | - Массив  -индекс последнего гласного элемента | - Массив непустой  - Есть гласные элементы | Изменённый массив |
| - Массив непустой  -Нет гласных элементов | Сообщение об отсутствие гласных букв в массиве  Не изменённый массив |
| -Массив пуст | Сообщение об ошибке |
| -Массив не инициализирован | Сообщение об ошибке |
| **4** | Проверка массива на пустоту и выделение памяти | - Массив | - Память выделена  - Массив непустой | false |
| - Память не выделена  или  - Массив пустой | true |
| **Задание 2** | | | | |
| **5** | Ввод строки с клавиатуры | - Вводимая строка  - invMsg – сообщение приглашение  - errMsg – сообщение при ошибке | - Строка не состоит только из разделителей  - Не пустая строка | Строка |
| - Строка разделителей  - Пустая строка | Сообщение об ошибке |
| **6** | Создание разметки строки для работы с предложениями | Строка | -Строка | Строка (вместо слов «\_», знаки препинания и пробелы на месте) |
| **7** | Создание строки слов (без знаков препинания) | - Строка | -Строка | Строка (Состоит из слов, нет знаков препинания) |
| **8** | Переворот слов в строке, длина которых совпадает с номером в предложении | -Строка(исходная)  -Строка с разметкой  -Строка со словами | - Строка не состоит только из разделителей  - Строка состоит только из разделителей | Изменённая строка. |
| Строка пуста, или под неё не выделена память | Сообщение об ошибке. |
| **9** | Переворачивание строки | - Строка | - Строка | - Перевернутая строка |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вспомогательные функции** | | | | |
| **16** | Ввод целого числа с клавиатуры | - Строка  - Строка-приглашение  - Строка-ошибка | Строка – представляет собой целое число | Число |
| Не является целым числом | Сообщение об ошибке, повторный ввод |
| Очень большое целое число (переполнение) |
| **17** | Ввод натурального числа N | Количество элементов в массиве, стоблце или строке | Целое число N > 0 | Число N |
| Целое число N <= 0 | Сообщение об ошибке, повторный ввод |
| Символ/строка |
| Веществ. число |
| Не является числом | Сообщение об ошибке, повторный ввод |
| Очень большое число (переполнение) |
| **18** | Функция выбора из двух вариантов | - Строка вариант 1  - Строка вариант 2 | - Целые числа 1 или 2 | Выбранный вариант в виде числа |
| - Числа не 1 и не 2 | Ошибка, неверный вариант |
| **19** | Вывод в консоль | - Массив, строка | - Массив  - Строка | Печать в консоль |
| **20** | Печать меню |  |  |  |
| **21** | Реализация меню | - Клавиша | - Клавиша входит в множество вариантов | Выполнение программы |
| - Клавиша не входит в множество вариантов | Ошибка |

**3. Проектирование**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Функция** | **Входные данные** | **Выходные данные** |
| **Задание 1** | | | |
| **1** | Формирование одномерного массива char  MakeArr | - N - кол-во элементов массива  - Элементы массива | - char[] arr – массив  - Сообщение об ошибке, повторный ввод |
| **2** | Ввод данных в массив с помощью клавиватуры  InitArrMan\_char | - N - кол-во элементов массива  - Элементы массива | - char[] arr – массив  - Сообщение об ошибке, повторный ввод |
| **3** | Ввод данных в массив с  помощью ДСЧ  InitArrRnd\_char | - N - кол-во элементов массива | - char[] arr – массив  - Сообщение об ошибке, повторный ввод |
| **4** | Поиск последней гласной буквы в массиве символов  FindLastVowel | Char[] - массив элементов char | Int k — индекс последней гласной буквы  -1 если гласных нет |
| **5** | Удаление последней гласной буквы в массиве  DelElVowel | - char[] arr – массив  - int k — индекс последнего гласного элемента | Char[] arr – изменённый массив  -Сообщение об ошибке |
| **6** | Проверка массива на пустоту и выделение памяти  IsNullOrEmpty | - int[] arr – массив | True  false |
| **Задание 2** | | | |
| **7** | Ввод строки с клавиатуры  InputString | - string - Вводимая строка  - string invMsg – сообщение приглашение  - string errMsg – сообщение при ошибке | Строка  Сообщение об ошибке |
| **8** | Создание строки разметки  SetLayout | string | Строка разметки  -Сообщение об ошибке |
| **9** | Создание строки без знаков препинания  SetSentence | - string s - строка | Строка из слов  -Сообщение об ошибке |
| **10** | Переворот слов в строке при совпадение длины слова с номером слова  ReverseWords | - string s – строка  -Строка разметки  -Строка слов | Изменённая строка  -Сообщение об ошибке |
| **11** | Переворачивание строки  StringReverse | - string str - строка | String str – перевернутая строка |
| **Вспомогательные функции** | | | |
| **12** | InputNumInt (Ввод целого числа с клавиатуры) | Строка, которую вводит пользователь  String invMsg – сообщение-приглашение  String errMsg – сообщение при ошибке | Целое число  Сообщение об ошибке |
| **13** | InputChar(Ввод числа элементов массива N) | символ | символ  Сообщение об ошибке |
| **14** | MakeChoice (выбор из двух вариантов) | String s1, s2 - варианты | Целое число 1 или 2  Ошибка, неверный вариант |
| **15** | Вывод в консоль  Print | -int arr[] – массив  - string s – cтрока | Печать в консоль |
| **16** | Печать меню  PrintMenu |  | Печать меню |
| **17** | Реализация меню  Menu | - key – код клавиши | Выполнение программы  Ошибка |

**4. Программа**

using System;

using System.Text.RegularExpressions;

namespace OOPPractice6

{

    class Program

    {

        ///

        /// Статические переменные

        ///

            static int rndSeed = 1337;

            static Random rndGenerator = new Random(rndSeed);

        ///

        /// Начало программы(Функция main)

        ///

        static void Main(string[] args)

        {

            RunMenu();

        }

        ///

        /// Меню

        ///

        static void RunMenu()

        {

            int step;

            do

            {

                Console.Clear();

                PrintMenu(0);

                step = InputNumInt("> ", "Неверное число");

                switch (step)

                {

                    case 1:

                    Task1Menu();

                    PrintMenu(0);

                    break;

                    case 2:

                    Task2Menu();

                    PrintMenu(0);

                    break;

                    case 0:

                    break;

                    case 10:

                    PrintMenu(0);

                    break;

                    default:

                    Console.WriteLine("Неверный шаг");

                    break;

                }

                rndGenerator.Next(-256,255);

            } while (step != 0);

        }

        static void PrintMenu(int type)

        {

            Console.Clear();

            switch(type)

            {

                case 0:

                Console.WriteLine("\tЗадание");

                Console.WriteLine("1. Задание 1");

                Console.WriteLine("2. Задание 2");

                break;

                case 1:

                Console.WriteLine("\tЗадание 1");

                Console.WriteLine("1. Создать массив");

                Console.WriteLine("2. Печать массива");

                Console.WriteLine("3. Удалить из массива последнюю гласную букву");

                break;

                case 2:

                Console.WriteLine("\tЗадание 2");

                Console.WriteLine("1. Создать строку");

                Console.WriteLine("2. Печать строки");

                Console.WriteLine("3. Перевернуть каждое слово, номер которого совпадает с его длиной");

                break;

            }

            rndSeed = rndGenerator.Next(-256,255);

            rndGenerator.Next(-256,255);

            Console.WriteLine("\n10. Очистка консоли");

            Console.WriteLine("0. Назад");

        }

        static void Task1Menu()

        {

            PrintMenu(1);

            int step;

            char[] arrch = null;

            do

            {

                step = InputNumInt(">> ", "Ошибка ввода");

                switch (step)

                {

                    case 1:

                        arrch = MakeArr\_char(InputNumOfElements("Кол-во элементов в массиве?"));

                        Console.WriteLine("Инициализация массива окончена");

                        Console.WriteLine("Нажмите любую кнопку для продолжения...");

                        Console.ReadKey();

                    break;

                    case 2:

                        PrintArr(arrch);

                    break;

                    case 3:

                        arrch = DelElVowel(arrch);

                    break;

                    default:

                    break;

                }

                PrintMenu(1);

                rndSeed = rndGenerator.Next(-256,255);

            } while (step != 0);

        }

        static void Task2Menu()

        {

            PrintMenu(2);

            int step;

            string str = null;

            do

            {

                step = InputNumInt(">> ", "Ошибка ввода");

                switch (step)

                {

                    case 1:

                        str = CreateString();

                    break;

                    case 2:

                        PrintStr(str);

                    break;

                    case 3:

                        if(str != null)PrintStr("Строка до перестановки: " + str);

                            str = ReverseWords(str);

                        if(str != null)PrintStr("Строка после перестановки: " + str);

                    break;

                    default:

                    break;

                }

                PrintMenu(2);

            } while (step != 0);

        }

        ///

        ///Работа с массивами

        ///

        static char[] MakeArr\_char(int n)

        {

            Console.Clear();

            char[] resultArr = new char[n];

            Console.WriteLine("Каким способом заполнить массив?");

            bool isManual = MakeChoice("Вручную","ДСЧ");

            if(isManual)

            {

                resultArr = InitArrRnd\_char(n);

            }

            else

            {

                resultArr = InitArrMan\_char(n);

            }

            return resultArr;

        }

        static char[] InitArrRnd\_char(int length)

        {

            char[] arr = new char[length];

            Random rnd = new Random(rndSeed);

            Console.WriteLine("Инициализированный массив");

            for(int i = 0; i < length; i++)

            {

                arr[i] = Convert.ToChar(rnd.Next(0,256));

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

            }

            Console.WriteLine();

            rndSeed = rnd.Next(-length,length);

            return arr;

        }

        static char[] InitArrMan\_char(int length)

        {

            char[] arr = new char[length];

            Console.WriteLine("Инициализация массива: ");

            for(int i = 0; i < length; i++)

            {

                arr[i] = InputChar("","Введите символ!");

            }

            return arr;

        }

        static void PrintArr(char[] arr)

        {

            try{

            if(isNullOrEmpty(arr))

                throw new Exception("Массив не инициализирован или пуст");

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

            {

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

            }

            Console.WriteLine("Нажмите любую кнопку для продолжения...");

            Console.ReadKey();

            }

            catch(Exception er){

                Console.WriteLine("Ошибка!\n" + er.Message);

                Console.WriteLine("Нажмите любую кнопку для продолжения...");

                Console.ReadKey();

            }

        }

        static char[] DelElVowel(char[] array)

        {

            try

            {

                if(isNullOrEmpty(array))

                    throw new Exception("Массив не инициализирован или пуст");

                int k = FindLastVowel(array);

                if(k == -1)

                {

                    Console.WriteLine("Массив не содержит гласных букв");

                    Console.ReadKey();

                    return array;

                }

                char[] result = new char[array.Length-1];

                for (int i = 0; i < k; i++)

                {

                    result[i] = array[i];

                }

                for(int i = k+1; i < array.Length;i++)

                {

                    result[i-1] = array[i];

                }

                if(result.Length>0)

                    PrintArr(result);

                return result;

            }

            catch(Exception er)

            {

                Console.WriteLine("Ошибка! " + er.Message);

                Console.WriteLine("Нажмите любую кнопку для продолжения...");

                Console.ReadKey();

                return null;

            }

        }

        static int FindLastVowel(char[] array)

        {

            char[] vowels = new char[] {'у','е', 'ё','ы','а','о','э','я','и','ю','e','u','o','a','У','Е', 'Ё','Ы','А','О','Э','Я','И','Ю','E','U','O','A'};

            int k = array.Length - 1;

            for (int i = k; i  >= 0;i--)

            {

                for (int j = 0; j < vowels.Length; j++)

                {

                    if(array[i] == vowels[j])

                        return i;

                }

            }

            return -1;

        }

        ///

        ///Работа со строкой

        ///

        static string InitStringByManual(string invMsg = "",string errMsg = "Ошибка. Строка не инициализирована")

        {

            string result;

            try

            {

                Console.WriteLine(invMsg);

                result = Console.ReadLine();

                Console.WriteLine("Строка инициализирована");

                Console.WriteLine("Нажмите любую кнопку для продолжения...");

                Console.ReadKey();

            return result;

            }

            catch (System.Exception)

            {

                Console.WriteLine("Ошибка! " + errMsg);

                return null;

            }

        }

        static string InitStringByTemplates()

        {

            Console.WriteLine("\t Выбор примера:");

            string[] strArr = new string[] {"Hello world! How are you, my little friend?",

                                            "How are you? What are you doing?",

                                            "Здравствуй Мартин Алексеевич, твоя жизнь как?",

                                            "Чем вы нас покормите? - спросил голодный двоишник Петя Максимов.",

                                            "Может быть, тебе дать еще ключ от квартиры, где деньги лежат?",

                                            "Почём опиум для народа?",

                                            "Я, конечно, не херувим. У меня нет крыльев, но я чту Уголовный кодекс. Это моя слабость."};

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

            {

                Console.WriteLine($"{i+1}. {strArr[i]}");

            }

            int k = InputNumK(strArr.Length);

            return strArr[k-1];

        }

        static string CreateString()

        {

            bool solution = MakeChoice("Ввод вручную","Ввод с помощью программы");

            if(!solution)

            {

                return InitStringByManual("Введите строку: ", "Строка не инициализирована");

            }

            else

            {

                return InitStringByTemplates();

            }

        }

        static void PrintStr(string str)

        {

            try

            {

                if(str == null)

                    throw new Exception("\nСтрока не инициализирована!");

                Console.WriteLine(str);

                Console.WriteLine("Нажмите любую кнопку для продолжения...");

                Console.ReadKey();

            }

            catch(Exception er)

            {

                Console.WriteLine("Ошибка! " + er.Message);

                Console.WriteLine("Нажмите любую кнопку для продолжения...");

                Console.ReadKey();

            }

        }

        static string ReverseWords(string str)

        {

            try

            {

                if(str == null)

                    throw new Exception("Строка не инициализирована!");

                string layout = SetLayout(str);

                str = SetSentence(str);

                string[] strArr = str.Split(new char[]{' '}, StringSplitOptions.RemoveEmptyEntries);

                string[] layoutArr = layout.Split(new char[] {' '}, StringSplitOptions.RemoveEmptyEntries);

                Regex regex = new Regex(@"[\_]");

                string result = "";

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

                {

                    if(strArr[i].Length == i+1)

                        strArr[i] = StringReverse(strArr[i]);

                    layoutArr[i] = regex.Replace(layoutArr[i],strArr[i])+" ";

                    result += layoutArr[i];

                }

                return result;

            }

            catch(Exception er)

            {

                Console.WriteLine($"Ошибка!\n{er.Message}");

                Console.WriteLine("Нажмите любую кнопку для продолжения...");

                Console.ReadKey();

                return null;

            }

        }

        static string SetSentence(string str)

        {

            string result = " ";

            Regex regex = new Regex(@"\W");

            result = regex.Replace(str,result);

            return result;

        }

        static string SetLayout(string str)

        {

            string result = "\_";

            Regex regex = new Regex(@"\w(?<!\d)[\w'-]\*");

            result = regex.Replace(str,result);

            return result;

        }

        static string StringReverse(string str)

        {

            char[] array = str.ToCharArray();

            Array.Reverse(array);

            return new string(array);

        }

        ///

        ///Вспомогательные функции

        ///

        static int InputNumK(int rBord)

        {

            int k;

            do

            {

                k = InputNumInt($"Введите число от 1 до {rBord}: ","Ошибка!");

            }while(k<1 || k > rBord);

            return k;

        }

        static bool isNullOrEmpty(char[] arr)

        {

            if(arr == null || arr.Length == 0)

                return true;

            return false;

        }

        static bool MakeChoice(string sol1 = "Вариант 1", string sol2 = "Вариант 2")

        {

            Console.WriteLine("Выбор:\n"+"(0)"+sol1+"\t"+"(1)"+sol2);

            bool f = Convert.ToBoolean(InputNumInt("0/1\t","Введите целое число"));

            return f;

        }

        static int InputNumInt(string invMsg = "", string errMsg ="")//Ввод целового числа

        {

            Console.Write(invMsg);

            while(true)

            {

                int result;

                if(int.TryParse(Console.ReadLine(),out result))

                    return result;

                Console.WriteLine(errMsg);

                Console.Write(invMsg);

            }

        }

        static char InputChar(string invMsg = "", string errMsg ="")//Ввод целового числа

        {

            Console.Write(invMsg);

            while(true)

            {

                char result;

                try

                {

                    if(!char.TryParse(Console.ReadLine(), out result))

                        throw new Exception(errMsg);

                    return result;

                }

                catch(Exception er)

                {

                    Console.WriteLine(er.Message);

                }

            }

        }

        static int InputNumOfElements(string invMsg = "")

        {

            Console.Clear();

            Console.WriteLine(invMsg);

            int a;

            do

            {

                a = InputNumInt("","Введите целое число больше 0");

            }while(a < 1);

            return a;

        }

    }

}

**5 Тестирование**

**1. Формирование одномерного массива char**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер теста** | **Входные данные** | **Ожидаемый результат** | **Выходные данные** | **Примечание** |
| **1** | (Вручную)  5  a 2 c d e | a 2 c d e | a 2 c d e |  |
| **2** | (Вручную)  1  ! | ! | ! |  |
| **3** | (ДСЧ)  5 | 5 символов | Ф D e 3 ; |  |
| **4** | (ДСЧ)  1 | 1 символ | @ |  |
| **5** | (Вручную)  4  21 asd 2,44 1 | Ошибка | Введён не символ |  |
| **6** | 0 | Ошибка | Введите целое число > 0 |  |
| **7** | -1 | Ошибка | Введите целое число > 0 |  |
| **8** | asd | Ошибка | Неверное число |  |
| **9** | 22222222222222 | Ошибка | Неверное число |  |
| **10** | 1,234 | Ошибка | Неверное число |  |

**Входные данные**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Тесты** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| n – int > 0,  arr – 5 char | + |  |  |  |  |  |  |  |  |  |
| n – int > 0,  arr – 1 char |  | + |  |  |  |  |  |  |  |  |
| n – int > 0,  arr – 5 char(ДСЧ) |  |  | + |  |  |  |  |  |  |  |
| n – int > 0,  arr – 1 char(ДСЧ) |  |  |  | + |  |  |  |  |  |  |
| n – int > 0,  arr – string,  double |  |  |  |  | + |  |  |  |  |  |
| n = 0 |  |  |  |  |  | + |  |  |  |  |
| n = int < 0 |  |  |  |  |  |  | + |  |  |  |
| n – string |  |  |  |  |  |  |  | + |  |  |
| n – size > int |  |  |  |  |  |  |  |  | + |  |
| n – вещ. число |  |  |  |  |  |  |  |  |  | + |

**Выходные данные**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ошибка |  |  |  |  | + | + | + | + | + | + |
| Массив | + | + | + | + |  |  |  |  |  |  |

**2. Поиск последней гласной буквы в массиве символов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер теста** | **Входные данные** | **Ожидаемый результат** | **Выходные данные** | **Примечание** |
| **1** | 3 2 l У | 3 | 3 |  |
| **2** | X r P П ; ; | -1 | -1 |  |
| **3** | A a a A A a | 5 | 5 |  |
| **4** | -3 -5 55 87 1 | Ошибка | Введены не символы |  |
| **6** | null | Ошибка | Массив не создан |  |

**Входные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тесты** | 1 | 2 | 3 | 4 | 6 |
| 1 гласный эл. | + |  |  |  |  |
| Ни одного гласного элемента |  | + |  |  |  |
| Несколько гласных элементов |  |  | + |  |  |
| На ввод поступают не символы |  |  |  | + |  |
| Массив не создан |  |  |  |  | + |

**Выходные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ошибка |  |  |  | + | + |
| Массив | + | + | + |  |  |

**3. Удаление последней гласной буквы в массиве символов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер теста** | **Входные данные** | **Ожидаемый результат** | **Выходные данные** | **Примечание** |
| **1** | 3 2 l У | 3 2 l | 3 2 l |  |
| **2** | X r P П ; ; | Ошибка | В строке гласных не обнаружено |  |
| **3** | A a a A A | A a a A A | A a a A A |  |
| **4** | -3 -5 55 87 1 | Ошибка | Введены не символы |  |
| **6** | null | Ошибка | Массив не создан |  |

**Входные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тесты** | 1 | 2 | 3 | 4 | 6 |
| 1 гласный эл. | + |  |  |  |  |
| Ни одного гласного элемента |  | + |  |  |  |
| Несколько гласных элементов |  |  | + |  |  |
| На ввод поступают не символы |  |  |  | + |  |
| Массив не создан |  |  |  |  | + |

**Выходные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ошибка |  | + |  | + | + |
| Массив | + |  | + |  |  |

**4. Печать**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер теста** | **Входные данные** | **Ожидаемый результат** | **Выходные данные** | **Примечание** |
| **1** | A b c d e | A b c d e | A b c d e |  |
| **2** | Arr – пустой массив | Ошибка | Массив пуст |  |
| **3** | arr = null | Ошибка | Создайте массив |  |

**Входные данные**

|  |  |  |  |
| --- | --- | --- | --- |
| **Тесты** | 1 | 2 | 3 |
| char arr[] – непустой | + |  |  |
| char arr[] - пустой |  | + |  |
| arr = null |  |  | + |

**Выходные данные**

|  |  |  |  |
| --- | --- | --- | --- |
| Ошибка |  | + | + |
| Печать | + |  |  |

**5. Ввод строки с клавиатуры**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер теста** | **Входные данные** | **Ожидаемый результат** | **Выходные данные** | **Примечание** |
| **1** | S = Тестовая строка | S = Тестовая строка | S = Тестовая строка |  |
| **2** | S =” aaa “ | S =” aaa “ | S =” aaa “ |  |
| **3** | S = a!!!!! b???? | S = a!!!!! b???? | S = a!!!!! b???? |  |
| **4** | S = “ ” | Ошибка | Неверная строка | Строка из разделителей |
| **5** | S = “” | Ошибка | Неверная строка | Пустая строка |

**Входные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тесты** | 1 | 2 | 3 | 4 | 5 |
| Стандартная строка | + |  |  |  |  |
| Строка с пробелами в начале и в конце |  | + |  |  |  |
| Строка со знаками |  |  | + |  |  |
| Строка из разделителей |  |  |  | + |  |
| Пустая строка |  |  |  |  | + |

**Выходные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ошибка |  |  |  | + | + |
| Строка | + | + | + |  |  |

**6. Создание строки разметки**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер теста** | **Входные данные** | **Ожидаемый результат** | **Выходные данные** | **Примечание** |
| **1** | S = Тестовая строка | \_ = \_ \_ | \_ = \_ \_ |  |
| **2** | Hello world! How are you, my little prince? | \_ \_! \_ \_ \_, \_ \_ \_? | \_ \_! \_ \_ \_, \_ \_ \_? |  |
| **3** | a!!!!! b???? | \_!!!!! \_???? | \_!!!!! \_???? |  |
| **4** | S = “ ” | Ошибка | Неверная строка | Строка из разделителей |
| **5** | S = “” | Ошибка | Неверная строка | Пустая строка |

**Входные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тесты** | 1 | 2 | 3 | 4 | 5 |
| Стандартная строка | + |  |  |  |  |
| Строка с пробелами в начале и в конце |  | + |  |  |  |
| Строка со знаками |  |  | + |  |  |
| Строка из разделителей |  |  |  | + |  |
| Пустая строка |  |  |  |  | + |

**Выходные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ошибка |  |  |  | + | + |
| Строка | + | + | + |  |  |

**7. Создание строки из слов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер теста** | **Входные данные** | **Ожидаемый результат** | **Выходные данные** | **Примечание** |
| **1** | S = Тестовая строка | S Тестовая строка | S Тестовая строка |  |
| **2** | Hello world! How are you, my little prince? | Hello world How are you my little prince | Hello world How are you my little prince |  |
| **3** | a!!!!! b???? | a b | a b |  |
| **4** | S = “ ” | Ошибка | Неверная строка | Строка из разделителей |
| **5** | S = “” | Ошибка | Неверная строка | Пустая строка |

**Входные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тесты** | 1 | 2 | 3 | 4 | 5 |
| Стандартная строка | + |  |  |  |  |
| Строка с пробелами в начале и в конце |  | + |  |  |  |
| Строка со знаками |  |  | + |  |  |
| Строка из разделителей |  |  |  | + |  |
| Пустая строка |  |  |  |  | + |

**Выходные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ошибка |  |  |  | + | + |
| Строка | + | + | + |  |  |

**8. Переворот слов в строке**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер теста** | **Входные данные** | **Ожидаемый результат** | **Выходные данные** | **Примечание** |
| **1** | A ba cda awma | A ab adc amwa | A ab adc amwa |  |
| **2** | Hello world! How are you, my little prince? | Hello world! woH are you, my little prince? | Hello world! woH are you, my little prince? |  |
| **3** | a!!!!! b???? | a b | a b |  |
| **4** | S = “ ” | Ошибка | Неверная строка | Строка из разделителей |
| **5** | S = “” | Ошибка | Неверная строка | Пустая строка |

**Входные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тесты** | 1 | 2 | 3 | 4 | 5 |
| Стандартная строка | + |  |  |  |  |
| Строка с пробелами в начале и в конце |  | + |  |  |  |
| Строка со знаками |  |  | + |  |  |
| Строка из разделителей |  |  |  | + |  |
| Пустая строка |  |  |  |  | + |

**Выходные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ошибка |  |  |  | + | + |
| Строка | + | + | + |  |  |

**9. Ввод целого числа с клавиатуры**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер теста** | **Входные данные** | **Ожидаемый результат** | **Выходные данные** | **Примечание** |
| **1** | 5 | 5 | 5 |  |
| **2** | -5 | -5 | -5 |  |
| **3** | 0 | 0 | 0 |  |
| **4** | фывфыв | Ошибка | Неверное число |  |
| **5** | 1,2345 | Ошибка | Неверное число |  |
| **6** | -5,1 | Ошибка | Неверное число |  |
| **7** | 22222222222222 | Ошибка | Неверное число |  |

**Входные данные**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Тесты** | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Int > 0 | + |  |  |  |  |  |  |
| Int < 0 |  | + |  |  |  |  |  |
| 0 |  |  | + |  |  |  |  |
| string |  |  |  | + |  |  |  |
| double |  |  |  |  | + |  |  |
| Double < 0 |  |  |  |  |  | + |  |
| Size > int |  |  |  |  |  |  | + |

**Выходные данные**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ошибка |  |  |  | + | + | + | + |
| Число | + | + | + |  |  |  |  |

**10. Выбор из 2 вариантов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер теста** | **Входные данные** | **Ожидаемый результат** | **Выходные данные** | **Примечание** |
| **1** | 1 | 1 вариант | 1 |  |
| **2** | 2 | 2 вариант | 2 |  |
| **3** | 0 | Ошибка | Неверный вариант |  |
| **4** | фывфыв | Ошибка | Неверное число |  |
| **5** | 3 | Ошибка | Неверный вариант |  |
| **6** | -5,1 | Ошибка | Неверное число |  |
| **7** | 22222222222222 | Ошибка | Неверное число |  |

**Входные данные**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Тесты** | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Int 1 | + |  |  |  |  |  |  |
| Int 2 |  | + |  |  |  |  |  |
| 0 |  |  | + |  |  |  |  |
| string |  |  |  | + |  |  |  |
| Int не 1 или 2 |  |  |  |  | + |  |  |
| Double |  |  |  |  |  | + |  |
| Size > int |  |  |  |  |  |  | + |

**Выходные данные**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ошибка |  |  | + | + | + | + | + |
| Число | + | + |  |  |  |  |  |