Міністерство освіти і науки України

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

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №7 з дисципліни «Алгоритми та структури даних-1. Основи алгоритмізації»

«Дослідження лінійного пошуку в послідовностях»

Варіант 11

Виконав студент ІП-12 Дулов Денис Валерійович

(шифр, прізвище, ім'я, по батькові)

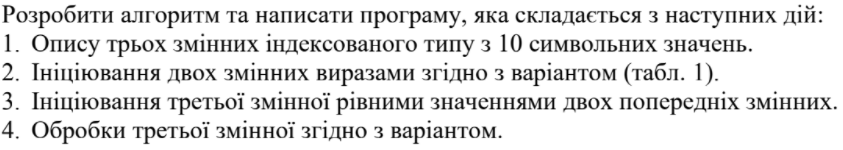
Перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( прізвище, ім'я, по батькові)

'

Київ 2021

**Мета** – дослідити методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій.

**Постановка задачі:**

**Математична модель**

|  |  |  |  |
| --- | --- | --- | --- |
| *Значення* | *Тип даних* | *Ім'я* | *Використання* |
| Масив з формулою елемента **2\*i+40** | *Індексований* | *Arr1* | *Проміжкове значення* |
| Масив з формулою елемента **52-2\*і** | *Індексований* | *Arr2* | *Проміжкове значення* |
| Третій масив | *Індексований* | *Arr3* | *Проміжне значення* |
| Розмір масивів | *Цілий* | *SIZE* | *Вхідне значення* |
| Елемент який має максимальний код | *Символьний* | *res* | *Вихідне значення* |
| Лічильник | *Цілий* | *k* | *Проміжне значення* |
| Функція, шо ініціює arr1 | *Фукнція* | *fillArray1(SIZE)* | *Допоміжна функція* |
| Функція, шо ініціює arr2 | *Функція* | *fillArray2(SIZE)* | *Допоміжна функція* |
| Функція, шо ініціює arr3 | *Функція* | *generateArray(arr1, arr2, SIZE, k)* | *Допоміжна функція* |
| Функція, що повертає максимальне значення масиву | *Фукнція* | getMax(arr3, k) | *Допоміжна функція* |

**Псевдокод**

*Крок 1*

**початок**

Ініціювання масивів arr1 та arr2

Ініціювання масиву arr3

Пошук елементу arr3 з найбільшим значенням коду

**кінець**

*Крок 2*

**початок**

arr1 = fillArray1(SIZE)

arr2 = fillArray2(SIZE)

Ініціювання масиву arr3

Пошук елементу arr3 з найбільшим значенням коду

**кінець**

*Крок 3*

**початок**

arr1 = fillArray1(SIZE)

arr2 = fillArray2(SIZE)

k = 0

arr3 = generateArray(arr1, arr2, SIZE, k)

Пошук елементу arr3 з найбільшим значенням коду

**кінець**

*Крок 4*

**початок**

arr1 = fillArray1(SIZE)

arr2 = fillArray2(SIZE)

k = 0

arr3 = generateArray(arr1, arr2, SIZE, k)

res = getMax(arr3, k)

**кінець**

Крок 5.

псевдокод фукнції fillArray1(SIZE)

**початок**

**повторити**

**для і від 0 до SIZE**

arr1[i] = 2\*i+40

**все повторити**

**повернути** arr1

Крок 6.

псевдокод фукнції fillArray2(SIZE)

**початок**

**повторити**

**для і від 0 до SIZE**

arr1[i] = 52-2\*і

**все повторити**

**повернути** arr2

Крок 7.

псевдокод фукнції generateArray(arr1, arr2, SIZE, k)

**початок**

**повторити**

**для і від 0 до SIZE**

**повторити для j від 0 до SIZE**

**якщо** arr1[i] == arr2[j]

**то**

**все якщо**

**все повторити**

**все повторити**

**повернути** arr3

*Крок 8*

псевдокод функції getMax(arr3, k)

max = arr3[0]

**повторити для і від 1 до k**

**якщо** arr[i] > max

**то**

max = arr3[i]

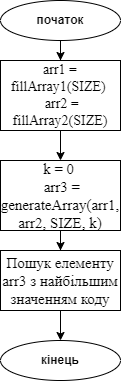
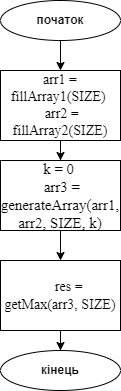
**все якщо**

**все повторити**

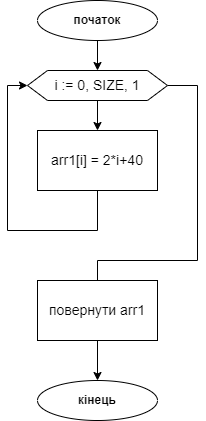
**повернути** max

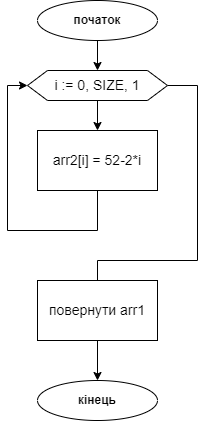
**Блок-схема**

*Крок 1 Крок 2 Крок 3 Крок 4*

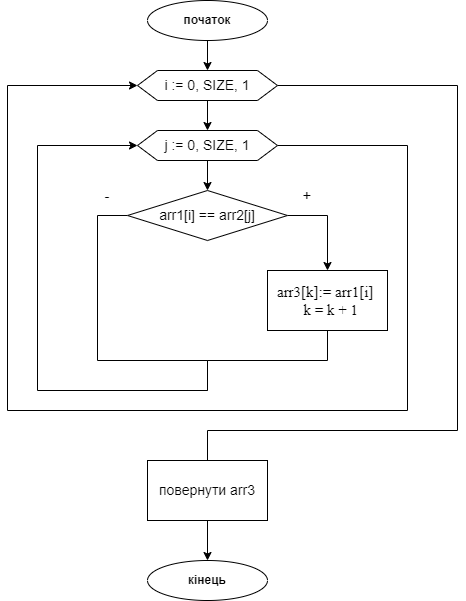


Крок 5 (блок-схема функції fillArray1(SIZE))

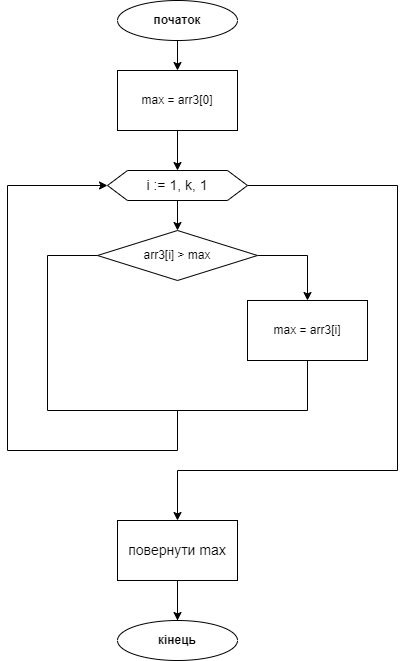


*Крок 6(блок-схема функції fillArray2(SIZE))*

*Крок 7 (блок-схема функції* generateArray(arr1, arr2, SIZE, k))



Крок 8 (блок-схема функції getMax(arr3, k))



*Код програми*

using System;

namespace Lab7

{

class Program

{

public static void Main()

{

const int SIZE = 10;

char[] arr1 = fillArray1(SIZE);

char[] arr2 = fillArray2(SIZE);

int len = 0;

char[] arr3 = generateArray(arr1, arr2, SIZE, len);

char res = getMax(arr3, SIZE);

Console.WriteLine("arr1: ");

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

{

Console.Write(arr1[i] + " ");

}

Console.WriteLine("\narr2: ");

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

{

Console.Write(arr2[i] + " ");

}

Console.WriteLine("\narr3: ");

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

{

Console.Write(arr3[i] + " ");

}

Console.WriteLine("\nmax element of arr3 is " + res);

}

static char[] fillArray1(int SIZE)

{

char[] arr = new char[SIZE];

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

{

arr[i] = (char)(2 \* i + 40);

}

return arr;

}

static char[] fillArray2(int SIZE)

{

char[] arr = new char[SIZE];

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

{

arr[i] = (char)(-2 \* i + 52);

}

return arr;

}

static int getLengthOfArr3(char[] arr1, char[] arr2, int SIZE)

{

int k = 0;

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

{

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

{

if(arr1[i].CompareTo(arr2[j]) == 0)

{

k++;

}

}

}

return k;

}

static char[] generateArray(char[] arr1, char[] arr2, int SIZE, int len)

{

char[] arr = new char[10];

int k = 0;

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

{

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

{

if(arr1[i].CompareTo(arr2[j]) == 0)

{

arr[k] = arr1[i];

k++;

}

}

}

return arr;

}

static char getMax(char[] arr, int len)

{

char max = arr[0];

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

{

if(arr[i] > max)

{

max = arr[i];

}

}

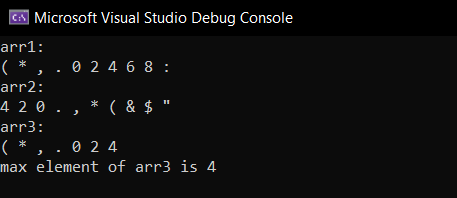
return max;

}

}

}

*2Тестування програми*



*Висновок*

*Під час лабораторної роботи було досліджено методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях і набуто практичних навичок їх використання під час складання програмних специфікацій.*