**Vampire Number Finder**

This C program efficiently finds vampire numbers within a given range using multi-threading. The program partitions the numbers and assigns each thread a range to check for vampire numbers. The details of vampire numbers are logged into an output file, and the total count is displayed.

**Table of Contents**

* [Prerequisites](https://chat.openai.com/c/8d965aae-2e7d-44a3-94ab-f3c522b57e53#prerequisites)
* [Instructions for Execution](https://chat.openai.com/c/8d965aae-2e7d-44a3-94ab-f3c522b57e53#instructions-for-execution)
* [File Descriptions](https://chat.openai.com/c/8d965aae-2e7d-44a3-94ab-f3c522b57e53#file-descriptions)
* [Customization](https://chat.openai.com/c/8d965aae-2e7d-44a3-94ab-f3c522b57e53#customization)
* [Performance Analysis](https://chat.openai.com/c/8d965aae-2e7d-44a3-94ab-f3c522b57e53#performance-analysis)

**Prerequisites**

* C compiler
* Windows operating system
* (For Linux users, the program can be modified to use pthread library)

**Instructions for Execution**

1. **Clone the Repository:**

*git clone https://github.com/your-username/vampire-number-finder.git cd vampire-number-finder*

1. **Compile the Program:**

*gcc -o vampire\_finder vampire\_finder.c -lpthread*

1. **Run the Program:**

*./vampire\_finder*

1. **Review Output:**

The program will generate an output file named **OutFile.txt** containing details of vampire numbers found by each thread. The total count of vampire numbers will also be displayed.

**File Descriptions**

* **vampire\_finder.c**: The main C program that finds vampire numbers using multi-threading.
* **input.txt**: Input file containing values for N (range) and M (number of threads).
* **OutFile.txt**: Output file logging vampire numbers found by each thread.

**Customization**

You can adjust the input parameters by modifying the **input.txt** file. The format is:

N M

* N: The range of numbers to search for vampire numbers.
* M: The number of threads to use.

**Performance Analysis**

The program's performance can be analyzed using the generated log file and by measuring the elapsed time for different input parameters. Use the provided performance analysis section in the readme to interpret the graphs and observations.