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 №20

Виконав студент ІП-12 Логвиненко Владислав Олексiйович

(шифр, прізвище, ім'я, по батькові)

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

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 7**

«Дослідження лінійного пошуку в послідовностях»

**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.

**Індивідуальне завдання:**

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису трьох змінних індексованого типу з 10 символьних значень.
2. Ініціювання двох змінних виразами згідно з варіантом (**130 - і; 120 + і**).
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної згідно з варіантом(**Елементи коди яких менше 127**)**.**

*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.Потім знайти кілкість елементів 3 массиву чий код менше за 127.

*Результат* -кількість елементів третього массиву.

Побудова математичної моделі.

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Розмір масивів | цілочисельний | ***size*** | Початкове значення |
| Масив з формулою елемента 130 **- і** | індексований | ***arrA*** | Проміжкове значення |
| Масив з формулою елемента 120 **+ і** | індексований | ***arrB*** | Проміжкове значення |
| Масив рівних елементів перший двух масивів | індексований | ***arrC*** | Проміжкове значення |
| Лічильник | цілочисельний | ***i*** | Проміжкове значення |
| Лічильник | цілочисельний | ***j*** | Проміжкове значення |
| формальний параметр(перший масив) | індексований | ***arr1*** | Проміжкове значення |
| формальний параметр(другий масив) | індексований | ***arr2*** | Проміжкове значення |
| формальний параметр(третій масив) | індексований | ***Arr3*** | Проміжкове значення |
| результат роботи підпрогрмамми | цілочисельний | ***count*** | Результат(підпрограмма) |
| результат роботи прогрмамми | цілочисельний | ***counter*** | Результат |

## 2) Розв’язання

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

**Крок 1.** Визначимо основні дії.

**Крок 2.**Деталізація ініціалізації промжкових змінних ***size***, ***arrA,*** ***arrB,arrC***

**Крок 3.**Деталізація заповненя масивів ***arrA, arrB*** елементами

**Крок 4.**Деталізація заповнення масива ***arrC*** рівними елементами масивів ***arrA, arrB***

**Крок 5.**Деталізація знаходження значення змінної ***counter***

3) Псевдокод

Крок 1.

**початок**

введення

ініціалізація промжкових змінних **size**, **arrA,** **arrB,arrC**

заповненя масивів **arrA, arrB** елементами

заповнення масива **arrC** рівними елементами масивів **arrA, arrB**

знаходження значення змінної **counter**

виведення **counter**

**кінець**

Крок 2.

**початок**

введення

**size=10**

**arrA[size]**

**arrB[size]**

**arrC[size]**

заповненя масивів **arrA, arrB** елементами

заповнення масива **arrC** рівними елементами масивів **arrA, arrB**

знаходження значення змінної **counter**

виведення **counter**

**кінець**

Крок 3.

**початок**

введення

**size=10**

**arrA[size]**

**arrB[size]**

**arrC[size]**

**fillArrays(arrA, arrB)**

заповнення масива **arrC** рівними елементами масивів **arrA, arrB**

знаходження значення змінної **counter**

виведення **counter**

**кінець**

Крок 4.

**початок**

введення

**size=10**

**arrA[size]**

**arrB[size]**

**arrC[size]**

**fillArrays(arrA, arrB)**

*thirdArray* **(arrA, arrB, arrC)**

знаходження значення змінної **counter**

виведення **counter**

**кінець**

Крок 5.

**початок**

введення

**size=10**

**arrA[size]**

**arrB[size]**

**arrC[size]**

**fillArrays(arrA, arrB)**

*thirdArray* **(arrA, arrB, arrC)**

**counter = countthirdArray(arrC)**

виведення **counter**

**кінець**

*Псевдокод(підпрогрма - fillArrays):*

*Крок 1.*

**функція fillArrays(arr1[], arr2[])**

заповнення порожніх масивів елементами

**кінець**

*Крок 2.*

**функція fillArrays(arr1[], arr2[])**

**повторити**

**для *і* від** 0 **до *size* із кроком** 1

***arr1[i]*** = 130 - ***i***

***arr2[i]*** = 120+ ***і***

**все повторити**

**кінець**

*Псевдокод(підпрогрма - thirdArray):*

*Крок 1.*

**функція** *thirdArray* ***(arr1, arr2, arr3)***

заповнення масива ***arrC*** рівними елементами масивів ***arrA, arrB***

**кінець**

Крок 2.

**функція** *thirdArray* ***(arr1, arr2, arr3)***

**повторити**

**для *і* від** 0 **до *size* із кроком** 1

***arr3 [i] = 0***

**повторити**

**для *j* від** 0 **до *size* із кроком** 1

**якщо arr1[i] == arr2[j] && arr[i] < 127**

**то**

***arr3* [i] = arr1[i]**

**все якщо**

***j ++***

**все повторити**

**все повторити**

**кінець**

*Псевдокод(підпрогрма -countthirdArray):*

*Крок 1.*

**функція** *countthirdArray* ***(arr3[])***

обрахування значення змінної **counter**

**кінець**

*Крок 2.*

**функція** *countthirdArray* ***(arr3[])***

count = 0

**повторити**

**для *і* від** 0 **до *size* із кроком** 1

**якщо *arr3[i] !=0 && arr[127]<127***

**то**

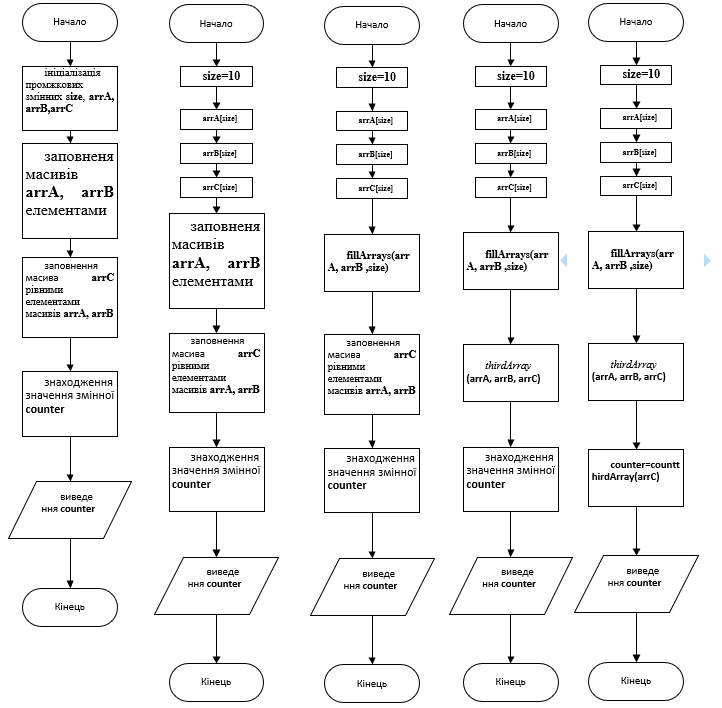
**count++**

**все якщо**

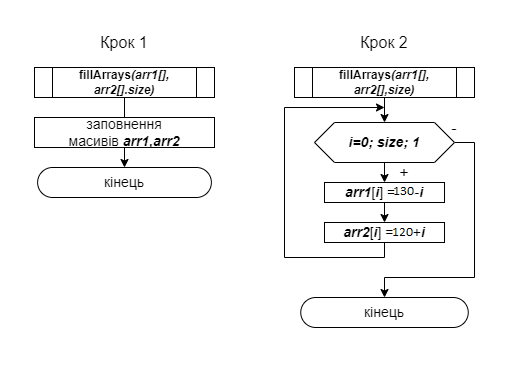
**все повторити**

count = counter

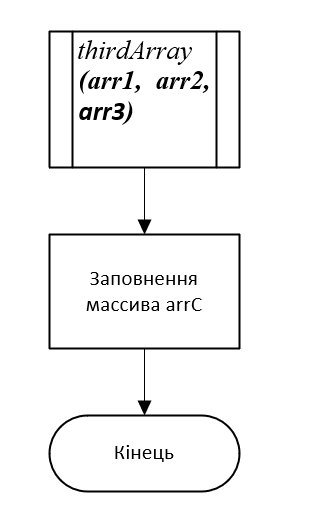
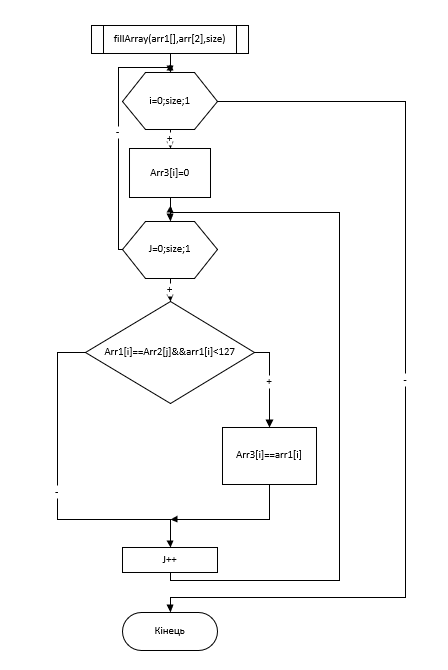
**кінець**

***4) Блок-схема***

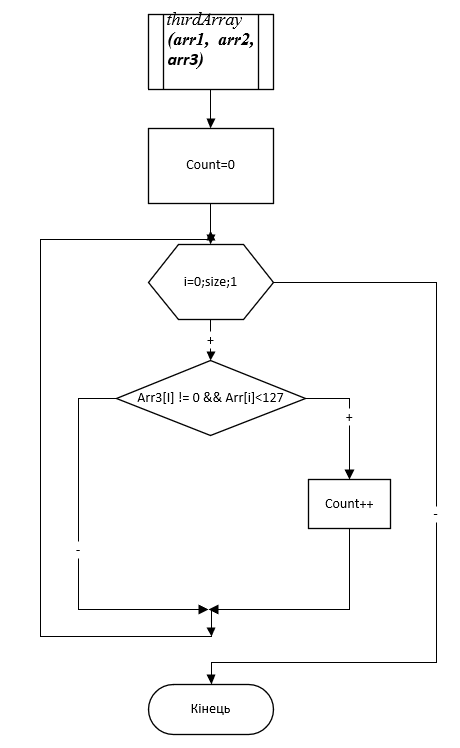
*Підпрограма(*fillArrays*)*



*Підпрограма(*thirdArray*)*

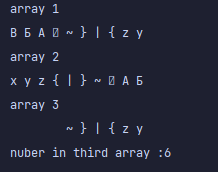
*(підпрогрма) -countthirdArray*



***Код програми на мові С++***

#include <iostream>  
#define **size** 10  
*using namespace* std;  
*void* fillArray (*unsigned char* array\_1[],*unsigned char* array\_2[]);  
*void* thirdArray(*unsigned char* array\_1[],*unsigned char* array\_2[],*unsigned char* array\_3[]);  
*void* showArray(*unsigned char* array[],string massage);  
*void* countthirdArray(*unsigned char* array\_3[]);  
*int* main() {  
 *unsigned char* array\_1[**size**],array\_2[**size**],array\_3[**size**];  
 fillArray(array\_1,array\_2);  
 thirdArray(array\_1,array\_2,array\_3);  
 showArray(array\_1,"array 1");  
 showArray(array\_2,"array 2");  
 showArray(array\_3,"array 3");  
 countthirdArray(array\_3);  
 *return* 0;  
}  
*void* fillArray (*unsigned char* array\_1[],*unsigned char* array\_2[]){  
 *for*(*int* i = 0;i < **size**;i++){  
 array\_1[i] = 130 - i;  
 array\_2[i] = 120 + i;  
 }  
}  
*void* thirdArray(*unsigned char* array\_1[],*unsigned char* array\_2[],*unsigned char* array\_3[]){  
 *for*(*int* i = 0;i < **size**;i++) {  
 array\_3[i] = 0;  
 *for* (*int* j = 0;j < **size**;j++) {  
 *if* (array\_1[i] == array\_2[j] && array\_1[i] < 127) {  
 array\_3[i] = array\_1[i];  
  
 }  
  
 }  
 }  
}  
*void* countthirdArray(*unsigned char* array\_3[]){  
 *int* count = 0;  
 *for* (*int* i = 0; i < **size**; i++) {  
 *if* (array\_3[i] != 0 && array\_3[i]< 127)  
 count++;  
 }  
 cout<<"nuber in third array :"<< count << endl;  
}  
*void* showArray(*unsigned char* array[],string massage){  
 cout << massage << endl;  
 *for* (*int* i = 0;i < **size**;i++){  
 cout << array[i] <<" ";  
 }  
 cout << endl;  
}

**Віпробування кода:**



**5) Висновок -** Було досліджено методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набуто практичних навичок їх використання під час складання програмних специфікацій.

Декомпозовано задачу на 3 етапи:

1. Генерація перших двох масивів.

2. Генерація третього масиву з рівних елементами перших двух

3.Знаходження кількості усіх елементів 3-го масиву менш за 127