Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт Информационных Технологий и Анализа Данных

Кафедра вычислительной техники

**Название работы** – “ Обработка командной строки ”

Отчет по лабораторной работе “Обработка командной строки”

Вариант 9

по дисциплине Информатика

Выполнил

Студент, номер группы ИСТБ-19-2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ M. В. Козлов

Дата: \_\_\_\_\_\_\_\_\_\_

Принял

Должность \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. А. Осипова

Дата: \_\_\_\_\_\_\_\_\_\_

Иркутск 2020

Задание 9.

9.Переписать элементы матрицы А в матрицу В, исключить из нее все повторяющиеся элементы или заменить их на определенный символ (0..9).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | args[0], args[1], args[2],  args[3],args[4] |  | | | |
| Main |
|  |
|  | | | | | | | | |
|  |  | matrix, size1, size2 |  | matrix, size1, size2 |  | size1, size2 |  | matrix, size1, size2 |
| InputMatr | UniqueMatrixChar | UniqueMatrix | InputRandomMatr | OutputArray |
| N | N | N | N |  |

|  |
| --- |
| n, size1,size2 |
| UserChoiceYes |
|  |

|  |
| --- |
| n, size1,size2 |
| UserChoiceNo |
|  |

Что делает каждая из функций:

Main() - главная функция, где происходит вызов других функций

InputMatr () – функция ввода значения элементов матриц

UniqueMatrix () – функция проверки матриц на проверяющиеся элементы

OutputArray () – функция вывода матриц

InputRandomMatr() – функция ввода случайных значений элементов матрицы

UserChoiceYes – функция, выполняющая расчёт работы при заданном параметре Yes

UserChoiceNo функция, выполняющая расчёт работы при заданном параметре No

Функция Main:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Входные данные. | | | Выходные данные. | | |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| args[0]  args[1]  args[2]  args[3]  args[4] | Параметр, обозначающий выбор пользователя | Текст |  |  |  |

Словесное описание:

1. ЕСЛИ ДЛИНА args >0
   1. ЕСЛИ args[0] == "Help" ВЫПОЛНИТЬ
      1. ВЫВОД СООБЩЕНИЯ (

[параметр0] - напишите Help, чтобы вызвать справку, чтобы продолжить работу программы, напишите \* вместо Help

[параметр1]-введите количество строк матрицы (от 2 до 10)

[параметр2] введите количество столбцов матрицы (от 2 до 10)

[параметр3] - напишите Yes, если хотите самостоятельно заполнить матрицу [параметр3] - напишите No, если хотите чтобы матрицу заполнил компьтер

[параметр4] - напишите Null,если хотите, чтобы все повторяющиеся элементы были заменены на 0

[параметр 4] - напишите Char, чтобы заменить повторяющиеся элементы на определённый символ(0...9)")

* 1. ЕСЛИ args[0] == "\*" ВЫПОЛНИТЬ
     1. size1 = args[1]
     2. size2 = args[2]
     3. ЕСЛИ size1>0 ИЛИ size2>0 ВЫПОЛНИТЬ
        1. ЕСЛИ args[3] РАВЕН Yes ВЫПОЛНИТЬ
           1. Вызов функции UserChoiceYes
           2. ЕСЛИ ВСЕ
        2. ЕСЛИ args[3] РАВЕН No ВЫПОЛНИТЬ
           1. Вызов функции UserChoiceNo
           2. ЕСЛИ ВСЕ

ИНАЧЕ

* + - 1. Вывод сообщения (" Введите корректный 3 параметр!")

ИНАЧЕ

* + 1. Вывод сообщения ("Ошибка: 1 и 2 параметры меньше или равны 0")
  1. Вывод сообщения ("Введите Help для вызова справки или \*")

1. Вывод сообщения (" Данная программа запускается ТОЛЬКО с определённым параметром, пример ввода параметра:

(Название программы) [параметр0] [параметр1] [параметр2] [параметр3] [параметр4].

Для вызова справки введите параметр Help вместо [параметр0], как показано на схеме(через пробел).")

Функция InputMatr:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Входные данные. | | | Выходные данные. | | |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| size1 | Количество строк матрицы | Целые числа | N | Матрица | Целые числа |
| size2 | Количество столбцов матрицы | Целые числа |  |  |  |

Словесное описание:

1. i=0
2. ПОКА i < size1 ВЫПОЛНИТЬ
   1. N[ i ] = [m]
      1. i=0
      2. ПОКА i < size1 ВЫПОЛНИТЬ
         1. j=0
         2. ПОКА j < m ВЫПОЛНИТЬ
            1. Объявляем value - ЦЕЛОЕ
            2. ПОКА value НЕ ЦЕЛОЕ

ВВОД value

ПОКА ВСЕ

* + - 1. j++
      2. ПОКА ВСЕ
    1. i++
    2. ПОКА ВСЕ
  1. i++
  2. ПОКА ВСЕ

1. ВОЗВРАТ N

Функция UniqueMatrixChar:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Входные данные. | | | Выходные данные. | | |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| matrix | Матрица с заданными значениями | Целые числа | new\_matrix | Матрица с проверенными значениями элементов | Целые числа |
| size1 | Количество строк матрицы | Целые числа |  |  |  |
| size2 | Количество столбцов матрицы | Целые числа |  |  |  |

Словесное описание:

1. ЕСЛИ (matrix.Length МЕНЬШЕ ИЛИ РАВНО 0) ИЛИ (matrix[0].Length МЕНЬШЕ ИЛИ РАВНО 0) ИЛИ (matrix.Length НЕ РАВНО size1) ИЛИ (matrix[0].Length НЕ РАВНО size1)
2. ВОЗВРАТ null
3. ЕСЛИ ВСЕ
4. [][] new\_matrix = new matrix [size1][]
5. i=0
6. ПОКА i < size1 ВЫПОЛНИТЬ
   1. new\_matrix[i] = new matrix [size2]
   2. СКОПИРОВАТЬ ЗНАЧЕНИЯ ЭЛЕМЕНТОВ МАССИВА matrix [i] В new\_matrix[i]
7. ПОКА ВСЕ
8. r=0
9. ПОКА r < size1 ВЫПОЛНИТЬ
   1. i=0
   2. ПОКА i < size2 ВЫПОЛНИТЬ
      1. value1 –ЦЕЛОЕ
      2. value1= new\_matrix[r][i]
      3. value2 = new\_matrix[size1-1][i]
      4. k=0
      5. ПОКА k < i ВЫПОЛНИТЬ
         1. ЕСЛИ new\_matrix [r][k] РАВНО value1
            1. new\_matrix [r][k]= Случайное число в диапозоне (0..9)
            2. new\_matrix [r][i] = Случайное число в диапозоне (0..9)
         2. ЕСЛИ ВСЕ
         3. ЕСЛИ new\_matrix[size1-1] [k] РАВНО value2
            1. new\_matrix[size1-1][k] = Случайное число в диапозоне (0..9)
            2. new\_matrix[size1-1][i] = Случайное число в диапозоне (0..9)
         4. ЕСЛИ ВСЕ
      6. k++
      7. ПОКА ВСЕ
      8. k=(i+1)
      9. ПОКА k<size2 ВЫПОЛНИТЬ
         1. ЕСЛИ new\_matrix[r][k] РАВНО value1
            1. new\_matrix[r][k] = Случайное число в диапозоне (0..9)
            2. new\_matrix[r][i] = Случайное число в диапозоне (0..9)
         2. ЕСЛИ ВСЕ
         3. ЕСЛИ new\_matrix[size1-1][k] РАВНО value2
            1. new\_matrix[size1-1][k] = Случайное число в диапозоне (0..9)
            2. new\_matrix[size1-1][i] = Случайное число в диапозоне (0..9)
         4. ЕСЛИ ВСЕ
      10. k++
      11. ПОКА ВСЕ
   3. i++
   4. ПОКА ВСЕ
10. r++
11. ПОКА ВСЕ
12. i=1
13. ПОКА i<size1 ВЫПОЛНИТЬ
    1. j=0
    2. ПОКА j<size2 ВЫПОЛНИТЬ
       1. k=0
       2. ПОКА k<i ВЫПОЛНИТЬ
          1. p=0
          2. ПОКА p<size2 ВЫПОЛНИТЬ
             1. ЕСЛИ new\_matrix[k][p] РАВНО new\_matrix[i][j]

new\_matrix [k][p] = Случайное число в диапозоне (0..9)

new\_matrix [i][j] = Случайное число в диапозоне (0..9)

* + - * 1. ЕСЛИ ВСЕ
      1. p++
      2. ПОКА ВСЕ
    1. k++
    2. ПОКА ВСЕ
  1. j++
  2. ПОКА ВСЕ

1. i++
2. ПОКА ВСЕ
3. ВОЗВРАТ new\_matrix

Функция UniqueMatrix:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Входные данные. | | | Выходные данные. | | |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| matrix | Матрица с заданными значениями | Целые числа | new\_matrix | Матрица с проверенными значениями элементов | Целые числа |
| size1 | Количество строк матрицы | Целые числа |  |  |  |
| size2 | Количество столбцов матрицы | Целые числа |  |  |  |

Словесное описание:

1. ЕСЛИ (matrix.Length МЕНЬШЕ ИЛИ РАВНО 0) ИЛИ (matrix[0].Length МЕНЬШЕ ИЛИ РАВНО 0) ИЛИ (matrix.Length НЕ РАВНО size1) ИЛИ (matrix[0].Length НЕ РАВНО size1)
2. ВОЗВРАТ null
3. ЕСЛИ ВСЕ
4. [][] new\_matrix = new matrix [size1][]
5. i=0
6. ПОКА i < size1 ВЫПОЛНИТЬ
   1. new\_matrix[i] = new matrix [size2]
   2. СКОПИРОВАТЬ ЗНАЧЕНИЯ ЭЛЕМЕНТОВ МАССИВА matrix [i] В new\_matrix[i]
7. ПОКА ВСЕ
8. r=0
9. ПОКА r < size1 ВЫПОЛНИТЬ
   1. i=0
   2. ПОКА i < size2 ВЫПОЛНИТЬ
      1. value1 –ЦЕЛОЕ
      2. value1= new\_matrix[r][i]
      3. value2 = new\_matrix[size1-1][i]
      4. k=0
      5. ПОКА k < i ВЫПОЛНИТЬ
         1. ЕСЛИ new\_matrix [r][k] РАВНО value1
            1. new\_matrix [r][k]= 0
            2. new\_matrix [r][i] = 0
         2. ЕСЛИ ВСЕ
         3. ЕСЛИ new\_matrix[size1-1] [k] РАВНО value2
            1. new\_matrix[size1-1][k] = 0
            2. new\_matrix[size1-1][i] = 0
         4. ЕСЛИ ВСЕ
      6. k++
      7. ПОКА ВСЕ
      8. k=(i+1)
      9. ПОКА k<size2 ВЫПОЛНИТЬ
         1. ЕСЛИ new\_matrix[r][k] РАВНО value1
            1. new\_matrix[r][k] = 0
            2. new\_matrix[r][i] = 0
         2. ЕСЛИ ВСЕ
         3. ЕСЛИ new\_matrix[size1-1][k] РАВНО value2
            1. new\_matrix[size1-1][k] = 0
            2. new\_matrix[size1-1][i] = 0
         4. ЕСЛИ ВСЕ
      10. k++
      11. ПОКА ВСЕ
   3. i++
   4. ПОКА ВСЕ
10. r++
11. ПОКА ВСЕ
12. i=1
13. ПОКА i<size1 ВЫПОЛНИТЬ
    1. j=0
    2. ПОКА j<size2 ВЫПОЛНИТЬ
       1. k=0
       2. ПОКА k<i ВЫПОЛНИТЬ
          1. p=0
          2. ПОКА p<size2 ВЫПОЛНИТЬ
             1. ЕСЛИ new\_matrix[k][p] РАВНО new\_matrix[i][j]

new\_matrix [k][p] = 0

new\_matrix [i][j] = 0

* + - * 1. ЕСЛИ ВСЕ
      1. p++
      2. ПОКА ВСЕ
    1. k++
    2. ПОКА ВСЕ
  1. j++
  2. ПОКА ВСЕ

1. i++
2. ПОКА ВСЕ
3. ВОЗВРАТ new\_matrix

Функция OutputArray:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Входные данные. | | | Выходные данные. | | |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| N | Матрица с заданными значениями элементами | Целые числа |  |  |  |
| size1 | Количество строк матрицы | Целые числа |  |  |  |
| size2 | Количество столбцов матрицы | Целые числа |  |  |  |

Словесное описание:

1. i=0
2. ПОКА i < size1 ВЫПОЛНИТЬ
   1. j=0
   2. ПОКА j < size 2
      1. ВЫВОД МАТРИЦЫ
   3. j++
   4. ПОКА ВСЕ
3. i++
4. ПОКА ВСЕ

Функция UserChoiceYes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Входные данные. | | | Выходные данные. | | |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| size1 | Количество строк матрицы | Целые числа |  |  |  |
| size2 | Количество столбцов матрицы | Целые числа |  |  |  |
| m | Параметр |  |  |  |  |

Словесное описание алгоритма:

1. ЕСЛИ m = Null
   1. А = Вызов функции InputMatr(size1, size2)
   2. Вызов функции OutputArray(A, size1, size2)
   3. В =Вызов функции UniqueMatrix(A, size1, size2)
2. ЕСЛИ m = Сhar
   1. А = Вызов функции InputMatr(size1, size2)
   2. Вызов функции OutputArray(A, size1, size2)
   3. В =Вызов функции UniqueMatrixChar(A, size1, size2)

ИНАЧЕ

1. Вывод сообщения:("Введите корректный 4 параметр!")

Функция UserChoiceNo:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Входные данные. | | | Выходные данные. | | |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| size1 | Количество строк матрицы | Целые числа |  |  |  |
| size2 | Количество столбцов матрицы | Целые числа |  |  |  |
| m | Параметр |  |  |  |  |

Словесное описание алгоритма:

1. ЕСЛИ m = Null
   1. А = Вызов функции InputRandomMatr(size1, size2)
   2. Вызов функции OutputArray(A, size1, size2)
   3. В =Вызов функции UniqueMatrix(A, size1, size2)
2. ЕСЛИ m = Сhar
   1. А = Вызов функции InputRandomMatr(size1, size2)
   2. Вызов функции OutputArray(A, size1, size2)
   3. В =Вызов функции UniqueMatrixChar(A, size1, size2)

ИНАЧЕ

1. Вывод сообщения:("Введите корректный 4 параметр!")

Функция InputRandomMatr:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Входные данные. | | | Выходные данные. | | |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| size1 | Количество строк матрицы | Целые числа | N | Рандомно сгенерированная  матрица |  |
| size2 | Количество столбцов матрицы | Целые числа |  |  |  |

Словесное описание:

Объявление переменной rnd = Random()

1. i=0
2. ПОКА i < size1 ВЫПОЛНИТЬ
   1. N[ i ] = [m]
   2. i=0
   3. ПОКА i < size1 ВЫПОЛНИТЬ
      1. j=0
      2. ПОКА j < m ВЫПОЛНИТЬ
         1. [] array = Enumeravle.Range (0, size1 \* size2).OrderBy (x => rnd.Next()).ToArray()
         2. k=0
         3. ПОКА k < size1
            1. r=0
            2. ПОКА r < size2

N[i][j] = array[k \* size2 + r] - array[k \* size2]

* + - * 1. r++
        2. ПОКА ВСЕ
      1. k++
      2. ПОКА ВСЕ
    1. j++
    2. ПОКА ВСЕ
  1. i++
  2. ПОКА ВСЕ

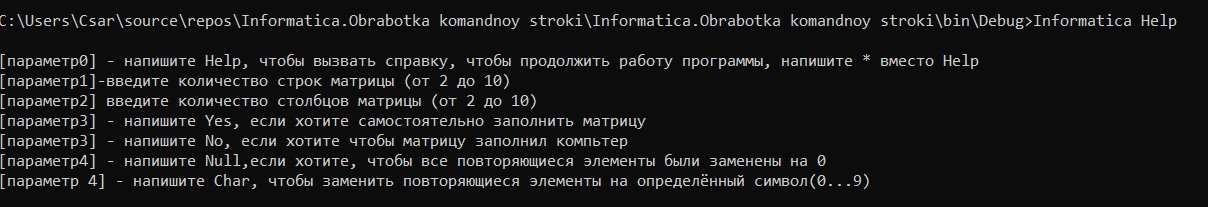
1. i++
2. ПОКА ВСЕ

Тесты.

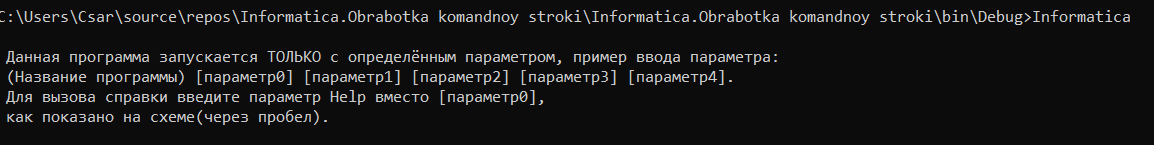
|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение теста | Входные данные | Выходные данные |
| 1 | Проверка правильности ввода args [1] | V | Сообщение("Введите корректный 1 параметр") |
| 2 | Проверка правильности ввода args [2] | 0 | Сообщение("Введите корректный 2 параметр") |
| 3 | Проверка правильности ввода args [3] | -1 | Сообщение("Введите корректный 3 параметр") |
| 3 | Проверка правильности ввода args [4] | ! | Сообщение("Введите корректный 4 параметр") |

Скриншоты выполненной работы:

Тест вызова справки:

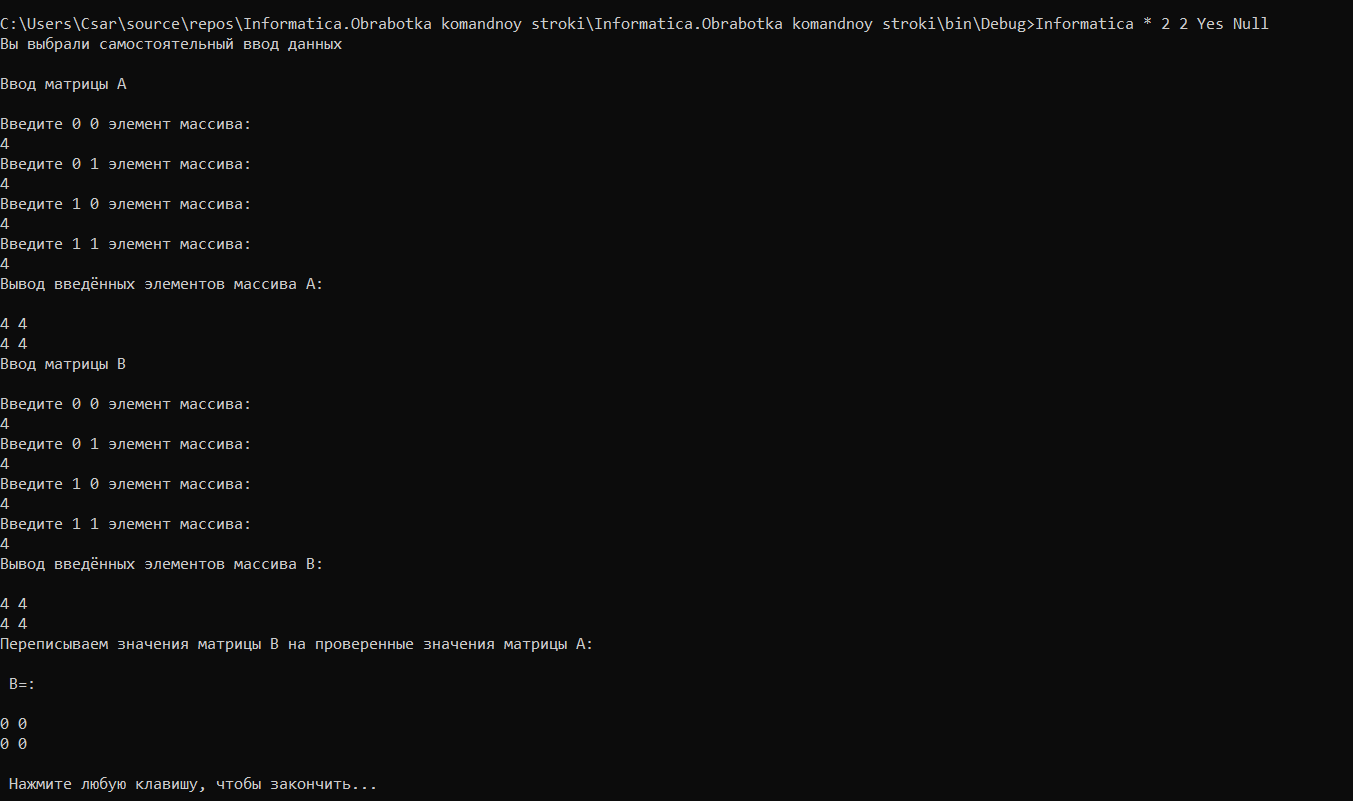


Тест ввода запуска программы без параметров:



Тест запуска: при args[0] = \*, args[1] = 2, args[2] = 2,args[3] = Yes

args[4] = Null



Тест запуска: при args[0] = \*, args[1] = 2, args[2] = 2,args[3] = Yes

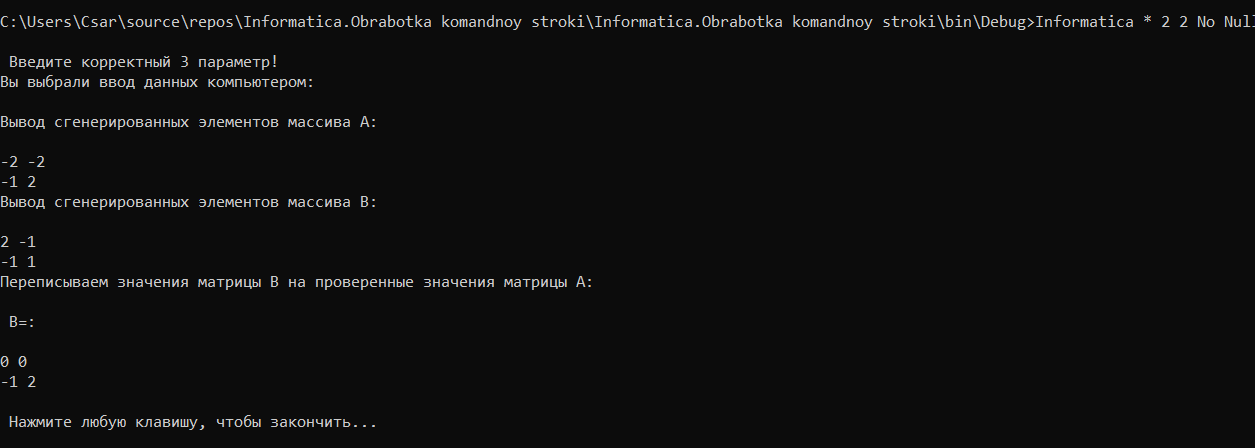
args[4] = Char

(повторяющиеся элементы заменились на 3)



Тест запуска: при args[0] = \*, args[1] = 2, args[2] = 2,args[3] = No

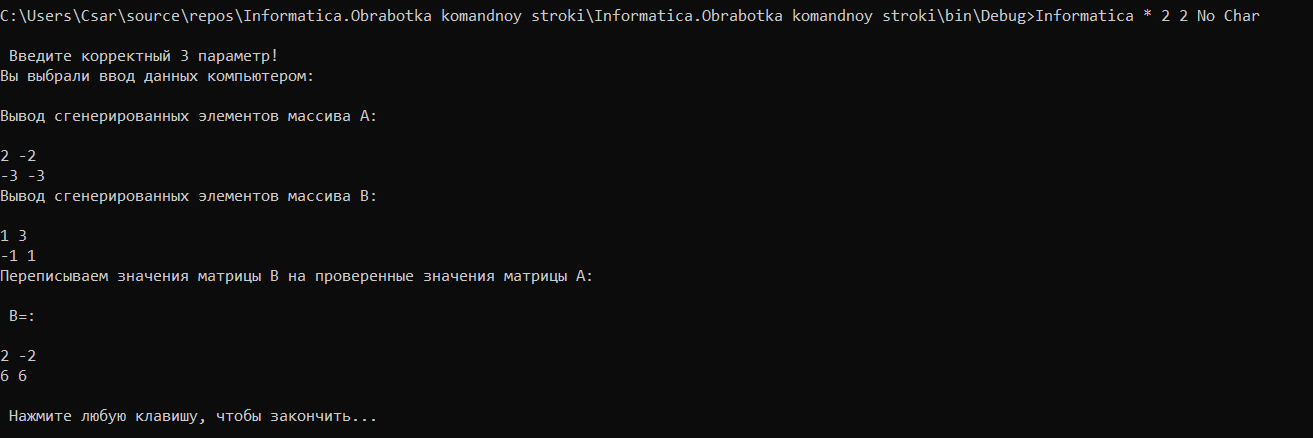
args[4] = Null



Код выдает ложное предупреждение, это связано с привязкой к одному аргументу 2 значений. Как мы видим, код прекрасно справился с задачей, заведомо обработав входные параметры.

Тест запуска: при args[0] = \*, args[1] = 2, args[2] = 2,args[3] = No

args[4] = Char



Текст программы:

using System;

using System.Linq;

namespace Лабораторная\_работа\_по\_информатике.\_Обработка\_командной\_строки

{

class Program

{

static Random rnd = new Random();

static int[][] InputRandomMatr(int size1, int size2)

{

int[][] N = new int[size1][];

for (int i = 0; i < size1; i++)

N[i] = new int[size2];

for (int i = 0; i < size1; i++)

for (int j = 0; j < size2; j++)

{

int[] array = Enumerable.Range(0, size1 \* size2).OrderBy(x => rnd.Next()).ToArray();

for (int k = 0; k < size1; k++)

for (int r = 0; r < size2; r++)

N[i][j] = array[k \* size2 + r] - array[k \* size2];

}

return N;

}

static int[][] InputMatr(int size1, int size2)

{

int[][] N = new int[size1][];

for (int i = 0; i < size1; i++)

N[i] = new int[size2];

for (int i = 0; i < size1; i++)

for (int j = 0; j < size2; j++)

{

int value;

Console.WriteLine($"Введите {i} {j} элемент массива: ");

while (!int.TryParse(Console.ReadLine(), out value))

Console.WriteLine("Введите корректные данные");

N[i][j] = value;

}

return N;

}

public static int[][] UniqueMatrix(int[][] matrix, int size1, int size2)

{

if ((matrix.Length <= 0) || (matrix[0].Length <= 0)

|| (matrix.Length != size1) || (matrix[0].Length != size2))

return null;

int[][] new\_matrix = new int[size1][];

for (int i = 0; i < size1; i++)

{

new\_matrix[i] = new int[size2];

matrix[i].CopyTo(new\_matrix[i], 0);

}

for (int r = 0; r < size1; r++)

{

for (int i = 0; i < size2; i++)

{

int value1 = new\_matrix[r][i],

value2 = new\_matrix[(size1 - 1)][i];

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

{

if (new\_matrix[r][k] == value1)

{

new\_matrix[r][k] = 0;

new\_matrix[r][i] = 0;

}

if (new\_matrix[(size1 - 1)][k] == value2)

{

new\_matrix[(size1 - 1)][k] = 0;

new\_matrix[(size1 - 1)][i] = 0;

}

}

for (int k = (i + 1); k < size2; k++)

{

if (new\_matrix[r][k] == value1)

{

new\_matrix[r][k] = 0;

new\_matrix[r][i] = 0;

}

if (new\_matrix[(size1 - 1)][k] == value2)

{

new\_matrix[(size1 - 1)][k] = 0;

new\_matrix[(size1 - 1)][i] = 0;

}

}

}

}

for (int i = 1; i < size1; i++)

{

for (int j = 0; j < size2; j++)

{

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

{

for (int p = 0; p < size2; p++)

if (new\_matrix[k][p] == new\_matrix[i][j])

{

new\_matrix[k][p] = 0;

new\_matrix[i][j] = 0;

}

}

}

}

return new\_matrix;

}

public static int[][] UniqueMatrixChar(int[][] matrix, int size1, int size2)

{

if ((matrix.Length <= 0) || (matrix[0].Length <= 0)

|| (matrix.Length != size1) || (matrix[0].Length != size2))

return null;

int[][] new\_matrix = new int[size1][];

for (int i = 0; i < size1; i++)

{

new\_matrix[i] = new int[size2];

matrix[i].CopyTo(new\_matrix[i], 0);

}

for (int r = 0; r < size1; r++)

{

for (int i = 0; i < size2; i++)

{

int value1 = new\_matrix[r][i],

value2 = new\_matrix[(size1 - 1)][i];

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

{

if (new\_matrix[r][k] == value1)

{

new\_matrix[r][k] = (new Random()).Next(0, 10);

new\_matrix[r][i] = (new Random()).Next(0, 10);

}

if (new\_matrix[(size1 - 1)][k] == value2)

{

new\_matrix[(size1 - 1)][k] = (new Random()).Next(0, 10);

new\_matrix[(size1 - 1)][i] = (new Random()).Next(0, 10);

}

}

for (int k = (i + 1); k < size2; k++)

{

if (new\_matrix[r][k] == value1)

{

new\_matrix[r][k] = (new Random()).Next(0, 10);

new\_matrix[r][i] = (new Random()).Next(0, 10);

}

if (new\_matrix[(size1 - 1)][k] == value2)

{

new\_matrix[(size1 - 1)][k] = (new Random()).Next(0, 10);

new\_matrix[(size1 - 1)][i] = (new Random()).Next(0, 10);

}

}

}

}

for (int i = 1; i < size1; i++)

{

for (int j = 0; j < size2; j++)

{

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

{

for (int p = 0; p < size2; p++)

if (new\_matrix[k][p] == new\_matrix[i][j])

{

new\_matrix[k][p] = 0;

new\_matrix[i][j] = 0;

}

}

}

}

return new\_matrix;

}

static void UserChoiceYes(string m, int size1, int size2)

{

if (m == "Null")

{

Console.WriteLine("Вы выбрали самостоятельный ввод данных\n");

Console.WriteLine($"Ввод матрицы А \n");

int[][] A = InputMatr(size1, size2);

Console.WriteLine("Вывод введённых элементов массива A:\n ");

OutputArray(A, size1, size2);

int[][] B = new int[size1][];

B = UniqueMatrix(A, size1, size2);

Console.WriteLine("Переписываем значения матрицы В на проверенные значения матрицы А:\n");

Console.WriteLine(" В=:\n");

OutputArray(B, size1, size2);

Console.WriteLine("\n Нажмите любую клавишу, чтобы закончить...");

}

if (m == "Char")

{

Console.WriteLine("Вы выбрали самостоятельный ввод данных\n");

Console.WriteLine($"Ввод матрицы А \n");

int[][] A = InputMatr(size1, size2);

Console.WriteLine("Вывод введённых элементов массива A:\n ");

OutputArray(A, size1, size2);

int[][] B = new int[size1][];

B = UniqueMatrixChar(A, size1, size2);

Console.WriteLine("Переписываем значения матрицы В на проверенные значения матрицы А:\n");

Console.WriteLine(" В=:\n");

OutputArray(B, size1, size2);

Console.WriteLine("\n Нажмите любую клавишу, чтобы закончить...");

}

else

{

Console.WriteLine("\n Введите корректный 4 параметр!");

}

}

static void UserChoiceNo(string m, int size1, int size2)

{

if (m == "Null")

{

Console.WriteLine("Вы выбрали ввод данных компьютером:\n");

int[][] A = InputRandomMatr(size1, size2);

Console.WriteLine("Вывод сгенерированных элементов массива A:\n ");

OutputArray(A, size1, size2);

int[][] B = new int[size1][];

B = UniqueMatrix(A, size1, size2);

Console.WriteLine("Переписываем значения матрицы В на проверенные значения матрицы А:\n");

Console.WriteLine(" В=:\n");

OutputArray(B, size1, size2);

Console.WriteLine("\n Нажмите любую клавишу, чтобы закончить...");

}

if (m == "Char")

{

Console.WriteLine("Вы выбрали ввод данных компьютером:\n");

int[][] A = InputRandomMatr(size1, size2);

Console.WriteLine("Вывод сгенерированных элементов массива A:\n ");

OutputArray(A, size1, size2);

int[][] B = new int[size1][];

B = UniqueMatrixChar(A, size1, size2);

Console.WriteLine("Переписываем значения матрицы В на проверенные значения матрицы А:\n");

Console.WriteLine(" В=:\n");

OutputArray(B, size1, size2);

Console.WriteLine("\n Нажмите любую клавишу, чтобы закончить...");

}

else

{

Console.WriteLine("\n Введите корректный 4 параметр!");

}

}

static void OutputArray(int[][] N, int size1, int size2)

{

for (int i = 0; i < size1; i++)

{

for (int j = 0; j < size2; j++)

{

Console.Write(N[i][j].ToString() + " ");

}

Console.WriteLine();

}

}

static void Main(string[] args)

{

int size1 = 0, size2 = 0;

if (args.Length > 0)

{

if (args[0] == "Help")

{

Console.WriteLine("\n[параметр0] - напишите Help, чтобы вызвать справку, чтобы продолжить работу программы, напишите \* вместо Help \n[параметр1]-введите количество строк матрицы (от 2 до 10) \n[параметр2] введите количество столбцов матрицы (от 2 до 10)\n[параметр3] - напишите Yes, если хотите самостоятельно заполнить матрицу \n[параметр3] - напишите No, если хотите чтобы матрицу заполнил компьтер \n[параметр4] - напишите Null,если хотите, чтобы все повторяющиеся элементы были заменены на 0 \n[параметр 4] - напишите Char, чтобы заменить повторяющиеся элементы на определённый символ(0...9)\n");

}

if (args[0] == "\*")

{

size1 = Convert.ToInt32(args[1]);

size2 = Convert.ToInt32(args[2]);

if ((size1 > 0) || (size2 > 0))

{

if (args[3] == "Yes")

{

string m = args[4];

UserChoiceYes(m, size1, size2);

}

if (args[3] == "No")

{

string m = args[4];

UserChoiceNo(m, size1, size2);

}

else

{

Console.WriteLine("Введите корректный 3 параметр!");

}

}

else

{

Console.WriteLine("Ошибка: 1 и 2 параметры меньше или равны 0");

}

}

else

{

Console.WriteLine("Введите Help для вызова справки или \*");

}

}

else

{

Console.WriteLine("\n Данная программа запускается ТОЛЬКО с определённым параметром, пример ввода параметра:\n (Название программы) [параметр0] [параметр1] [параметр2] [параметр3] [параметр4].\n Для вызова справки введите параметр Help вместо [параметр0],\n как показано на схеме(через пробел).\n");

}

}

}

}