# 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 № 4 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження алгоритмів розгалуження»

Варіант 7

Виконав студент ІП-11 Головня Олександр Ростиславович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2021

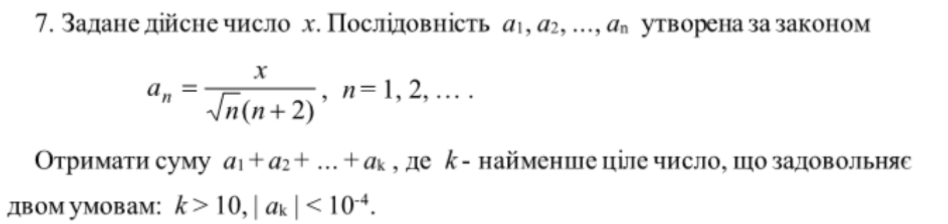
**Лабораторна робота №3**

**Дослідження ітераційних циклічних алгоритмів**

**Мета –** дослідити подання операторів повторення дій та набути практичних навичок їх використання під час складання циклічних програмних специфікацій.

**Індивідуальне завдання:**

Варіант 7.



**Постановка задачі**

Створюємо змінні які потрібні. Створюємо цикл, який буде рахувати **An** та додавати це число в змінну **Sum**, також присвоювати змінній **k** змінну **n** та збільшувати її на 1, таким чином цикл працює поки не виконається певні умови, як тільки умова виконується – виходимо з циклу

(Піднесення до степеня – функція **pow()** )

**Побудова математичної моделі**

Складемо таблицю змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| 1 | Дійсний | X | Початкове дане |
| 2 | Дійсний | n | Початкове дане |
| 3 | Дійсний | k | Проміжне дане |
| 4 | Дійсний | a | Проміжне дане |
| 4 | Дійсний | Sum | Вихідні дані |

**Розв’язання**

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

Крок 1. Визначимо основні дії:

Крок 2. Деталізація дії знаходження числа послідовності та суми за допомогою циклу

Крок 3. Деталізація дії виведення суми

**Псевдокод**

Крок 1

**Початок**

Знаходження числа послідовності та суми за допомогою циклу

Виведення Суми

**Кінець**

Крок 2

**Початок**

Поки !(k > 10 Та abs(a) < 0.00001)

то a = 0;

a = x / (pow(n, 0.5) \* (n + 2));

k = n;

n++;

sum = sum + a;

все повторити

Виведення Суми

**Кінець**

Крок 3

**Початок**

Поки !(k > 10 Та abs(a) < 0.00001))

то a = 0;

a = x / (pow(n, 0.5) \* (n + 2));

k = n;

n++;

sum = sum + a;

все повторити

cout << sum << endl;

**Кінець**

**Блок-схема**

**Крок 1**



**Крок 2**



**Крок 3**



Перевірка

**Нехай n = 10**

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | Цикл a = 0;  a = 10 / (pow(n, 0.5) \* (n + 2));  k = n;  n++;  sum = sum + a;  Повторити |
| 3 | Вивід sum = 15.1968 |

**Нехай n = 0**

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | Цикл a = 0;  a = 0 / (pow(n, 0.5) \* (n + 2));  k = n;  n++;  sum = sum + a;  Повторити |
| 3 | Вивід sum = 0 |

**Висновок**

Отже, ми дослідили подання операторів повторення дій та набули практичних навичок їх використання під час складання циклічних програмних специфікацій.