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 алгоритмів пошуку та сортування»

Варіант 9

Виконав студент ІП-12 Волошинівський Олександр

(шифр, прізвище, ім'я, по батькові)

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

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 7**

**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 9**

*1. Задача.* Знайти кількість елементів які діляться на 5 у послідовності яку ми ініціюємо з однакових елементів двох послідовностей ініційованих з виразів

i \* i + 76 та 85 - i.

*2. Постановка задачі.* Результатом розв’язку даної задачі є кількість елементів які діляться на 5 у послідовності яку ми ініціюємо з однакових елементів двох послідовностей ініційованих з виразів

i \* i + 76 та 85 - i.

*3. Побудова математичної моделі.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Послідовність 1 | Послідовність | A | Початкове дане\проміжкове дане |
| Послідовність 2 | Послідовність | B | Початкове дане\проміжкове дане |
| Послідовність 3 | Послідовність | C | Проміжкове дане |
| Індекс послідовності та циклу | Цілий | і | Початкове дане\проміжкове дане |
| Кількість членів послідовності які діляться на 5 | Цілий | K | Результат |
| Змінна яка контролює наповненість масиву | Цілий | index | Проміжкове дане |

Математичне формування задачі можна сформулювати як вивід кількості елементів які діляться на 5 у послідовності яку ми ініціюємо з однакових елементів двох послідовностей ініційованих з виразів

i \* i + 76 та 85 - i.

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

*Крок 1. Ініціалізація змінних та перших двох масивів згідно завданням*

*Крок 2. Виведення перших двох масивів на екран*

*Крок 3. Пошук рівних елементів перших двох масивів*

*Крок 4. Обрахування кількості елементів, коди яких діляться на 5 та виведення результатів на екран*

*4. Псевдокод алгоритму*

*Допоміжні функції:*

**Функція** poshuk(arr, len, element)

i=0

**поки** і<len

**якщо** arr[i]==element

**то**

повернути 1

**все якщо**

і=і+1

**все поки**

**повернути** 0

**все функція**

**Функція** vuvid(arr, len)

i=0

**поки** і<len

**виведення** arr[i]

i=i+1

**все поки**

**все функція**

**Функція** dilnik5(arr, len)

i=0, count=0

**поки** і<len

**якщо** arr[i]%5==0

**то**

count=count+1

**все якщо**

і=і+1

**все поки**

**повернути** count

**все функція**

*Крок 1*

**Початок**

SIZE = 10, result = 0

A[SIZE], B[SIZE, C[SIZE]

index\_c = 1, i = 1

**поки** i <= SIZE

A[i] = i \* i + 76

B[i] = 85 - i

i = i + 1

**все поки**

Виведення масивів

Ініціалізація третього масиву рівними

значеннями

Обрахування кількості

кодів які кратні 5 виведення масивів

**Кінець**

*Крок 2*

**Початок**

SIZE = 10, result = 0

A[SIZE], B[SIZE, C[SIZE]

index\_c = 1, i = 1

**поки** i <= SIZE

A[i] = i \* i + 76

B[i] = 85 - i

i = i + 1

**все поки**

vuvid(A, SIZE)

vuvid(B, SIZE)

Ініціалізація третього масиву рівними

значеннями

Обрахування кількості

кодів які кратні 5 виведення масивів

**Кінець**

*Крок 3*

**Початок**

SIZE = 10, result = 0

A[SIZE], B[SIZE, C[SIZE]

index\_c = 1, i = 1

**поки** i <= SIZE

A[i] = i \* i + 76

B[i] = 85 - i

i = i + 1

**все поки**

vuvid(A, SIZE)

vuvid(B, SIZE)

i = 1

**поки** i <= SIZE

**якщо** poshuk(A, SIZE, B[i])==1

**то**

C[index\_c] = B[i]

index\_c = index\_c + 1

**все якщо**

i = i + 1

**все поки**

vuvid(C, index\_c)

Обрахування кількості

кодів які кратні 5 виведення масивів

**Кінець**

*Крок 4*

**Початок**

SIZE = 10, result = 0

A[SIZE], B[SIZE, C[SIZE]

index\_c = 1, i = 1

**поки** i <= SIZE

A[i] = i \* i + 76

B[i] = 85 - i

i = i + 1

**все поки**

vuvid(A, SIZE)

vuvid(B, SIZE)

i = 1

**поки** i <= SIZE

**якщо** poshuk(A, SIZE, B[i])==1

**то**

C[index\_c] = B[i]

index\_c = index\_c + 1

**все якщо**

i = i + 1

**все поки**

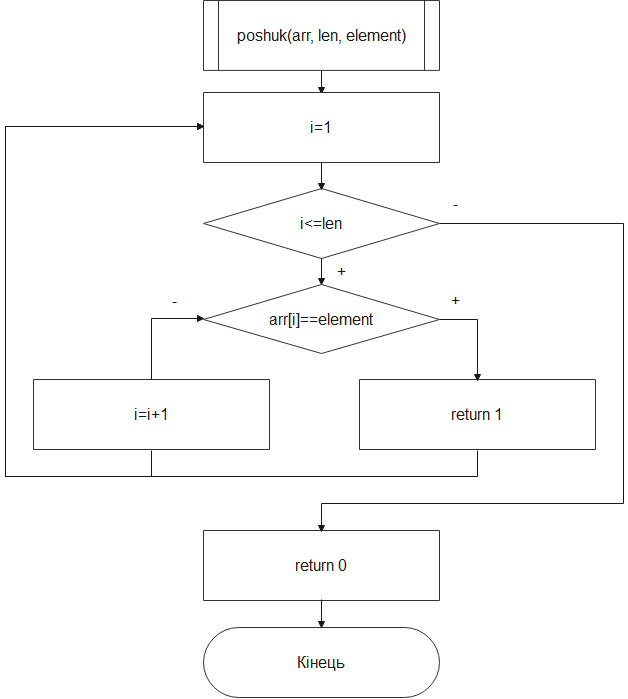
vuvid(C, index\_c)

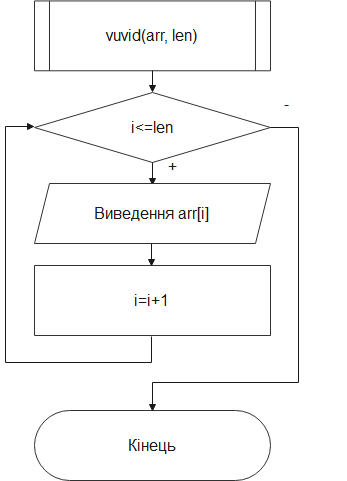
result = dilnik5(C, index\_c)

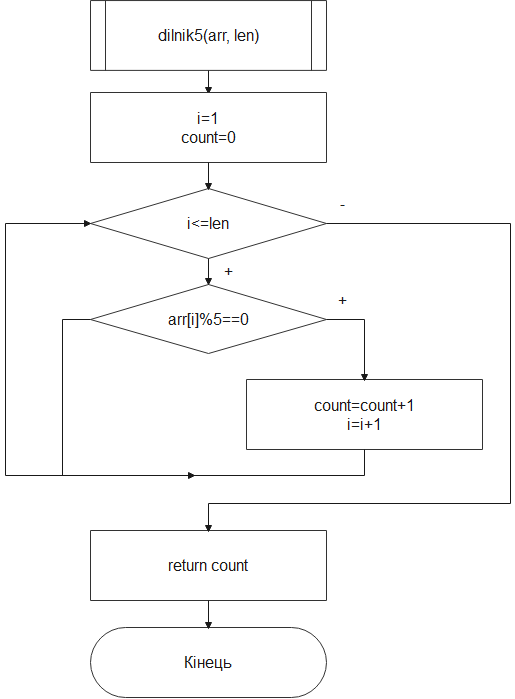
**виведення** result

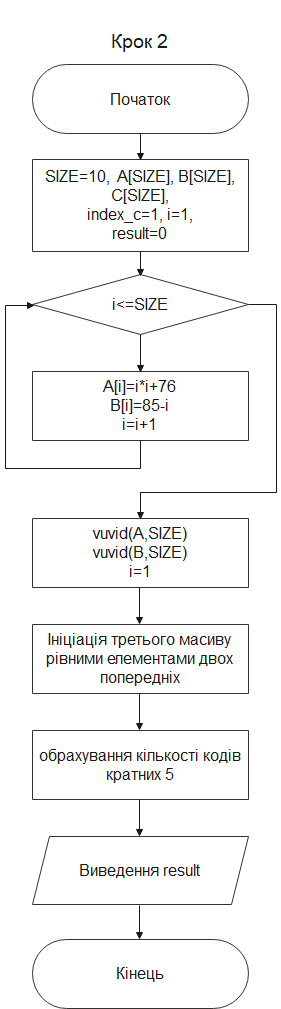
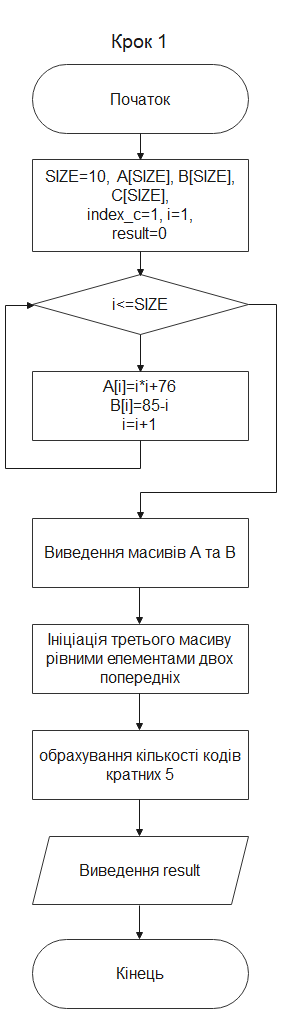
**Кінець**

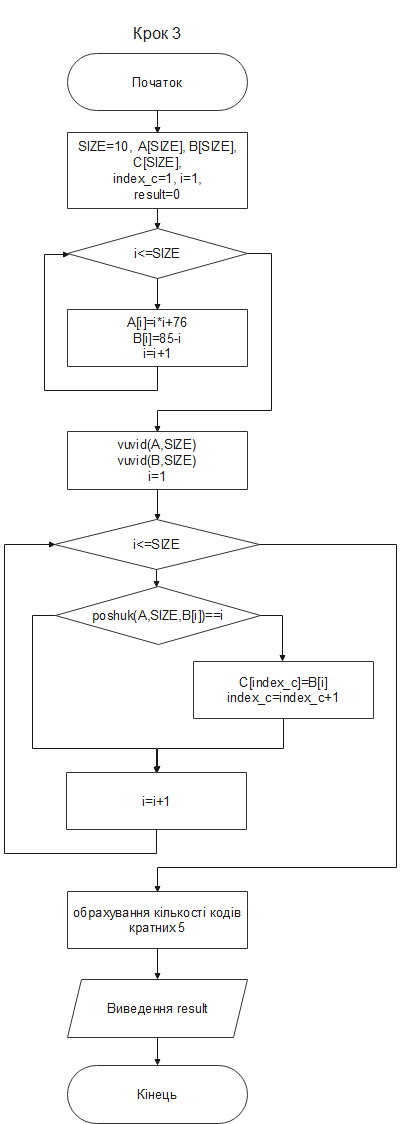
*5. Блок-схема*  *алгоритму*

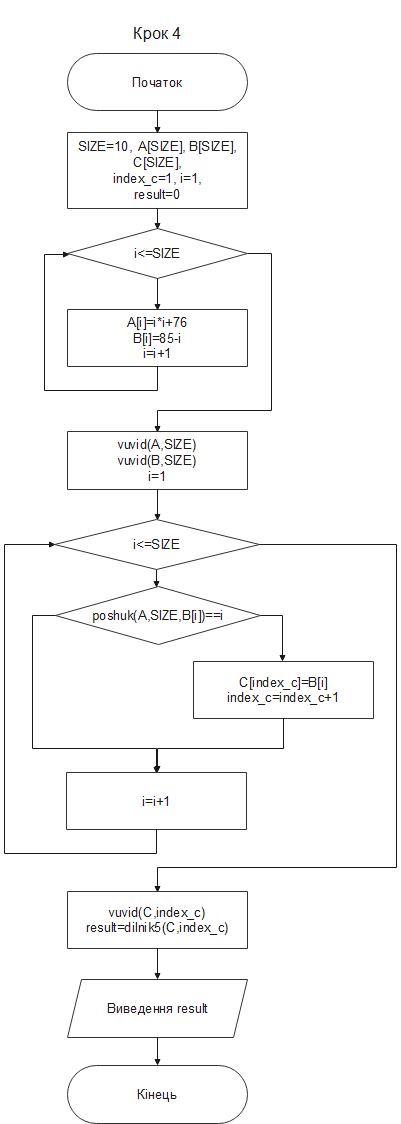












*6. Випробування алгоритму*

|  |  |
| --- | --- |
| **Блок** | **Дія** |
|  | Початок |
| **1** | A=M, P, L, R, W, I, T, Q, S, F; B=X, A, G, Y, E, O, J, M, P, L |
| **2** | C=M, P, L(код М-77, код Р-80, код L-76) |
| **3** | 77 кратне 5 – ні; 80 кратне 5 – так; 76 кратне 5 – ні; |
| **4** | Кількість = 1 |
| **5** | Кінець |

*7. Висновок*

В результаті виконання лабораторної роботи було дослідженно методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій. Та використано для лінійного пошуку елементів кратних 5 у послідовності.