# 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 і структури даних

»

Варіант 23

Виконав студент Панченко Сергій Віталійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 202

## Лабораторна робота 7

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

**Постановка:**



**Математична модель**:

**Складемо таблицю імен змінних**

|  |  |  |  |
| --- | --- | --- | --- |
| Зміна | Тип | Ім’я | Призначення |
| Arr1 | Char[] | Перший масив | Проміжне значення |
| Arr2 | Char[] | Другий масив | Проміжне значення |
| Arr3 | Char[] | Третій масив | Проміжне значення |
| I | Int | Змінна циклу | Проміжне значення |
| k | Int | Змінна циклу | Проміжне значення |
| Average | Double | Середнє значення | Результат |
| Sum | Int | Сума | Проміжне значення |
| count | int | Кількість елементів, коди яких > 38 | Проміжне значення |

Заповнивши масиви відповідними символьними значеннями, знайдемо їх спільні елементи та перенесемо їх у третій масив. Коди елементів 3-го масиву, які > 38, просумуємо та розділимо на їх кількість, отримавши середнє значення.

**Псевдокод:**

Крок 1: Визначимо основні дії

Крок 2: деталізуємо заповнення arr1 та arr2

Крок 3: деталізуємо знаходження спільних елементів arr1, arr2 та заповнимо 3-ій масив

Крок 4: деталізуємо знаходження сер. значення

Крок 1:

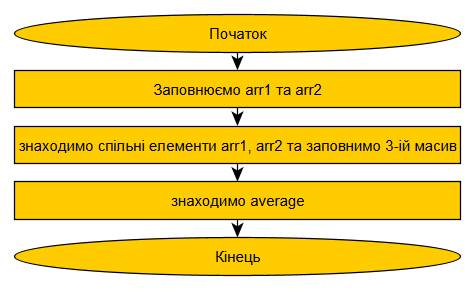
Початок

Заповнюємо arr1 та arr2

знаходимо спільні елементи arr1, arr2 та заповнимо 3-ій масив

знаходимо average

Кінець



Крок 2:

Початок

Поки i=0,10:

arr1[i] = char(35 + 3 \* i);

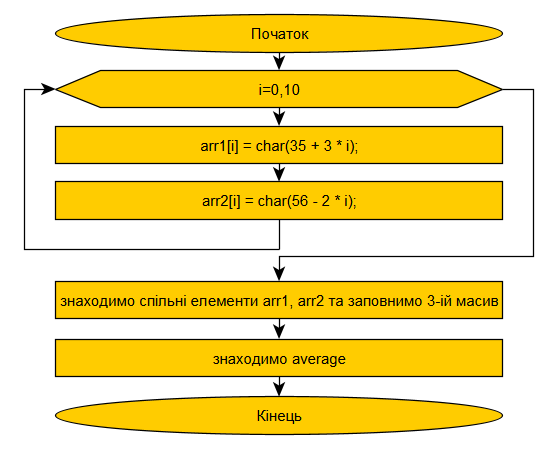
arr2[i] = char(56 - 2 \* i);

Поки все

знаходимо спільні елементи arr1, arr2 та заповнимо 3-ій масив

знаходимо average

Кінець



Крок 3:

Початок

Поки i=0,10:

arr1[i] = char(35 + 3 \* i);

arr2[i] = char(56 - 2 \* i);

Поки все

Поки i=0,10:

Поки k=0,10:

Якщо 35 + 3 \* i == 56 - 2 \* k:

arr3[i] = char(35 + 3 \* i);

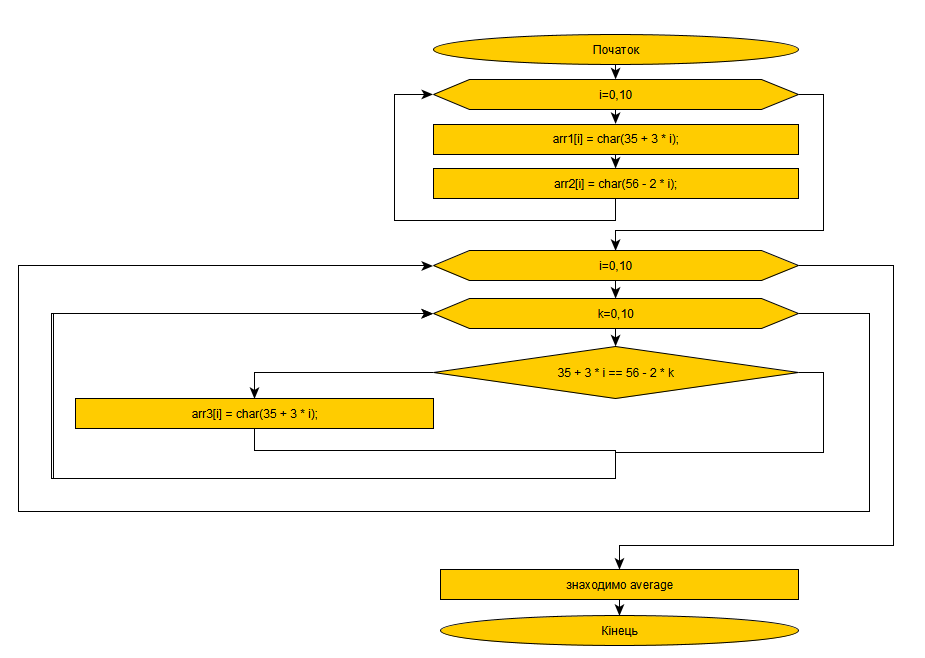
Все якщо

Поки все

Поки все

знаходимо average

Кінець



Крок 4:

Початок

Поки i=0,10:

arr1[i] = char(35 + 3 \* i);

arr2[i] = char(56 - 2 \* i);

Поки все

Поки i=0,10:

Поки k=0,10:

Якщо 35 + 3 \* i == 56 - 2 \* k:

arr3[i] = char(35 + 3 \* i);

Все якщо

Поки все

Поки все

Поки i=0,10:

Якщо int(arr3[i]>38):

Sum+=int(arr3[i]);

Count++;

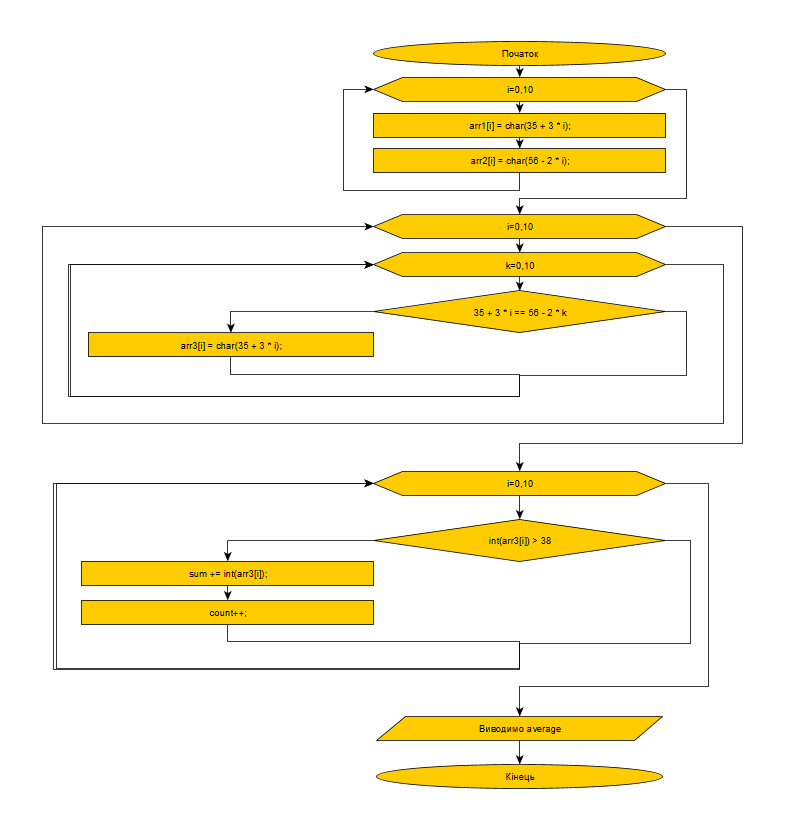
Все якщо

Поки все

Average = sum / count;

Виводимо average;

Кінець



Код C++:

#include <iostream>

using namespace std;

int main()

{

char arr1[10];

char arr2[10];

char arr3[10];

int sum = 0, count = 0;

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

arr1[i] = char(35 + 3 \* i);

arr2[i] = char(56 - 2 \* i);

}

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

for (int k = 0;k < 10;k++) {

if (35 + 3 \* i == 56 - 2 \* k) {

arr3[i] = char(35 + 3 \* i);

}

}

}

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

if (int(arr3[i]) > 38) {

sum += int(arr3[i]);

count++;

}

}

double average = sum / count;

cout << "Average sum of codes > 38: " << average << endl;

system("pause");

}

Перевірка:

Створимо таблицю для перевірки:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| index | 35 + 3 \* i | Arr1 | 56 - 2 \* i | Arr2 | **common** | **Arr3** |
| 0 | 35 | # | 56 | 8 |  |  |
| 1 | 38 | & | 54 | 6 | 38 | & |
| 2 | 41 | ) | 52 | 4 |  |  |
| 3 | 44 | , | 50 | 2 | 44 | , |
| 4 | 47 | / | 48 | 0 |  |  |
| 5 | 50 | 2 | 46 | . | 50 | 2 |
| 6 | 53 | 5 | 44 | , |  |  |
| 7 | 56 | 8 | 42 | \* | 56 | 8 |
| 8 | 59 | ; | 40 | ( |  |  |
| 9 | 62 | > | 38 | & |  |  |

Отже, вручну розрахувавши, бачимо, що алгоритм працює. Сума та кількість, середнє значення підраховані правильно. Count =3 , Sum =150, average = 50

**Висновок:**

Під час лабораторної роботи дослідили методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях. Математична модель, блок-схеми, псевдокод, код випробовування наведені. Оскільки перевірені вручну результати розрахунку правильні, то алгоритм правильно подає результат.