**Алгоритми та структури даних. Основи алгоритмізації**

*Додаток 1*

Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний  інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації

і управління

Звіт

з лабораторної роботи № 8 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант\_\_\_21\_\_

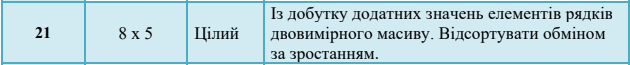
Виконав студент \_\_\_Мартаков Данило Кирилович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 (шифр, прізвище, ім'я, по батькові)

Перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 ( прізвище, ім'я, по батькові) 

Київ 2021

**Назва роботи:** Дослідження алгоритмів перетворення матриць та отримання агрегатних значень  
**Мета:** дослідити підходи до пошуку та перетворення на матрицях та набути практичних навичок використання укладених керувальних дій повторення і їх з’єднання під час складання програмних специфікацій. програмних специфікацій.  
**Варіант**: 21  
**Умова задачі**: 

Створити цілочисельний двовимірний масив, що складається з 8 рядків та 5 стовпців. Після створити цілочисельний одновимірний масив, елементами якого є добутки додатніх елементів кожного рядка двовимірного масиву.

**Постановка задачі:**

Для вирішення даної задачі нам необхідно створити двовимірний масив, який ми заповнимо рандомними значеннями. Після створення даної матриці ми маємо порахувати добутки всіх її рядків і створити одновимірний масив з цих добутків, який у наступному кроці маємо відсортувати від більшого значення до меншого.

**Побудова математичної моделі:** складемо таблицю імен змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| матриця | Двох вимірний масив чисел | matrix[,] | Двохвимірний масив 8 на 5 |
| масив | Масив чисел | dobArray | Масив добутків |
| Рандом | Random | rnd | рандом |
| добуток | Ціле число | dob | Допоміжне значення |
| Тимчасова зміна | Ціле число | temp | Допоміжне значення |
| Функція заповненя матриці випадковими числами | Функція | FillMatrix(matrix[,], row, colum) | Допоміжна Функція |
| Функція виводу значень матриці на екран | Функція | MatrixOutput(matrix[,], row, colum) | Допоміжна Функіця |
| Функція заповнення масиву добутків | Функція | FillDobArray(array[],matrix[,], row, colum) | Допоміжна Функція |
| Функція сортування масиву | Функція | SortArray(array[]) | Допоміжна Функція |
| Функція виводу значень масиву на екран | Функція | ArrayOutput(array[]) | Допоміжна Функція |

**Крок 1: Визначимо основні дії  
Крок 2: Деталізуємо основні зміні та визови функцій  
Крок 3: Деталізуємо тіло функції FillMatrix – заповнення матриці випадковими числами  
Крок 4: Деталізуємо тіло функції MatrixOutput – вивід значень матриці на екран  
Крок 5: Деталізуємо функцію FillDobArray – функція знаходження добутків додатних значень рядків і заповнення масиву цими значеннями.  
Крок 6: Деталізуємо функцію ArrayOutput – вивід значень масиву на екран  
Крок 7: Деталізуємо функцію SortArray – сортування масиву методом обміну за зростанням**

**Псевдокод**

Крок 1  
Початок

створення матриці matrix

заповнення matrix випадковими значеннями

вивід matrix

створення масиву dobArray

заповнення массиву dobArray

вивід масиву dobArray

сортування масиву dobArray

вивід масиву dobArray

Кінець

Крок 2  
Початок

matrix[8, 5]

FillMatrix(matrix, 8, 3)

MatrixOutput(matrix, 8, 3)

dobArray[8]

FillDobArray(dobArray, matrix, 8, 3)

ArrayOutput(dobArray)

SortArray(dobArray)

ArrayOutput(dobArray)

Кінець

Крок 3 (псевдокод функції FillMatrix(matrix[,], int row, int colum))

Початок

rnd

для i = 0; i < row; i++

для j = 0; j < colum; j++

matrix[i,j] - rnd.Next(-50, 50)

все для

все для

Кінець

Крок 4 (псевдокод функції MatrixOutput(matrix[,], int row, int colum))

Початок

для i = 0; i < row; i++

для j = 0; j < colum; j++

Вивести matrix[i, j] + "\t"

все для

Вивести пусту строку

все для

Кінець

Крок 5 (псевдокод функції FillDobArray(array[], matrix[m], int row, int colum))

для i = 0; i < row; i++

dob = 1

для j = 0; j < colum; j++

якщо matrix[i, j] > 0

dob \*= matrix[i, j]

все для

array[i] = dob

все для

Кінець

Крок 6 (псевдокодфункції ArrayOutput(array[]))

для i = 0; i < array.Length; i++

Вивести array[i] + "\t"

все для

Кінець

Крок 7 (псевдокод функції SortArray(array[]))

для i = 0; i < array.Length; i++

для j = 0; j < array.Length; j++

якщо array[j] > array[j + 1]

temp = array[j + 1]

array[j + 1] = array[j]

array[j] = temp

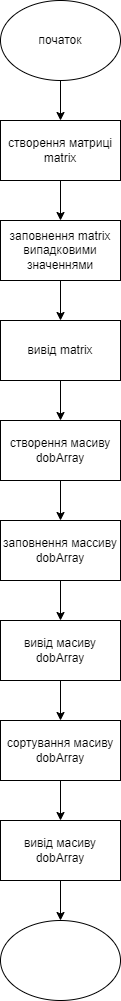
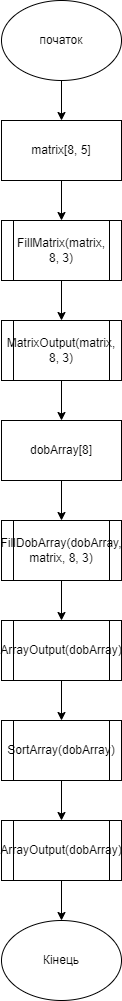
все якщо

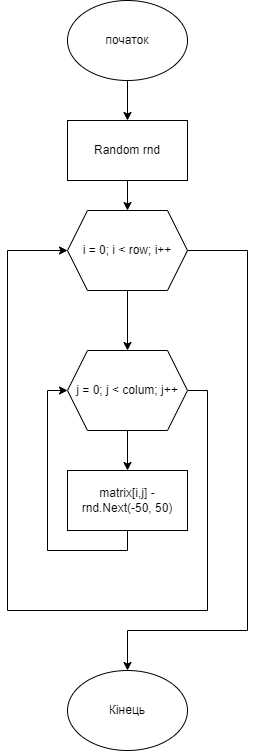
все для

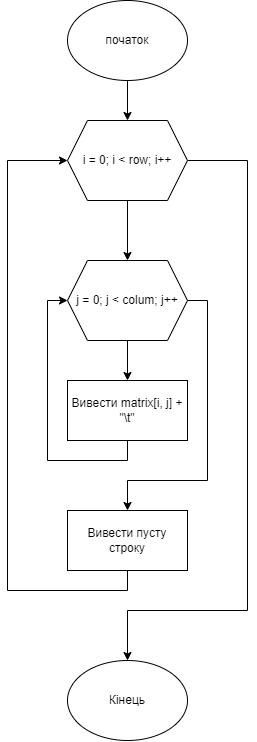
все для

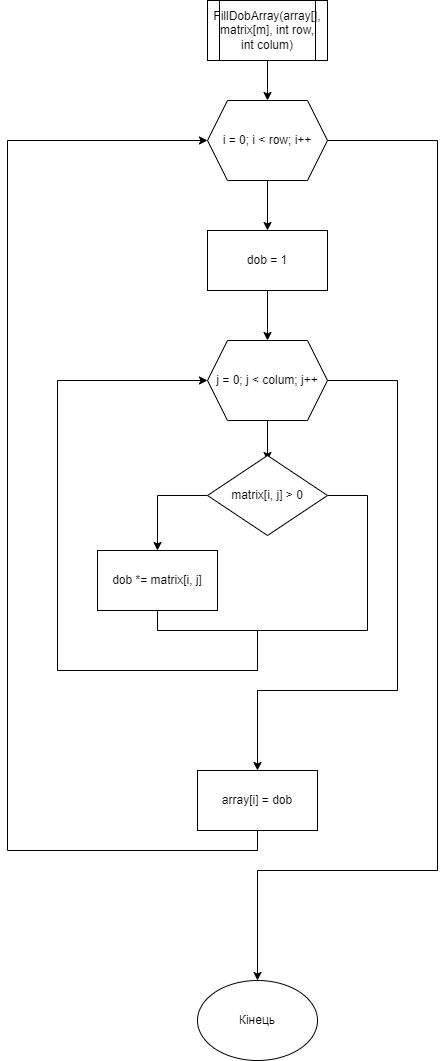
Кінець

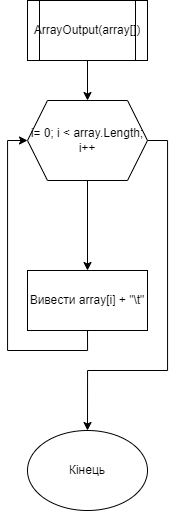
**Блок Схема**

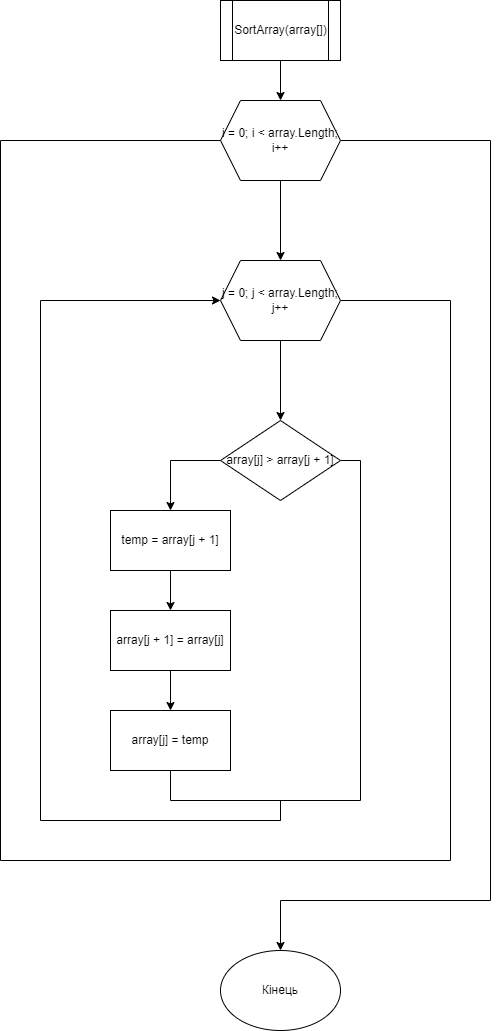
 











**Код Алгоритму**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace asdlab8

{

class Program

{

static void Main(string[] args)

{

int[,] matrix = new int[8, 5];

FillMatrix(matrix, 8, 3);

MatrixOutput(matrix, 8, 3);

int[] dobArray = new int[8];

FillDobArray(dobArray, matrix, 8, 3);

ArrayOutput(dobArray);

SortArray(dobArray);

ArrayOutput(dobArray);

Console.ReadLine();

}

public static void FillMatrix(int[,] matrix, int row, int colum)

{

Random rnd = new Random();

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

{

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

{

matrix[i, j] = rnd.Next(-50, 50);

}

}

}

public static void MatrixOutput(int[,] matrix, int row, int colum)

{

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

{

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

{

Console.Write(matrix[i, j] + "\t");

}

Console.WriteLine();

}

Console.WriteLine();

}

public static void FillDobArray(int[] array, int[,] matrix, int row, int colum)

{

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

{

int dob = 1;

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

{

if (matrix[i, j] > 0)

dob \*= matrix[i, j];

}

array[i] = dob;

}

}

public static void SortArray(int[] array)

{

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

{

for (var j = 0; j < array.Length - i; j++)

{

if (array[j] > array[j + 1])

{

int temp = array[j + 1];

array[j + 1] = array[j];

array[j] = temp;

}

}

}

}

public static void ArrayOutput(int[] array)

{

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

{

Console.Write(array[i] + "\t");

}

Console.WriteLine();

}

}

}

**Випробування алгоритму**



**Висновок**Було досліджено алгоритми пошуку та сортування, набуто практичних навичок використання цих алгоритмів під час складання програмних специфікацій на прикладі завдання, у якому потребувалося створення двовимірного масиву та виконання операцій створення одновимірного з певними умовами. Під час даної лабораторної роботи було побудовано математичну модель, створений як псевдокод, так і код на мові с#, блок-схема алгоритму та його тестування.