Міністерство освіти і науки України

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

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації

і управління

Звіт

з лабораторної роботи № 6 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійного пошуку в послідовностях»

Варіант 6

Виконав студент ІП-01 Гагарін Артем Олексійович

 (шифр, прізвище, ім'я, по батькові)

Перевірив Вітковська Ірина Іванівна

  ( прізвище, ім'я, по батькові)

Київ 2020

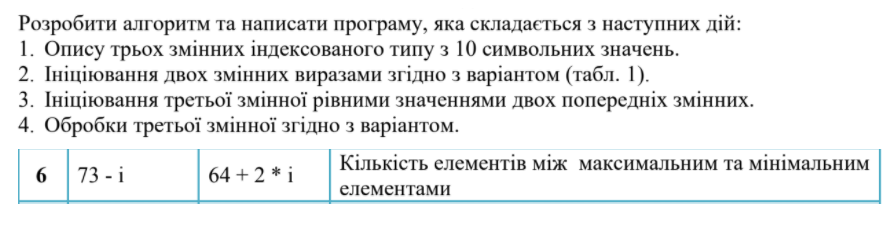
**Лабораторна робота 7**

**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.

**Задача 6.** 

**1. Постановка задачі.** Результатом розв’язку є обчислена кількість елементів між максимальним і мінімальним елементами третього масиву, утвореного рівними елементами першого і другого масивів, елементи яких задані наступними виразами: 73 – і та 64 + 2 \* і.

**2**. **Побудова математичної моделі**. Складемо таблицю імен змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Масив А | Символьний | А | Початкове дане |
| Масив B | Символьний | B | Початкове дане |
| Масив С | Символьний | C | Масив, що містить спільні елементи масивів A і B |
| len | Цілий | len | Кількість елементів масиву С |
| num | Цілий | num | Результат |

За умовою задачі необхідно задати два масиви за відповідними виразами. Значення обчислених виразів відповідає десятковому коду певного символу. Третій масив формується з спільних елементів першого і другого.

Далі необхідно знайти кількість елементів між мінімальним та максимальним елементами третього масиву.

Загалом можна виділити три підпрограми, що виконуватимуть наступні функції: введення масивів A і B, виведення цих масивів, створення масиву C зі значень A i B та обрахунок кількості елементів.

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

**Псевдокод**

product\_of\_elements

main

**Початок**

**введення** а0, an, d

multiply = 1

**якщо** a0 >= an

**то**

multiply = a0

multiply \*= product\_of\_elements(a0 - d, an, d)

**все** **якщо**

**виведення** multiply

**Кінець**

**Початок**

**введення** A[10], B[10], C[10]

get\_lists(A, B);

print\_list(A, 10), print\_list(B, 10);

int len = form\_c(A, B, C);

**виведення** product\_of\_elements(a0, an, d)

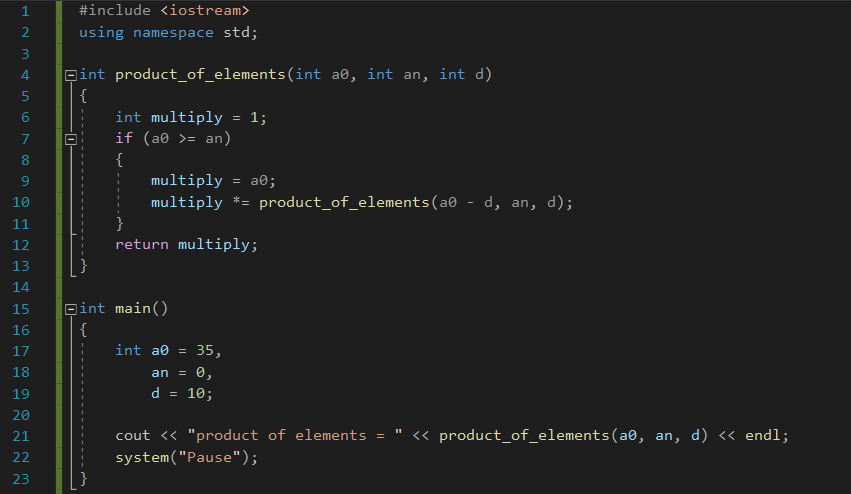
**Кінець**

**Блок-схема**





**Код**

****

4. Перевіримо роботу програми.

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | a0 = 35, an = 5, d = 10 |
| 2 | multiply(35): multiply = 35 |
| 3 | multiply(35-10=25): multiply = 875 |
| 4 | multiply(25-10=15): multiply = 13125 |
| 5 | multiply(15-10=5): multiply = 65625 |
| 6 | Вивід: 65625 |
| 7 | Кінець. |

**Висновок:** побудовано алгоритм розв’язання поставленої задачі, що включає рекурентну функції. Рекурентною називається функція, що містить виклик самої себе. Перевірено правильність виконання алгоритму. Складність: О(1).