

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №3**

з дисципліни “Основи програмування”

тема “Масиви даних”

|  |  |  |
| --- | --- | --- |
| Виконав  студент I курсу  групи КП-01  Беліцький Олександр Сергійович  (*прізвище, ім’я, по батькові*)  варіант №3 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

Київ 2020

**Мета роботи**

Навчитися працювати зі масивами різних типів даних та розмірності.

Застосувати на практиці різні види циклічних конструкцій при роботі з одновимірними та багатовимірними масивами даних.

**Постановка завдання**

#### **Частина 1. Одновимірні масиви**

Дано масив цілих чисел, де кожне число - висота землі. Приклад:

-4, 4, -5, 0, 1, -4, 1, 0, 6, 0, -4

a) Створити новий масив цілих чисел і записати в нього зсунуні висоти відносно найменшого значення так, щоби воно стало нульовим, приклад:

1, 9, 0, 5, 6, 1, 6, 5, 11, 5, 1

b) Створти масив дійсних чисел такого ж розміру і записати в нього нормалізовані висоти зсунутого масиву відносно найбільшого значення так, щоби воно стало одиницею, а всі інші значення - дробами (приклад).

0.091, 0.818, 0.000, 0.455, 0.545, 0.091, 0.545, 0.455, 1.000, 0.455, 0.091

c) Користувач може задати число - висоту рівня води (від 0 до максимальної висоти включно). Створити одновимірний масив цілих чисел, що міститиме висоти стовпців води у кожній з секцій. Приклад, нехай рівень води 4, обчислення проводити відносно зсунутого масиву висот:

3, 0, 4, 0, 0, 3, 0, 0, 0, 0, 3

d) Нехай 1 клітинка - 1 м3 об'єму і 1 м2 площі поверхні. Знайти і вивести сумарний об'єм землі над рівнем води.

e) Реалізувати функцію, що

* виведе обриси землі (вид збоку) вписану в прямокутник від мінімальної (0) до максимальної висоти
* висоту кожного рівня
* покаже рівень води із заповненням порожнини під рівнем будь-яким спеціальним символом (або кольором фону).

#### **Частина 2. Двовимірні масиви**

Дано двовимірний масив значень 0 та 1 де 0 - вода, 1 - земля (вид зверху на групу островів). Приклад

{

{0, 1, 0, 0, 0, 1, 1, 0},

{0, 0, 1, 1, 1, 0, 0, 0},

{0, 1, 0, 0, 0, 1, 0, 1},

{0, 1, 1, 1, 1, 1, 1, 0},

{1, 0, 1, 0, 1, 0, 0, 1},

{0, 0, 1, 0, 1, 1, 0, 1},

{1, 1, 0, 0, 0, 1, 0, 0},

{0, 1, 0, 1, 1, 0, 1, 0},

{0, 1, 1, 0, 1, 0, 1, 1},

{0, 0, 0, 0, 0, 0, 0, 0},

{1, 0, 1, 0, 0, 1, 1, 0},

{0, 0, 1, 1, 1, 1, 1, 0},

{1, 1, 1, 1, 1, 1, 1, 1}

}

a) Знайти кількість островів.

b) Реалізувати функцію, що виводить у консоль вид на острови і воду зверху.

**Аналіз вимог і проектування**

#### **Частина 1**

Приклад виводу в першій частині завдання 1.e) (без рівня води):

------------- 12

| N | 11

| N | 10

| N N | 9

| N N | 8

| N N | 7

| N N N N | 6

| N NN NNNN | 5

| N NN NNNN | 4

| N NN NNNN | 3

| N NN NNNN | 2

|NN NNNNNNNN| 1

------------- 0

Приклад виводу в першій частині завдання (з рівнем води):

------------- 12

| N | 11

| N | 10

| N N | 9

| N N | 8

| N N | 7

| N N N N | 6

| N NN NNNN | 5

|~N~NN~NNNN~| 4 (water level)

|~N~NN~NNNN~| 3

|~N~NN~NNNN~| 2

|NN~NNNNNNNN| 1

------------- 0

#### **Частина 2**

2.a) Дві клітинки землі чи води належать одній частині якщо вони є сусідами зверху, знизу, зліва чи справа.

+--+

|10| <-- дві частини землі і дві частини води

|01|

+--+

+--+ +--+

|10| |11| <-- одна частина землі і одна частина води

|10| |00| у кожному з прикладів

+--+ +--+

Приклад виводу в другій частині завдання 2.c):

+--------+

| N NN |

| NNN |

| N N N|

| NNNNNN |

|N N N N|

| N NN N|

|NN N |

| N NN N |

| NN N NN|

| |

|N N NN |

| NNNNN |

|NNNNNNNN|

+--------+

**Тексти коду програм**

**Частина 1**

|  |
| --- |
| **Program.cs** |
| using System;  using static System.Console;  using static System.Math;  namespace part1  {  class Program  {  static void Main(string[] args)  {  WriteLine("Enter size of array: "); // n розмір  int n = int.Parse(ReadLine());  WriteLine("Enter water level: ");  int wlvl = int.Parse(ReadLine());  if (n > 150)  {  WriteLine("Size of array can't be more than 150");  }  else  {  int[] a = new int[n];  Random rnd = new Random();  int min = a[0];  for (int index = 0; index < a.Length; index++) //запис масиву випадковими числами + обчислення мінімального  {  a[index] = rnd.Next(-7, 7);  int item = a[index];  if (item < min)  {  min = item;  }  // Write(" {0}", a[index]);  }  // WriteLine();  int[] b = new int[n]; //нормалізований масив  double max = b[0];  for (int index = 0; index < a.Length; index++)  {  b[index] = a[index] - min;  double item = b[index];  if (item > max)  {  max = item;  }  // Write(" {0}", b[index]);  }  // WriteLine();  double[] c = new double[n]; // 0 - 1 масив  for (int index = 0; index < b.Length; index++)  {  c[index] = b[index] / max;  // Write(" {0:F3}", c[index]);  }  // WriteLine();  int[] waterlvl = new int[n]; //array of waterlevel  int sumoflan = 0;  for (int index = 0; index < b.Length; index++)  {  waterlvl[index] = b[index] - wlvl;  if (waterlvl[index] < 0)  {  waterlvl[index] = 0;  }  // Write(" {0}", waterlvl[index]);  sumoflan = sumoflan + waterlvl[index];  }  // WriteLine();  WriteLine("Volume of land above water: {0}", sumoflan);  Pic(n, max, b, a, min, wlvl);  }  }  static void Pic(int n, double max, int[] b, int[] a, int min, int wlvl )  {  for (int i = 0; i < n + 2; i++)  {  Write("-");  }  WriteLine(" {0}", max + 1);  double[,] la = new double[(int)max + 1, n + 1];  for (int i = (int)max; i >= 1; i += -1)  {  Write("|");  for (int index = 0; index < n; index++)  {  b[index] = a[index] - min;  if (b[index] >= i) // цей блок можна спростити, але залишив для зручності і кращого розуміння  {  la[i, index] = 1;  }  else  {  la[i, index] = 0;  }  if (la[i, index] == 1)  {  Write("N");  }  else  {  if (i <= wlvl)  {  Write("~");  }  else  {  Write(" ");  }  }  }  Write("|");  Write(" {0}", i);  if (wlvl == i)  {  Write(" (waterlevel)");  }  WriteLine();  }  for (int i = 0; i < n + 2; i++)  {  Write("-");  }  WriteLine(" 0");  }  }  } |

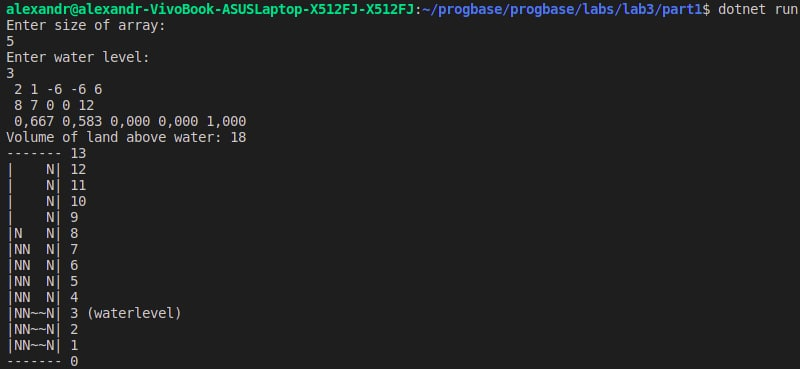
**Частина 2**

|  |
| --- |
| **Program.cs** |
| **using System;**  **using static System.Console;**  **using System.Linq;**  **namespace part2**  **{**  **class Program**  **{**  **static void Main(string[] args)**  **{**  **WriteLine("\*\*Enter size of territory\*\*");**  **Write("Enter length of territory: ");**  **string Str1 = ReadLine();**  **int n;**  **bool isNum1 = int.TryParse(Str1, out n);**  **if (isNum1)**  **{**  **if (n != 0)**  **{**  **Write("Enter width of territory: ");**  **string Str2 = ReadLine();**  **int m;**  **bool isNum2 = int.TryParse(Str2, out m);**  **if (isNum2)**  **{**  **if (m != 0)**  **{**  **WriteLine();**  **Random random = new Random();**  **int[,] a = new int[n, m];**  **for (int i = 0; i < n; i++)**  **{**  **for (int j = 0; j < m; j++)**  **{**  **a[i, j] = random.Next(0, 2);**  **Write(" {0}", a[i, j]);**  **}**  **WriteLine();**  **}**  **int[,] b = new int[a.GetLength(0), a.GetLength(1)];**  **Array.Copy(a, b, a.GetLength(0) \* a.GetLength(1));**  **int islands = 0;**  **foreach (int i in Enumerable.Range(0, a.GetLength(0)))**  **{**  **foreach (int j in Enumerable.Range(0, a.GetLength(1)))**  **{**  **if (Land(a, i, j))**  **islands++;**  **}**  **}**  **WriteLine("Number of islands is: {0}", islands);**  **Picture(b, n, m);**  **}**  **else**  **{**  **WriteLine("Width can't be 0");**  **}**  **}**  **else**  **{**  **WriteLine("Please, enter integer number");**  **}**  **}**  **else**  **{**  **WriteLine("Length can't be 0");**  **}**  **}**  **else**  **{**  **WriteLine("Please, enter integer number");**  **}**  **}**  **static bool Land(int[,] a, int i, int j)**  **{**  **if ((i < 0) || (i >= a.GetLength(0))) return false;**  **if ((j < 0) || (j >= a.GetLength(1))) return false;**  **bool island = a[i, j] == 1;**  **a[i, j] = 0;**  **if (island)**  **{**  **Land(a, i, j + 1);**  **Land(a, i, j - 1);**  **Land(a, i + 1, j);**  **Land(a, i - 1, j);**  **}**  **return island;**  **}**  **static void Picture(int[,] b, int n, int m)**  **{**  **for (int i = 0; i < m + 2; i++)**  **{**  **if ((i == 0) || (i == m + 1))**  **{**  **Write("+");**  **}**  **else**  **{**  **Write("-");**  **}**  **}**  **WriteLine();**  **for (int i = 0; i < n; i++)**  **{**  **Write("|");**  **for (int j = 0; j < m; j++)**  **{**  **if (b[i, j] == 1)**  **{**  **Write("N");**  **}**  **else**  **{**  **Write(" ");**  **}**  **}**  **Write("|");**  **WriteLine();**  **}**  **for (int i = 0; i < m + 2; i++)**  **{**  **if ((i == 0) || (i == m + 1))**  **{**  **Write("+");**  **}**  **else**  **{**  **Write("-");**  **}**  **}**  **WriteLine();**  **}**  **}**  **}** |

**Приклади результатів**

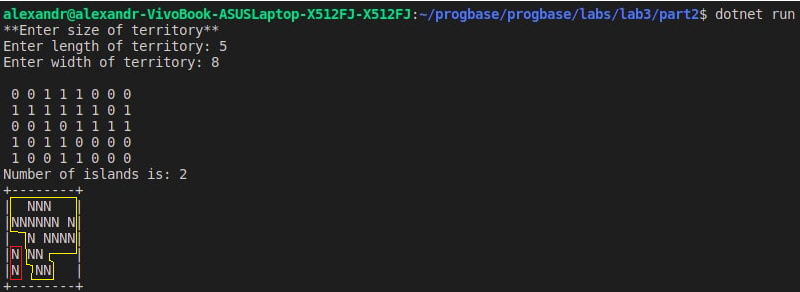
**Частина 1**

Вивід першої частини завдання (з рівнем води):

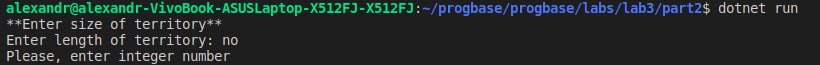


**Частина 2**

Вивід другої частини завдання. Виділені окремі острови.



Вивід при некоректному вводі даних:



**Висновки**

Виконавши дану лабораторну роботу було опрацьовано тему про масиви різних типів даних та розмірності. Застосувовано на практиці різні види циклічних конструкцій при роботі з одновимірними та багатовимірними масивами даних та функцій мови С#. Мова С# надає багато засобів роботи з масивами одного та баготьох вимірів.

Також, за допомогою різних типів циклів, можливо виконувати різні обрахунки на основі вхідних даних, звернення до елементів масиву, дії над кожним елементом масиву, що й було показано на практиці у завданнях 1 і 2.

Було встановлено, що цикл for більш гнучкий в порівнянні з foreach. Якщо foreach послідовно витягує елементи контейнера і тільки для читання, то в циклі for ми можемо перескакувати на кілька елементів вперед в залежності від збільшення лічильника, а також можемо змінювати елементи

Компіляція всього коду відбувалася за допомогою утиліти dotnet.