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-11 Лошак Віктор Іванович**

(шифр, прізвище, ім'я, по батькові)

Перевірив: **Мартинова О.П.**

Київ 2021

**Лабораторна робота**

**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 №20**

**Задача:**

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 (табл. 1).

3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.

4. Знайти кількість елементів, коди яких менше 127.

**Розв’язок:**

1)Постановка задачі

Результатом розв’язку є число значення якого отримується

шляхом знаходження елементів масиву що відповідають заданій умові( менші за значенням ніж 127) . Для ініціалізації масивів значеннями заданими в таблиці 1 використаємо підпрограми що будуть реалізувати присвоєння значень елементам послідовностей всередині арифметичного циклу. Для правильної ініціалізації третьої змінної індексованого типу використаємо алгоритм лінійного пошуку, що шукатиме спільні елементи масиву. Даний алгоритм можна реалізувати використовуючи два цикли, один з яких є вкладеним і ітерується по елементах arr2 для пошуку вибраного в зовнішньому циклі елементу, перевіряючи відповідність за допомогою логічного оператора, і у випадку успіху записуючи значення шуканого елемента в масив 3. Попередньо ініціалізуємо arr3 нулями при створенні змінної для забезпечення правильної роботи алгоритму. Для знаходження кількості елементів третього масиву що відповідають заданій умові використаємо підпрограму що містить арифметичний цикл з вкладеним логічним оператором, в якому при ітерації по всіх елементах масиву, для кожного з них перевірятимемо відповідність вимогам задачі. Для правильної реалізації цієї перевірки скористаємося таблицею ASCII кодів символів. Значенням n є число що вводиться користувачем за запитом на початку програми.

2)Побудова математичної моделі

Складемо таблицю імен змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Внутрішня змінна функції main | індексований тип з символьних значень | arr1 | Початкове дане/ результат обчислень |
| Внутрішня змінна функції main | індексований тип з символьних значень | arr2 | Початкове дане/ результат обчислень |
| Внутрішня змінна функції main | індексований тип з символьних значень | аrr3 | Проміжний результат |
| Рахівник зовнішнього/ внутрішнього/простого арифметичного циклу | Цілочисельний тип | і, j | Проміжний результат |
| Параметр функцій | Показчик | arr, arr1, arr2, arr3 | Зберігання значення адреси фактичного параметра |
| Внутрішня змінна функції array3Init | цілочисельний | indexArr | Індекс елемента для поточного запису в arr3 |
| Внутрішня змінна функції resultCount | цілочисельний | n | Проміжний результат обчислень |

3)Програмні специфікації напишемо у псевдокоді та графічній формі у вигляді блок-схеми.

*Крок 1:* Визначимо основні дії.

*Крок 2:* Деталізуємо дії оголошення масивів всередині main

*Крок 3:* Деталізуємо дії викликів функцій всередині main

5)Псевдокод

*Крок 1*

**Початок**

**-** оголошення масивівта ініціалізація

-ініціалізаціяarr1 та arr2

-ініціалізаціяarr3

- знаходження кількості необхідних елементів третього масиву в підпрограмі та виводу результату

**Кінець**

*Крок 2*

**Початок**

**- оголошення** arr1[10], arr2[10], arr3[10];

-ініціалізаціяarr1 та arr2

-ініціалізаціяarr3

- знаходження кількості необхідних елементів третього масиву в підпрограмі та виводу результату

**Кінець**

*Крок 3*

**Початок**

**- оголошення** arr1[10], arr2[10],

arr3[10]=0;

- **виклик** array1Init( arr1), array2Init( arr2)

- **виклик** array3Init( arr3)

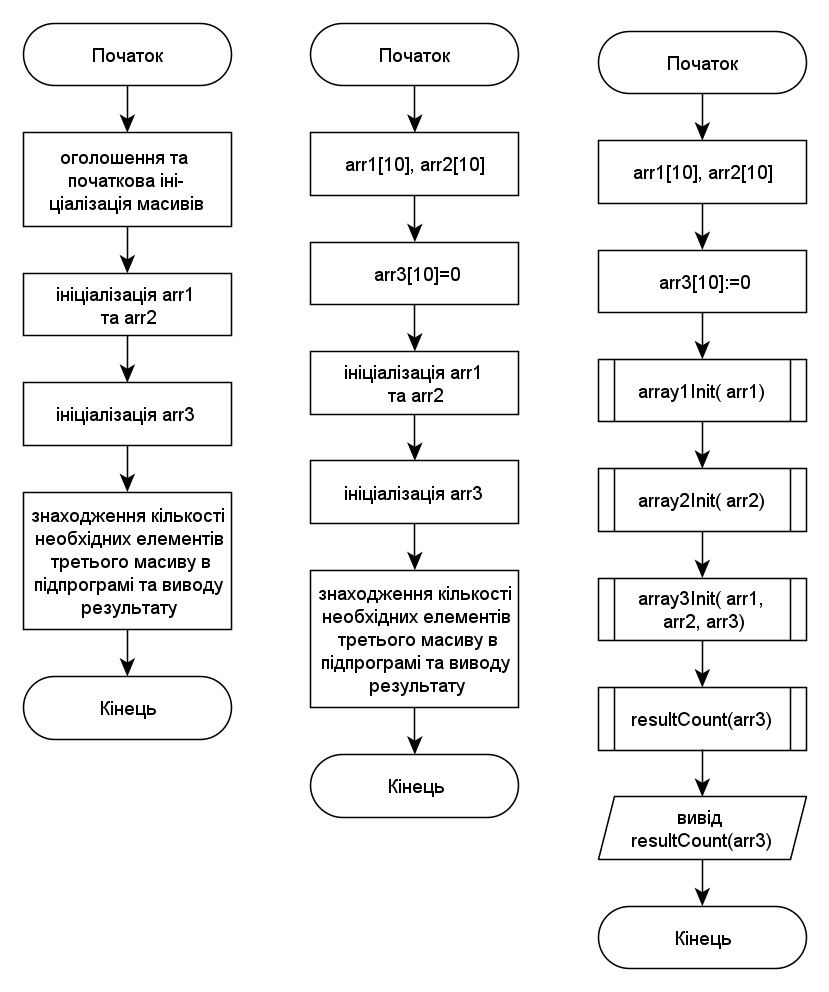
- **виклик** resultCount(arr3**)**

**Вивід результату** resultCount(arr3**)**

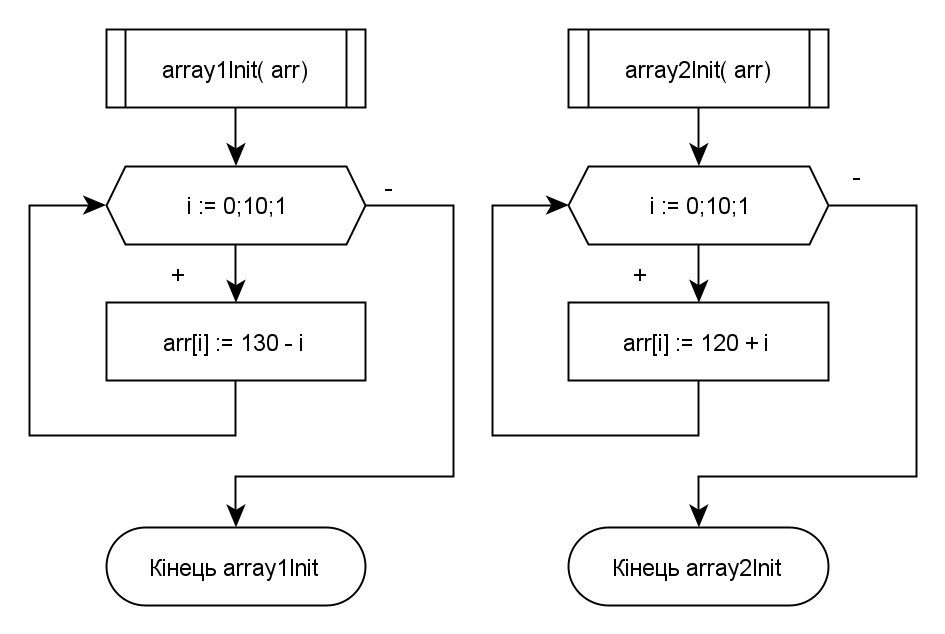
**Кінець**

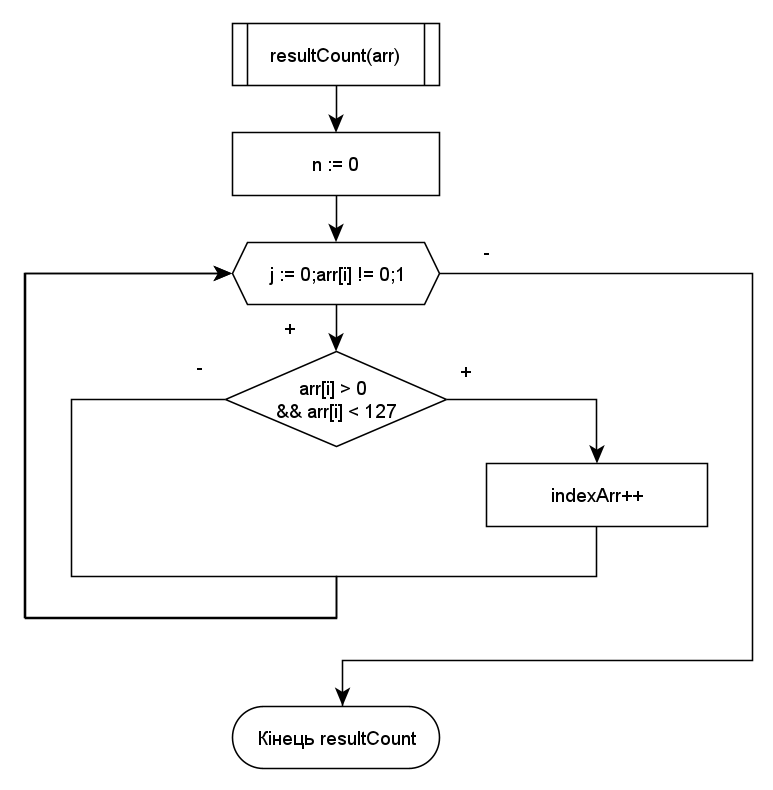
**Блок-Схема**

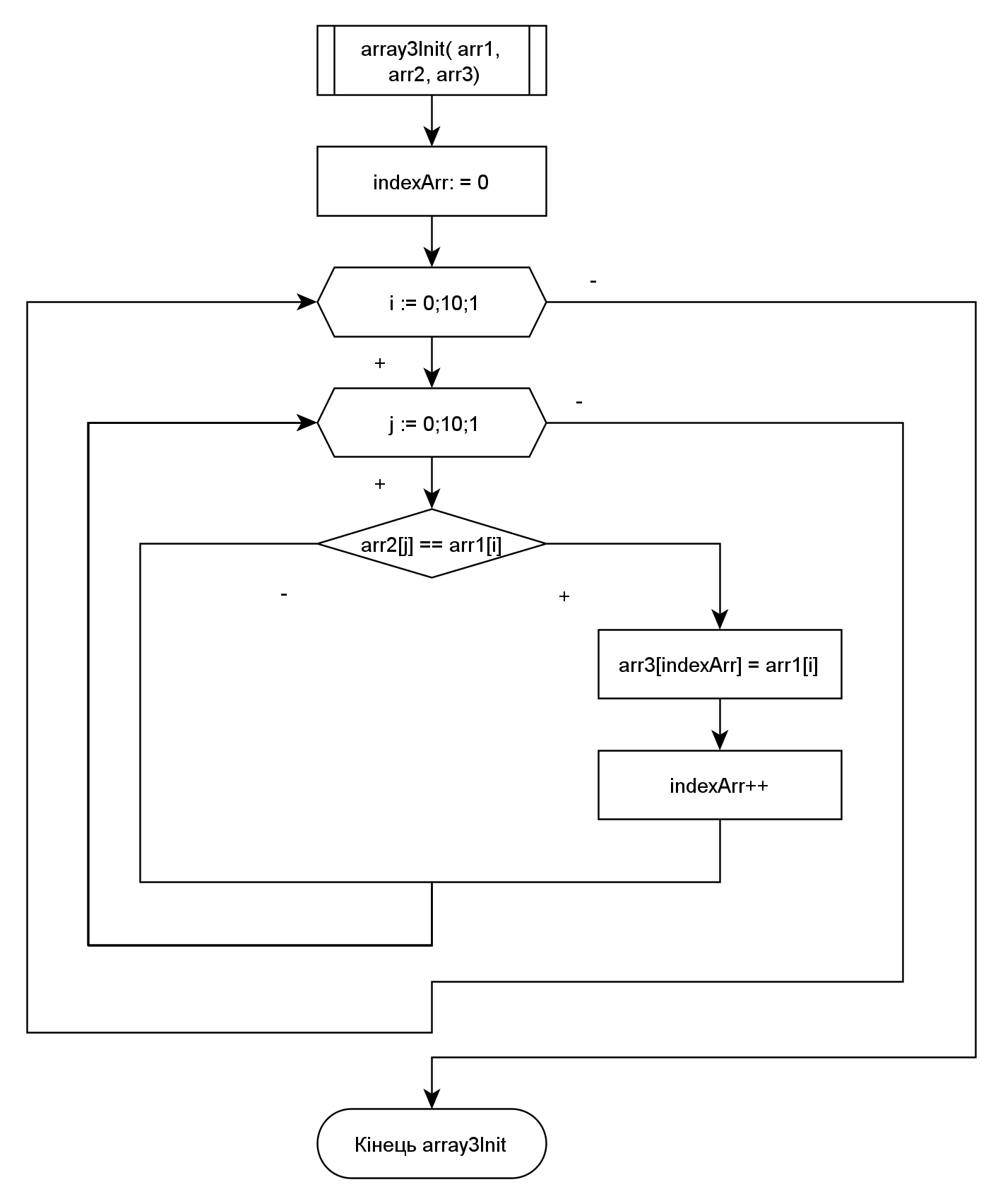
**Крок 1 Крок 2 Крок 3**

****

**Підпрограми:**

****





**Код програми c++:**

#include <iostream>

void array1Init(char\* arr);

void array2Init(char\* arr);

void array3Init(char\* arr1, char\* arr2, char\* arr3);

int resultCount(char\* arr);

int main( )

{

char arr1[10], arr2[10], arr3[10] = { 0 };

array1Init(arr1);

array2Init(arr2);

array3Init(arr1, arr2, arr3);

std::cout <<"The amount of characters in array 3, which have numeric values less then 127 are: " << resultCount(arr3) << std::endl;

}

void array1Init(char\* arr)

{

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

arr[i] = 130 - i;

}

void array2Init(char\* arr)

{

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

arr[i] = 120 + i;

}

void array3Init(char\* arr1, char\* arr2, char\* arr3)

{

int indexArr = 0;

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

{

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

{

if (arr2[j] == arr1[i])

{

arr3[indexArr] = arr1[i];

indexArr++;

}

}

}

}

int resultCount(char\* arr)

{

int n = 0;

for (int i = 0; arr[i] != 0; ++i)

{

if (arr[i] > 0 && arr[i] < 127) n++;

}

return n;

}

Перевіримо правильність алгоритму cтворивши функцію print для відображення проміжних результатів роботи програми і відслідковування правильності виконання алгоритму:

Код функції на с++:

void printArr(char\* arr)

{

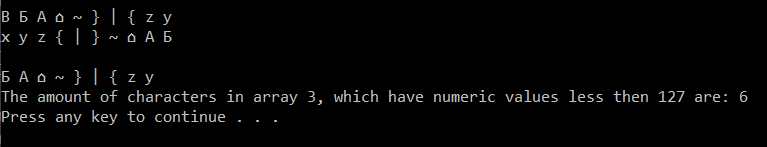
for (int i = 0; i < 10; ++i)

std::cout << arr[i] << " ";

std::cout << '\n';

}

Результат запуску програми для перевірки:



|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | Ініціалізація першого масивy char значеннями від 130 до 120;(див перший рядок) |
| 2 | Ініціалізація другого масивy char значеннями від 120 до 130;(див другий рядок) |
| 3 | Ініціалізація третього масивy char значеннями спільними для першого і другого масивів ;(див третій рядок) |
| 4 | Вивід результату обчислення де:  Б=-128 А=-127 ⌂=-126 в коді ASCII відповідно |
| 7 | Виведення 6 |
|  | Кінець |

**Висновок:**

В ході виконання даної лабораторної роботи я навчився використовувати алгоритм лінійного пошуку , досліджував нелінійні програмні специфікації для ініціалізації і аналізу структур даних, набув практичних навичок даних використання таблиці кодів ASCII для пошуку заданих умовою значень. Навчився створювати блок-схеми алгоритмів що містять алгоритм лінійного пошуку та описувати програму за допомогою псевдокоду. Використовуючи отримані знання я успішно обрахував і відобразив на екрані кількість спільних елементів заданих масивів що мають код менший за 127, на компільованій мові с++.