***Лабораторна робота №3****by Тукало Іван  
11 варіант*

*Завдання (блок) 1:*

Вставити в початок масиву мінімум з усіх значень масиву, а в кінець — максимум.

Слід зробити «максимально вручну», без складних бібліотечних методів (дозволені лише new, Length, квадратні дужки, і, можливо, Resize). Зокрема, прямо заборонено вживати в цьому блоці Array.ConvertAll, а також складну роботу з рядками де б не було, крім безпосередньо введення та виведення.

Якщо вже зроблено цими «максимально ручними» засобами, то, як додаткове завдання на додаткові бали можна зробити ще раз, максимально користуючись бібліотечними методами. Також на додаткові бали можна зробити ще раз, використавши замість масиву колекцію List і максимально користуючись методами колекції List, але теж лише після того, як зроблено масивами «максимально вручну».

Для цього завдання слід зробити і код (текст) програми, і блок-схеми (такі, як у 1-му семестрі; блок-схеми (багато) — в тому смислі, що для кожного методу окремо), і діаграми посилань (аналогічні наведеним у tutorial\_dyn\_arrays.pdf,

*Завдання (блок) 2, без варіантів:*

2а) Прочитайте єдине натуральне число n, спочатку сформуйте у пам'яті у вигляді зубчастого масиву з рядками різної довжини, потім виведіть на екран такі переліки: для кожного i від 0 до n–1, послідовність номер i містить, у порядку зростання, ті й тільки ті числа від 1 до n, які кратні сумі цифр числа i.

2б) Результат виведення на екран повинен бути абсолютно таким самим, як у попередньому пункті, але в пам'яті повинно бути значно менше різних «вкладених» одновимірних масивів, за рахунок того, що різні числа можуть мати однакову суму цифр, у цих випадках виходить однакова послідовність, і її можна тримати в пам'яті один раз.

Проведіть вимірювання технічними засобами обсягів використаної пам'яті в кожному з цих варіантів і проаналізуйте це своїми словами.

Для цього завдання слід зробити лише код (текст) програми та щойно згаданий аналіз. Ні блок-схем, ні діаграм посилань не треба.

*Завдання (блок) 3:*

Додати рядок після рядка, що містить максимальний елемент (якщо у різних місцях є кілька елементів з однаковим максимальним значенням, то брати перший з них).

На додаткові бали можна зробити також із використанням замість масиву колекції List; але зробити через зубчастий масив (звичайний масив звичайних масивів) теж треба, і спосіб з List-ом перевірятиметься й оцінюватиметься лише якщо зроблено через звичайний масив звичайних масивів.

Зробити діаграми посилань

*Завдання (блок) 4:*

Переписати з кожного рядка матриці P у відповідні рядки матриці Q лише непарні елементи. Відсортувати рядки отриманої матриці Q за зростанням. Сформувати з рядків матриці Q одновимірний масив та визначити ті його елементи, значення яких співпадають із власним індексом.

***КОД:***

***БЛОК – СХЕМА:***

***ДІАГРАМА ПОСИЛАНЬ:***

Do\_Block1 Початок

Бажаєте випадкове заповнення масиву? (так[1]/ні[2]):

int[] array

string userInput = Console.ReadLine().ToLower();

userInput == "1"

array = GenerateRandomArray()

Бажаєте вводити кожен елемент у окремому рядку? (так[1]/ні[2]):

userInput == "1"

array = InputArraySingleLine()

array = InputArraySeparateLines()

userInput = Console.ReadLine().ToLower();

+

-

+

-

\nВаш масив:

PrintArray(array)

TransformArray(ref array)

\nМасив після вставки мінімуму в початок та максимуму в кінець:

PrintArray(array)

Do\_Block1 Кінець

GenerateRandomArray()

Введіть кількість елементів масиву:

int length = int.Parse(Console.ReadLine())

Random random = new Random();

int[] array = new int[length];

Введіть мінімальне число масиву:

int min = int.Parse(Console.ReadLine())

Введіть максимальне число масиву:

int max = int.Parse(Console.ReadLine())

int i = 0; i < length; i++

array[i] = random.Next(min, max)

+

return array

Кінець

-

PrintArray(int[] array)

string.Join(", ", array)

Кінець

InputArraySeparateLines()

Введіть кількість елементів масиву:

int length = int.Parse(Console.ReadLine())

int[] array = new int[length]

Введіть елементи масиву по одному у кожному рядку:

int i = 0; i < length; i++

+

return array

$"Елемент {i + 1}: "

array[i] = int.Parse(Console.ReadLine())

-

Кінець

TransformArray(ref int[] array)

InputArraySingleLine()

Введіть елементи масиву, розділені пробілами або табуляціями:

string[] input = Console.ReadLine().Split(new char[] { ' ', '\t' }, StringSplitOptions.RemoveEmptyEntries)

int[] array = new int[input.Length]

int i = 0; i < input.Length; i++

+

return array

-

Кінець

array[i] = int.Parse(input[i])

array.Length == 0

+

-

return

int minValue = array[0];

int maxValue = array[0]

int i = 1; i < array.Length; i++

array[i] < minValue

+

-

minValue = array[i]

array[i] > maxValue

+

-

minValue = array[i]

+

Array.Resize(ref array, array.Length + 2)

-

Кінець

int i = array.Length - 1; i > 1; i--

array[i] = array[i - 2]

+

array[0] = minValue;

array[array.Length - 1] = maxValue;

-