Пермский филиал федерального государственного автономного образовательного учреждения высшего образования   
«Национальный исследовательский университет   
«Высшая школа экономики»

*Факультет экономики, менеджмента и бизнес-информатики*

Соколов Данила Дмитриевич

**Отчет по лабораторной работе №7**

студента образовательной программы «Бизнес-информатика»

по направлению подготовки *Бизнес-информатика*

Руководитель:

Преподаватель кафедры информационных технологий в бизнесе

Сахипова М.С

Пермь, 2020 год

**Постановка задачи**

1. Разработать Windows – приложение, реализующее работу с одномерными, двумерными и рваными массивами.
2. Реализовать сохранение массивов в файле и загрузку ранее сохраненного массива из файла.
3. Сформировать динамический одномерный массив, заполнить его случайными числами и вывести на печать.
4. Выполнить указанное в варианте задание и вывести полученный массив на печать.
5. Сформировать динамический двумерный массив, заполнить его случайными числами и вывести на печать.
6. Выполнить указанное в варианте задание и вывести полученный массив на печать.
7. Сформировать динамический двумерный массив, заполнить его случайными числами и вывести на печать.
8. Выполнить указанное в варианте задание и вывести полученный массив на печать.
9. При реализации функций необходимо продемонстрировать использование параметров разных типов и различные способы организации функций (параметры по умолчанию, перегрузку функций, и т. д.)

|  |  |  |  |
| --- | --- | --- | --- |
| № варианта | Одномерный массив | Двумерный массив | Рваный массив |
| 17 | Добавить после каждого четного элемента элемент со значением 0 | Удалить все строки, в которых есть хотя бы один нулевой элемент | Добавить строку с заданным номером |

**Анализ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Описание функции | Входные данные | Классы входных данных | Выходные данные |
| 1 | Меню для диалога с пользователем | * Число * Левая граница диапазона * Правая граница диапазона | Натуральное число в диапазоне | * Сообщение об ошибке |
| Натуральное число вне диапазона |
| Ненатуральное число |
| Не число |
| 2 | Печать одномерного массива | * Массив * Длина массива | Одномерный массив | * Массив (в виде строки) * Сообщение о том, что массив пустой |
| Пустой массив |
| 3 | Печать двумерного массива | * Массив * Кол-во строк * Кол-во столбцов | Двумерный массив | * Массив (в виде нескольких строк) * Сообщение о том, что массив пустой |
| Пустой массив |
| 4 | Печать рваного массива | * Массив * Кол-во строк * Кол-во столбцов | Рваный массив | * Массив (в виде нескольких строк) * Сообщение о том, что массив пустой |
| 5 | Формирование одномерного массива с помощью ДСЧ | * Кол-во элементов в массиве | n – натуральное число | * Массив целых чисел |
| n – ненатуральное число |
| n – не число |
| 6 | Формирование одномерного массива с клавиатуры | * Числа * Кол-во элементов(n) | Все числа целые | * Массив целых чисел |
| Есть хотя бы одно нецелое число |
| Есть хотя бы одно не число |
| n – натуральное число |
| n – ненатуральное число |
| n – не число |
| 7 | Формирование двумерного массива с помощью ДСЧ | * Кол-во строк(n) * Кол-во столбцов(m) | n – натуральное число | * Двумерный массив целых чисел |
| n – ненатуральное число |
| n – не число |
| m – натуральное число |
| m – ненатуральное число |
| m – не число |
| 8 | Формирование двумерного массива с клавиатуры | * Числа * Кол-во строк(n) * Кол-во столбцов(m) | Все числа целые | * Двумерный массив целых чисел |
| Есть хотя бы одно нецелое число |
| Есть хотя бы одно не число |
| n – натуральное число |
| n – ненатуральное число |
| n – не число |
| m – натуральное число |
| m – ненатуральное число |
| m – не число |
| 9 | Добавление нуля после каждого чётного элемента одномерного массива | * Одномерный массив * Длина массива | Массив, в котором есть чётный элементы | * Одномерный массив * Сообщение о том, что изменения не внесены, так как чётных нет |
| Массив, в котором нет чётных элементов |
| Пустой массив |
| 10 | Удаление из массива строк, содержащих нуль | * Двумерный массив * Кол-во строк(n) | Массив, в котором есть строки, содержащие ноль | * Двумерный массив * Сообщение о том, что изменения не внесены, так как строк, содержащих 0 нет |
| Массив, в котором нет строк, содержащих ноль |
| Пустой массив |
| 11 | Добавление строк в конец массива | * Двумерный рваный массив * Число строк(n) | Массив | * Двумерный рваный массив |
| Пустой массив |
| n - натуральное |
| n - ненатуральное |
| n – не число |

Проектирование форм

**Form 1**

Ниже представлен скриншот (см. рис. 1) первой формы. В ней пользователь работает с одномерным массивом. Массив может быть создан вручную, с помощью ДСЧ, открыт из файла, сохранен в файл, а так же могут быть добавлены «0» после четных элементов.

Первая форма содержит в себе следующие элементы:

|  |  |
| --- | --- |
| Элемент | Выполняемая функция |
| ComboBox1 | Происходит выбор 3 типов работы с программы:  1) одномерный  2) двумерный  3) рваный массивы |
| TextBox3 | Ввод количества элементов |
| TextBox1 | Поле для ввода элементов, не только на этой, но и на последующих страницах |
| TextBox2 | Поле для вывода элементов, не только на этой, но и на последующих страницах |
| Button1 | Считывание массива из TextBox1, не только на этой, но и на последующих страницах |
| Button2 | Считывание массива из файла, не только на этой, но и на последующих страницах |
| Button3 | Сохранение итогового массива в файл, не только на этой, но и на последующих страницах |
| Button4 | Сохранение изначального массива в файл, не только на этой, но и на последующих страницах |

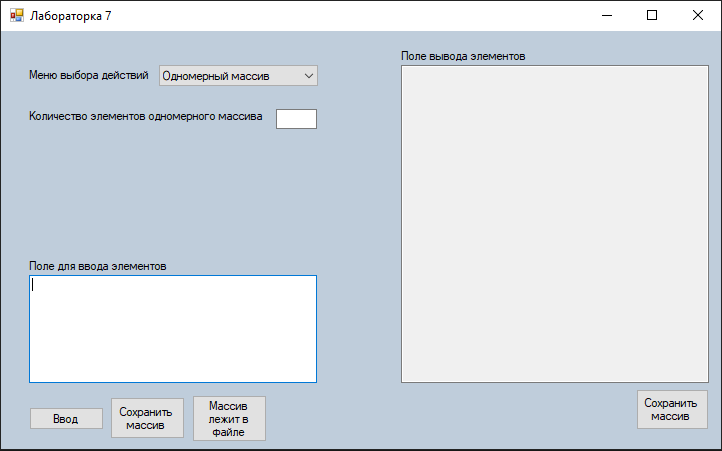


Рисунок 1. Form 1

**Form 2**

Ниже представлен скриншот (см. рис.2) второй формы. В ней пользователь работает с двумерным массивом. Массив может быть создан вручную, с помощью ДСЧ, открыт из файла, сохранен в файл, а так же могут быть удалены строки с «0».

Вторая форма содержит в себе следующие элементы:

|  |  |
| --- | --- |
| Элемент | Выполняемая функция |
| TextBox4 | Ввод количества строк в массиве |
| TextBox6 | Ввод количества столбцов массива |

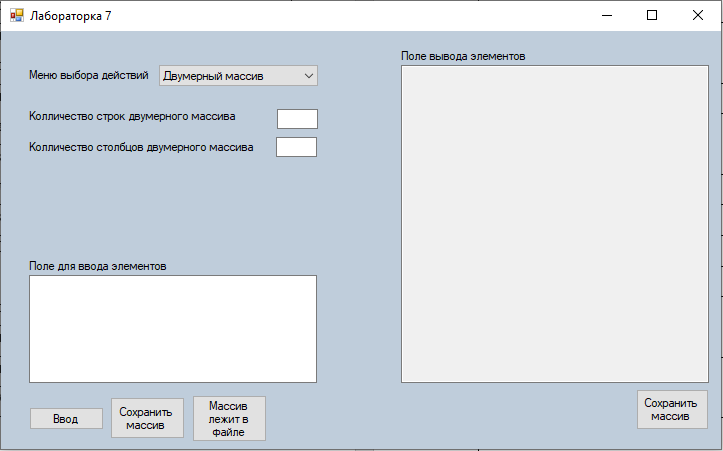


Рисунок 2. Form 2

**Form3**

Ниже представлен скриншот (рис. 3) третьей формы. В ней пользователь работает с рваным массивом. Массив может быть создан вручную, с помощью ДСЧ, открыт из файла, сохранен в файл, а так же в массив могут быть добавлены K строк, начиная с последней строки.

Третья форма содержит в себе следующие элементы:

|  |  |
| --- | --- |
| Элемент | Выполняемая функция |
| TextBox9 | Ввод индекса строки, в которую будет вставлена стока |
| TextBox10 | Ввод количества элементов строки |

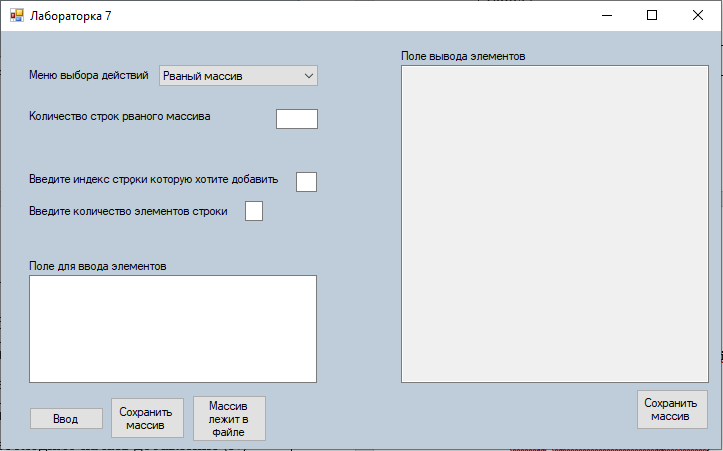


Рисунок 3. Form 3

**Разработка**

1. **Form1**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace L7

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form2 form2 = new Form2();

form2.Show();

Hide();

}

private void button2\_Click(object sender, EventArgs e)

{

Form3 form3 = new Form3();

form3.Show();

Hide();

}

private void button3\_Click(object sender, EventArgs e)

{

Form4 form4 = new Form4();

form4.Show();

Hide();

}

}

}

1. **Form2**

using System;

using System.Text;

using System.Windows.Forms;

using System.IO;

namespace laba7

{

public partial class Form1 : Form

{

public Form1()

{

StartPosition = FormStartPosition.CenterScreen;

InitializeComponent();

textBox2.ReadOnly = true;

comboBox1.SelectedIndex = 0;

label2.Show();

textBox3.Show();

label5.Hide();

textBox4.Hide();

label8.Hide();

textBox6.Hide();

label6.Hide();

textBox5.Hide();

label12.Hide();

textBox9.Hide();

label11.Hide();

textBox10.Hide();

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

switch (comboBox1.SelectedIndex)

{

case 0:

label2.Show();

textBox3.Show();

label5.Hide();

textBox4.Hide();

label8.Hide();

textBox6.Hide();

label6.Hide();

textBox5.Hide();

label12.Hide();

textBox9.Hide();

label11.Hide();

textBox10.Hide();

break;

case 1:

label2.Hide();

textBox3.Hide();

label5.Show();

textBox4.Show();

label8.Show();

textBox6.Show();

label6.Hide();

textBox5.Hide();

label12.Hide();

textBox9.Hide();

label11.Hide();

textBox10.Hide();

break;

case 2:

label2.Hide();

textBox3.Hide();

label5.Hide();

textBox4.Hide();

label8.Hide();

textBox6.Hide();

label6.Show();

textBox5.Show();

label12.Show();

textBox9.Show();

label11.Show();

textBox10.Show();

break;

}

}

public bool CheckO()

{

char[] c = { ' ', '\n' };

string tmp = textBox1.Text;

string[] s = tmp.Split(c);

int n = int.Parse(textBox3.Text);

if (s.Length != n)

{

return false;

} else

{

return true;

}

}

public bool CheckDArr()

{

char[] c = { ' ', '\n' };

string tmp = textBox1.Text;

string[] s = tmp.Split(c);

int n = int.Parse(textBox4.Text), m = int.Parse(textBox6.Text);

if (s.Length != n\*m)

{

return false;

}

else

{

return true;

}

}

public bool CheckText()

{

bool fl=true;

switch (comboBox1.SelectedIndex)

{

case 0:

if (textBox1.Text == "" || textBox3.Text == "" || textBox1.Text.Length==0 || textBox3.Text.Length == 0)

{

fl = false;

}

else

{

fl = true;

}

break;

case 1:

if (textBox1.Text == "" || textBox1.Text.Length == 0 || textBox4.Text == "" || textBox4.Text.Length == 0 || textBox6.Text == "" || textBox6.Text.Length == 0)

{

fl = false;

}

else

{

fl = true;

}

break;

case 2:

if (textBox1.Text == "" || textBox1.Text.Length == 0 || textBox5.Text == "" || textBox5.Text.Length == 0 || textBox9.Text == "" || textBox9.Text.Length == 0 || textBox10.Text == "" || textBox10.Text.Length == 0)

{

fl = false;

}

else

{

fl = true;

}

break;

}

if (fl==false)

MessageBox.Show("Ошибка. Есть пустое поле, запоните все поля ввода");

return fl;

}

public bool CheckRvArr()

{

char[] c = { ' ', '\n' };

string tmp = textBox1.Text;

string[] s = tmp.Split(c);

int n = int.Parse(textBox5.Text);

int q = 0,sum=n;

for (int i = 0; i < n; ++i, ++q)

{

int w = int.Parse(s[q]);

sum += w;

for (int y = 0; y < w; ++y, ++q)

{

try

{

int.Parse(s[q]);

}

catch

{

MessageBox.Show("Ошибка. Размер массива не совпадает с количетсвом элементов");

break;

}

}

}

if (q!=sum)

{

return false;

}

else

{

return true;

}

}

public bool CheckRvK()

{

int k = int.Parse(textBox9.Text);

int n = int.Parse(textBox5.Text);

if (k > n)

{

MessageBox.Show($"Неправильно введен индекс. Введите индекс от 0 до {n + 1}");

return false;

}

return true;

private void button1\_Click(object sender, EventArgs e)

{

bool fl;

StringBuilder s = new StringBuilder();

if (CheckText() == false) return;

switch (comboBox1.SelectedIndex)

{

case 0:

if (CheckO())

{

s = Osnova.Osnova.ArrOne(textBox3.Text, textBox1.Text);

textBox2.Text = s.ToString();

}

else

{

MessageBox.Show("Ошибка. Размер массива не совпадает с количетсвом элементов");

}

break;

case 1:

fl = true;

if (fl && CheckDArr())

{

s = Osnova.Osnova.ArrDouble(textBox4.Text, textBox6.Text, textBox1.Text);

textBox2.Text = s.ToString();

}

else

{

if (fl) MessageBox.Show("Ошибка. Размер массива не совпадает с количетсвом элементов");

}

break;

case 2:

fl = CheckRvK();

if (fl && CheckRvArr())

{

s = Osnova.Osnova.ArrRv(textBox5.Text, textBox1.Text, textBox9.Text, textBox10.Text);

textBox2.Text = s.ToString();

}

else

{

if (fl) MessageBox.Show("Ошибка. Размер массива не совпадает с количетсвом элементов");

}

break;

}

}

private void OpenText(object sender, EventArgs e)

{

OpenFileDialog OPF = new OpenFileDialog();

string s = AppDomain.CurrentDomain.BaseDirectory;

OPF.InitialDirectory = s;

OPF.Filter = "txt files (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

OPF.FilterIndex = 2;

OPF.RestoreDirectory = true;

if (OPF.ShowDialog() == DialogResult.Cancel)

return;

string filename = OPF.FileName;

string fileText = File.ReadAllText(filename);

StringBuilder txt = new StringBuilder();

txt.Append(fileText);

switch (int.Parse(txt[0].ToString()))

{

case 0:

comboBox1.SelectedIndex = int.Parse(txt[0].ToString());

txt.Remove(0,2);

textBox3.Text = txt[0].ToString();

txt.Remove(0, 2);

textBox1.Text = txt.ToString();

break;

case 1:

comboBox1.SelectedIndex = int.Parse(txt[0].ToString());

txt.Remove(0, 2);

textBox4.Text = txt[0].ToString();

textBox6.Text = txt[2].ToString();

txt.Remove(0, 4);

textBox1.Text = txt.ToString();

break;

case 2:

comboBox1.SelectedIndex = int.Parse(txt[0].ToString());

txt.Remove(0, 2);

textBox5.Text = txt[0].ToString();

txt.Remove(0, 2);

textBox1.Text = txt.ToString();

break;

}

}

private void textBox4\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar <= 47 || e.KeyChar >= 58) && e.KeyChar != 32 && e.KeyChar != 13 && e.KeyChar != 8 && e.KeyChar != 46)

{

e.Handled = true;

}

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar <= 47 || e.KeyChar >= 58) && e.KeyChar != 8 && e.KeyChar != 46)

{

e.Handled = true;

}

}

private void SaveText(object sender, EventArgs e)

{

SaveFileDialog SFD = new SaveFileDialog();

string s = AppDomain.CurrentDomain.BaseDirectory;

SFD.InitialDirectory = s;

SFD.Filter = "txt files (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

SFD.FilterIndex = 2;

SFD.RestoreDirectory = true;

switch (comboBox1.SelectedIndex)

{

case 0:

if (SFD.ShowDialog() != null)

File.WriteAllText(SFD.FileName, 0 + "\n" + textBox3.Text + "\n" + textBox1.Text);

break;

case 1:

if (SFD.ShowDialog() != null)

File.WriteAllText(SFD.FileName, 1 + "\n" + textBox4.Text + " " + textBox6.Text + "\n" + textBox1.Text);

break;

case 2:

if (SFD.ShowDialog() != null)

File.WriteAllText(SFD.FileName, 2 + "\n" + textBox5.Text + "\n" + textBox1.Text);

break;

}

}

}

}

}

using System;

using System.Text;

namespace Osnova

{

class Osnova

{

private static void AddElem(ref int[,] arr,int k,int n)

{

Random rnd = new Random();

int[,] f = new int[arr.GetLength(0), arr.GetLength(1) + k];

for (int i = 0; i < f.GetLength(0); i++)

{

for (int y = 0; y < n; ++y)

{

f[i, y] = arr[i, y];

}

}

for (int i = 0; i < f.GetLength(0); i++)

{

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

{

f[i, y + n] = rnd.Next(100);

}

}

for (int i = 0; i < f.GetLength(0); i++)

{

for (int y = n; y < arr.GetLength(1); ++y)

{

f[i, y + k] = arr[i, y];

}

}

arr = f;

}

private static void AddStr(ref int[][] arr,int k,int n)

{

Random rnd = new Random();

int[] tmp = new int[n];

int[][] f = new int[arr.Length + 1][];

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

tmp[i] = rnd.Next(0, 100);

for (int i = 0; i < Math.Min(k, arr.Length); ++i)

{

f[i] = arr[i];

}

f[k] = tmp;

for (int i = k + 1; i < f.GetLength(0); ++i)

{

f[i] = arr[i - 1];

}

arr = f;

}

public static void AddZero(ref int[] mas)

{

int countZeroes = 0;

int[] tmp = new int[mas.Length];

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

{

if (mas[i] % 2 == 0) countZeroes++;

tmp[i] = mas[i];

}

if (countZeroes != 0)

{

int count = 0;

Array.Resize(ref tmp, mas.Length + countZeroes);

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

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

{

count++;

tmp[i + count - 1] = mas[i];

tmp[i + count] = 0;

}

else

tmp[i + count] = mas[i];

}

Array.Resize(ref mas, mas.Length + countZeroes);

mas = tmp;

}

public static StringBuilder ArrOne(string a,string b)

{

char[] c = {' ', '\n'};

int n = int.Parse(a);

int[] arrO = new int[n];

string[] s = b.Split(c);

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

arrO[i] = int.Parse(s[i]);

AddZero(ref arrO);

StringBuilder ans = new StringBuilder();

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

{

ans.Append( arrO[i] + " ");

}

return ans;

}

public static StringBuilder ArrDouble(string a,string z, string b)

{

char[] c = { ' ', '\n' };

int n = int.Parse(a), m = int.Parse(z);

string[] s = b.Split(c);

int[,] arrD = new int[n,m];

int q = 0;

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

{

for (int y = 0; y < m; ++y, ++q)

{

arrD[i, y] = int.Parse(s[q]);

}

}

DeleteStringWithZero(ref arrD);

StringBuilder ans = new StringBuilder();

for (int i = 0; i < arrD.GetLength(0); ++i)

{

for (int y = 0; y < arrD.GetLength(1); ++y)

{

ans.Append(arrD[i, y] + " ");

}

ans.Append("\r\n");

}

return ans;

}

public static void DeleteStringWithZero(ref int[,] matr)

{

int count = 0;

for (int i = 0; i < matr.GetLength(0); i++)

{

for (int j = 0; j < matr.GetLength(1); j++)

{

if (matr[i, j] == 0)

{

count++;

break;

}

}

}

int[,] tmp = new int[matr.GetLength(0) - count, matr.GetLength(1)];

if (count != 0)

{

count = 0;

for (int i = 0; i < matr.GetLength(0); i++)

{

for (int j = 0; j < matr.GetLength(1); j++)

{

if (matr[i, j] == 0)

{

count++;

break;

}

else if (j == matr.GetLength(1) - 1)

{

for (int k = 0; k < matr.GetLength(1); k++)

{

tmp[i - count, k] = matr[i, k];

}

}

}

}

matr = new int[tmp.GetLength(0), tmp.GetLength(1)];

for (int i = 0; i < matr.GetLength(0); i++)

{

for (int j = 0; j < matr.GetLength(1); j++)

{

matr[i, j] = tmp[i, j];

}

}

}

}

public static StringBuilder ArrRv(string a,string k, string fi, string se)

{

char[] c = { ' ', '\n' };

int n = int.Parse(a);

string[] s = k.Split(c);

int[][] arrR = new int[n][];

int q = 0;

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

{

int w = int.Parse(s[q]);

++q;

arrR[i] = new int[w];

for (int y = 0; y < w; ++y, ++q)

{

arrR[i][y] = int.Parse(s[q]);

}

}

int e = int.Parse(fi), t = int.Parse(se);

AddStr(ref arrR, e, t);

StringBuilder ans = new StringBuilder();

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

{

for (int y = 0; y < arrR[i].Length; ++y)

{

ans.Append( arrR[i][y] + " ");

}

ans.Append("\r\n");

}

return ans;

}

}

}

**Тестирование**

* + - 1. Формирование одномерного массива

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Входные данные | Ожидаемый результат | Реальный результат | Примечание |
| 1 | 6 | 6 чисел | 53 -1 -5 36 40 98 | Ввод размера массива |
| 2 | 0 | Сообщение об ошибке | Ошибка! Создание пустого массива! |
| 3 | -3 | Сообщение об ошибке | Ошибка! Создание пустого массива! |
| 4 | 1,34 | Сообщение об ошибке | Ошибка! Создание пустого массива! |
| 5 | q | Сообщение об ошибке | Ошибка! Создание пустого массива! |
| 6 | 1 | 1 число | 4 |
| 7 | 124 | 124 | 124 | Ввод элемента массива с клавиатуры |
| 8 | 3254352347657 | Сообщение об ошибке | Ошибка при вводе числа |
| 9 | А | Сообщение об ошибке | Ошибка при вводе числа |
| 10 | -100 | -100 | -100 |
| 11 | 1.25 | Сообщение об ошибке | Ошибка при вводе числа |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № теста | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Ввод с ошибкой размера массива |  | + | + | + | + |  |  |  |  |  |  |
| Правильный ввод размера массива | + |  |  |  |  | + |  |  |  |  |  |
| Ввод с ошибкой элементов массива |  |  |  |  |  |  |  | + | + |  | + |
| Правильный ввод элементов массива |  |  |  |  |  |  | + |  |  | + |  |

* + - 1. Вывод одномерного массива

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Входные данные | Ожидаемый результат | Реальный результат |
| 1 | 53 -1 -5 36 40 98 | 3 -11 -5 36 0 42 0 98 0 | 3 -11 -5 36 0 42 0 98 0 |
| 2 | Пустой массив | Ошибка! Массив не создан! | Ошибка! Массив не создан! |

* + - 1. Добавление 0 после четных элементов массива

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Входные данные | Ожидаемый результат | Реальный результат |
| 1 | 13 25 63 80 90 3 | 13 25 63 80 0 90 0 3 | 13 25 63 80 0 90 0 3 |
| 2 | 913 | 913 | 913 |

* + - 1. Создание двумерного массива

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Входные данные | Ожидаемый результат | Реальный результат | Примечание |
| 1 | 2 | 2 строки | Продолжение работы (ожидание ввода кол-ва столбцов) | Ввод количества строк в массиве |
| 2 | 0 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 3 | -3 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 4 | 1,34 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 5 | q | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 6 | 1 | 1 строка | Продолжение работы (ожидание ввода кол-ва столбцов) |
| 7 | 2 | 2 столбца | 5 7  3 9 | Ввод количества столбцов в массиве |
| 8 | 0 | Сообщение об ошибке | Ошибка! |
| 9 | -3 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 10 | 1,34 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 11 | q | Сообщение об ошибке | Ошибка! Повторите ввод! |
|  |  |  |  |  |
| 12 | 1 | 1 столбец | 5 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № теста | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Ввод с ошибкой количества строк в массиве |  | + | + | + | + | + |  |  |  |  |  |  |
| Правильный ввод количества строк в массиве | + |  |  |  |  |  | + |  |  |  |  | + |
| Ввод с ошибкой количества столбцов в массиве |  |  |  |  |  |  |  | + | + | + | + |  |
| Правильный ввод количества столбцов в массиве |  |  |  |  |  |  | + |  |  |  |  | + |

* + - 1. Печать двумерного массива

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Входные данные | Ожидаемый результат | Реальный результат |
| 1 | 3 4 0 0  54 6 53 12 | 54 6 53 12 | 54 6 53 12 |
| 2 | Пустой массив | Сообщение об ошибке | Ошибка! Массив не создан! |

* + - 1. Удаление строк с хотя бы одним 0.

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Входные данные | Ожидаемый результат | Реальный результат |
| 1 | 0 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 2 | -3 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 3 | 1,34 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 4 | q | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 5 | 1 | 1 столбец на добавление | Продолжение работы программы (ожидание ввода N) |
| 6 | 3 | 3 столбца на добавление | Продолжение работы программы (ожидание ввода N) |
| 7 | 3 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 8 | -3 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 9 | q | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 10 | 0 | Сообщение об ошибке | Ошибка! Повторите ввод! |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Есть 0 | + | + | + | + |  |  |  |  |  |  |  |
| Нет 0 |  |  |  |  | + | + |  |  |  |  |  |
| Ввод с ошибкой |  |  |  |  |  |  | + | + | + | + |  |
| Правильный ввод |  |  |  |  |  |  |  |  |  |  | + |

* + - 1. Создание рваного массива

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Входные данные | Ожидаемый результат | Реальный результат | Примечание |
| 1 | 0 | Сообщение об ошибке | Ошибка! Повторите ввод! | Ввод количества строк в массиве |
| 2 | -3 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 3 | 1,34 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 4 | q | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 5 | 1 | 1 строка | Продолжение работы программы (ожидание ввода остальных параметров) |
| 6 | 3 | 3 строки | Продолжение работы программы (ожидание ввода остальных параметров) |
| 7 | 0 | Сообщение об ошибке | Ошибка! Повторите ввод! | Ввод количества элементов в строке |
| 8 | -3 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 9 | 1,34 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 10 | q | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 11 | 1 | 1 элемент в строке | Продолжение работы программы (ожидание ввода элементов) |
| 12 | 3 | 3 элемента в строке | Продолжение работы программы (ожидание ввода элементов) |
| 13 | -3 | Сообщение об ошибке | Ошибка! Повторите ввод! | Ввод элемента строки |
| 14 | 1,34 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 15 | q | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 16 | 0 | Первый элемент в первой строке | Продолжение работы программы (ожидание ввода остальных элементов) |
| 17 | 10 | Первый элемент в первой строке | Продолжение работы программы (ожидание ввода остальных элементов) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| Ввод с ошибкой количества строк в массиве | + | + | + | + |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Правильный ввод количества строк в массиве |  |  |  |  | + | + |  |  |  |  |  |  |  |  |  |  |  |
| Ввод с ошибкой количества элементов в строке |  |  |  |  |  |  | + | + | + | + |  |  |  |  |  |  |  |
| Правильный ввод количества элементов в строке |  |  |  |  |  |  |  |  |  |  | + | + |  |  |  |  |  |
| Ввод с ошибкой элемента строки |  |  |  |  |  |  |  |  |  |  |  |  | + | + | + |  |  |
| Правильный ввод элемента строки |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | + | + |

* + - 1. Печать рваного массива

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Входные данные | Ожидаемый результат | Реальный результат |
| 1 | -89  -70 72  27 -60 57 | -89  **-**70 72  27 -60 57 | -89  **-**70 72  27 -60 57 |
| 2 | Пустой массив | Сообщение об ошибке | Ошибка! Массив не создан! |

* + - 1. Добавление строки с заданным номером

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Входные данные | Ожидаемый результат | Реальный результат | Примечание |
| 1 | 0 | Сообщение об ошибке | Ошибка! Повторите ввод! | Ввод номера строки |
| 2 | -3 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 3 | 1,34 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 4 | q | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 5 | 1 | Добавление строки в начало массива | Продолжение работы программы (ожидание ввода остальных элементов) |
| 6 | 3 | Добавление строки на вторую (третью для пользователя) строку | Продолжение работы программы (ожидание ввода остальных элементов) |
| 7 | 0 | Сообщение об ошибке | Ошибка! Повторите ввод! | Ввод количества элементов в строке |
| 8 | -3 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 9 | 1,34 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 10 | q | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 11 | 0 | Сообщение об ошибке | Ошибка! Повторите ввод! |
| 12 | 1 | Один элемент | 8  -50 29  91 83 74 |
| 13 | 3 | Три элемента | -50 29  91 83 74  6 10 3 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Ввод с ошибкой номера строки | + | + | + | + |  |  |  |  |  |  |  |  |  |
| Правильный ввод номера строки |  |  |  |  | + | + |  |  |  |  |  |  |  |
| Ввод с ошибкой количества элементов в строке |  |  |  |  |  |  | + | + | + | + | + |  |  |
| Правильный ввод количества элементов в строке |  |  |  |  |  |  |  |  |  |  |  | + | + |