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 № 6 з дисципліни

«Алгоритми та структури даних.

Основи алгоритмізації»

«Дослідження рекурсивних алгоритмів»

Варіант 16­­

Виконав студент Зубарев Микола Костянтинович

Перевірив Вітковська Ірина Іванівна

Київ 2022

**Лабораторна робота 6**

**Дослідження рекурсивних алгоритмів**

**Мета:** *дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.*

**Варіант 16**

**Задача.**

****

Розв'язання.

**Постановка задачі.** Результатом розв’язку буде виведення простих дільників числа. Обчислювальний процес в ході виконання алгоритму буде здійснюватися за допомогою заданих математичних формул та нерівностей.

**Математична модель.** Складемо таблицю змінних

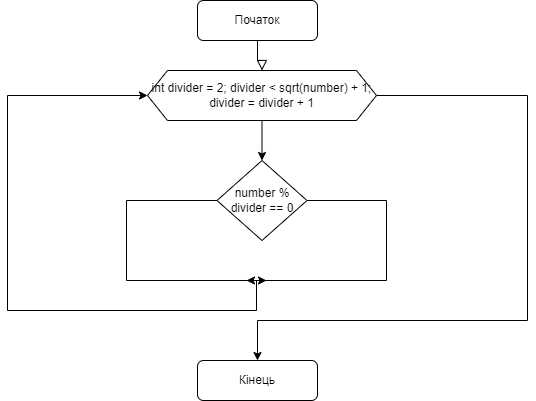
|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім'я | Призначення |
| Натуральне число | Цілий | ***n*** | Ввід |
| Початковий | Цілий | ***is\_prime*** | Функція |
| Дільник | Цілий | ***divider*** | Початкове дане |

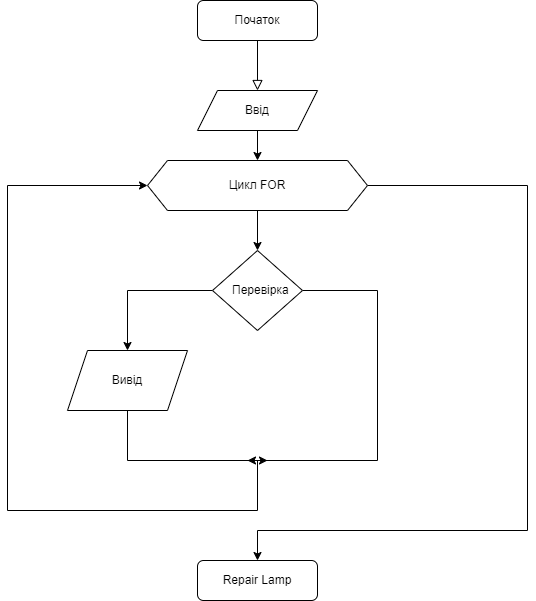
**Псевдокод**

|  |  |  |
| --- | --- | --- |
| *Крок 1* | *Крок 2 Функція is\_prime* | *Крок 3* |
| **Початок** | **Початок** | **Початок** |
| **Введення**  Розрахунок | **Для divider від 2 до sqrt(number)+1**  **divider = divider + 1**  **Якщо number % divider = 0**  **Все повторити**  **Все повторити** | **Введення: n**  **Для divider від 2 до (n/2)+1**  **divider = divider + 1**  **Якщо n % divider та is\_prime(divider)**  **То Вивід divider**  **Все повторити**  **Все повторити** |
| **Кінець** | **Кінець** | **Кінець** |

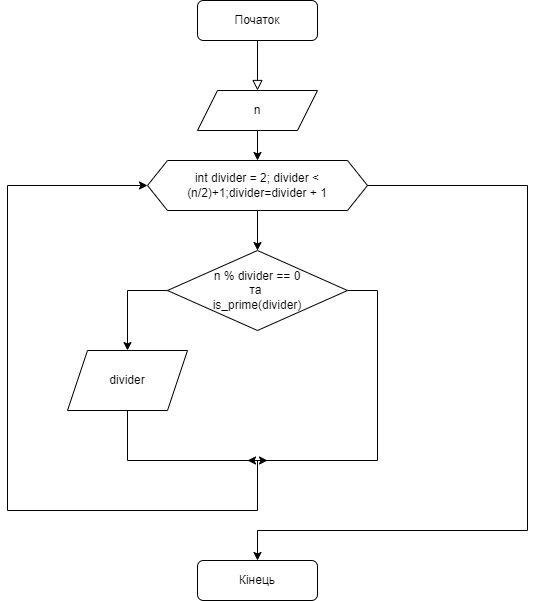
**Блок схема**

*Крок 1 Крок 2*





Крок 3



**Код програми**

#include <stdio.h>

#include <math.h>

int is\_prime(int number) {

for (int divider = 2; divider < (int)(sqrt(number)) + 1; divider++) {

if (number % divider == 0) {

return 0;

}

}

return 1;

}

int main() {

int n;

printf("n = ");

scanf\_s("%u", &n);

for (int divider = 2; divider < (int)(n / 2) + 1; divider++) {

if (n % divider == 0 && is\_prime(divider)) {

printf("%u ", divider);

}

}

}

**Випробування алгоритму.** Перевіримо роботу алгоритму в залежності від n

|  |  |  |  |
| --- | --- | --- | --- |
| Блок | Випробування 1 | Випробування 2 | Випробування 3 |
|  | Початок | Початок | Початок |
| 1. Введення | 4 | 21 | 24 |
| 2. Розв'язок | Пошук простих дільників числа | | |
| 3. Виведення | Прості дільники: 2 | Прості дільники: 3, 7 | Прості дільники: 2, 3 |
|  | Кінець | Кінець | Кінець |

**Висновок.** Коли я виконував цю лабораторну роботу я не навчився використовувати рекурсивний метод, проте я гарно повторив цикли. Під час виконання даної лабораторної роботи можна було набути практичні навички використання рекурсивного методу під час складання програмної специфікації.