**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 № 7 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 лінійного пошуку в послідовностях  
**Мета:** дослідити методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій.  
**Варіант**: 21  
**Умова задачі**: 

1. Опису трьох змінних індексованого типу з 10 символьних значень.

2. Ініціювання двох змінних виразами згідно з варіантом (74 - i; 64 + 2 \* i ).

3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.

4. Обробки третьої змінної згідно з варіантом (сума двох мінімальних чисел ).

**Постановка задачі:**

Початкові дані - із початкових даних маємо лише кількість елементов масиву.

*За даними вимогами до формування елементів сгенерувати 3 массиви. Перші два масива - сгенерувати за даними формулами, а третій - з рівних елементів перших двох массивів. Після цього виконати дії над третім масивом, зазначені в умові (знайти суму непарних кодів елементів)*

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Перший масив | Масив чисел | a | масив з 10 символьних значень |
| Другий масив | Масив чисел | b | масив з 10 символьних значень |
| Третій масив | Масив чисел | c | масив символьних значень |
| Перший мінімальний елемент | Ціле число | firstMin | Допоміжне значення |
| Другий Мінімальний елемент | Ціле число | secondMin | Допоміжне значення |
| Функція ініцілізування перших двох масивів за умовою | Функція | InitArrays(a[], b[]) | Допоміжна Функція |
| Функція ініцілізування третього масива | Функція | InitThirdArray(a[], b[], c[]) | Допоміжна Функіця |
| Функція виводу значень масиву на екран | Функція | ArrayOutput(array[]) | Допоміжна Функція |
| Функція знаходження суми двох мінімальних елементів массиву |  | FindMinSum(array[]) | Допоміжна Функція |

**Крок 1: Визначимо основні дії  
Крок 2: Деталізуємо ввід початкових даних та ініціалізуемо усі 3 змінні  
Крок 3: Деталізуємо тіло функції InitArrays – ініціалізація перших двох змінних за умовою даною в задачі  
Крок 4: Деталізуємо тіло функції InitThirdArray – ініціалізація третьої змінної   
Крок 5: Деталізуємо функцію ArrayOutput – функція виводу значень масиву на екран.  
Крок 6: Деталізуємо функцію FindMinSum – функція яка знаходить суму двох мінімальних значень масиву.**

**Псевдокод**

Крок 1

Початок

Ініціалізація масиву a та b

Вивід значень масивів на екран

Розрахунок значення суми мінімальних значень c

Кінець

Крок 2

Початок

a[10], b[10], c[10]

InitArrays(a, b)

InitThirdArray(a, b, c)

ArrayOutput(a)

ArrayOutput(b)

ArrayOutput(c)

Вивести FindMinSum(c)

Крок 3 (псевдокод функції InitArrays(a[], b[]))

Початок

для i = 0; i < 10; i++

a[i] = 74 - i

b[i] = 64 + 2 \* i

все для

кінець

Крок 4 (псевдокод функції InitThirdArray(a[], b[], c[]))

Початок

для i = 0; i < 10; i++

j = 0

поки j < 10 && c[i] == 0

якщо a[i] == b[j]

c[i] = b[j]

все якщо

j++

все поки

все для

Кінець

Крок 5 (псевдокод функції ArrayOutput(array[]))

Початок

для i= 0; i < array.Length; i++

Вивести array[i] + "\t"

все для

Крок 6 (псевдокод функції FindMinSum(array[]))

Початок

firstMin = array[0]

secondMin = -1

для i = 1; i < array.Length; i++

якщо array[i] != 0

якщо firstMin > array[i]

secondMin = firstMin

firstMin = array[i]

все якщо

якщо secondMin != -1 && secondMin > array[i] && array[i] != firstMin

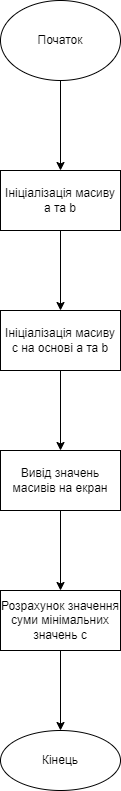
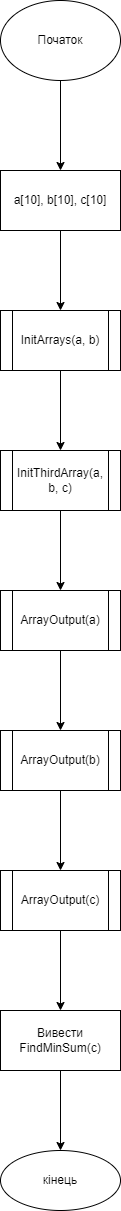
secondMin = array[i]

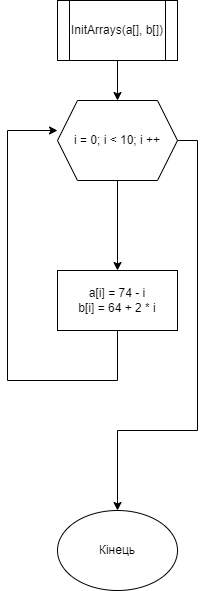
все якщо

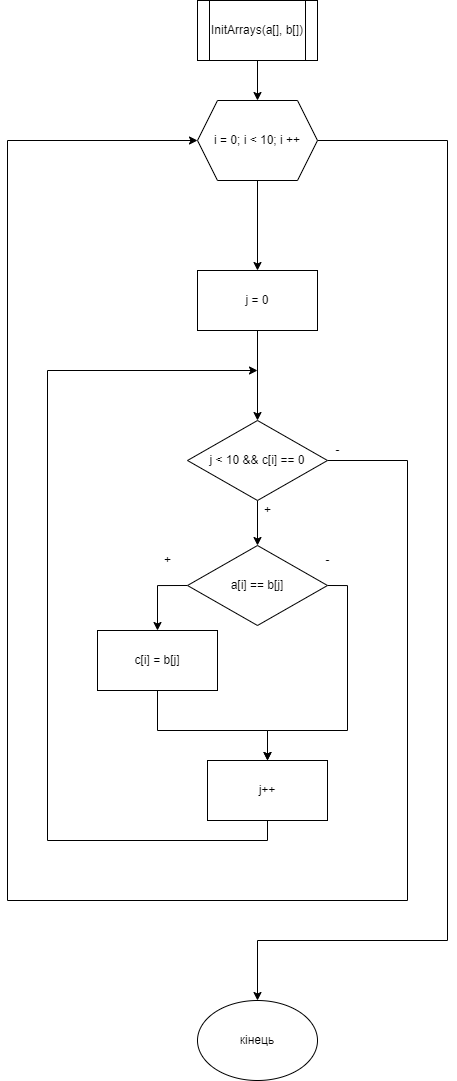
все якщо

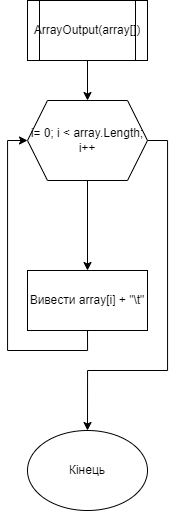
все для

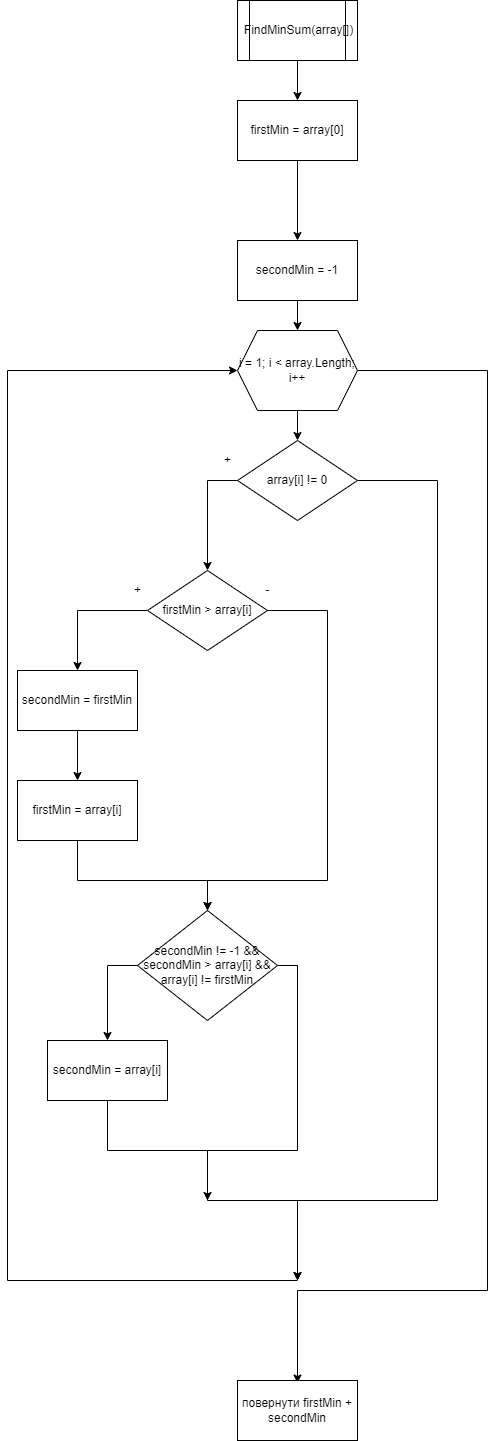
повернути firstMin + secondMin  
**Блок Схема**









**Код Алгоритму**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace asdlab7

{

class Program

{

static void Main(string[] args)

{

int[] a = new int[10];

int[] b = new int[10];

int[] c = new int[10];

InitArrays(a, b);

InitThirdArray(a, b, c);

ArrayOutput(a);

ArrayOutput(b);

ArrayOutput(c);

Console.WriteLine(FindMinSum(c));

Console.ReadLine();

}

public static void InitArrays(int[] a, int[] b)

{

for (int i = 0; i < 10; i++)

{

a[i] = 74 - i;

b[i] = 64 + 2 \* i;

}

}

public static void InitThirdArray(int[] a, int[] b, int[] c)

{

for (int i = 0; i < 10; i++)

{

int j = 0;

while (j < 10 && c[i] == 0)

{

if (a[i] == b[j])

c[i] = b[j];

j++;

}

}

}

public static void ArrayOutput(int[] array)

{

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

{

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

}

Console.WriteLine();

}

public static int FindMinSum(int[] array)

{

int firstMin = array[0];

int secondMin = -1;

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

{

if (array[i] != 0)

{

if (firstMin > array[i])

{

secondMin = firstMin;

firstMin = array[i];

}

if (secondMin != -1 && secondMin > array[i] && array[i] != firstMin)

secondMin = array[i];

}

}

return firstMin + secondMin;

}

}

}

**Випробування алгоритму**

