Міністерство освіти і науки України

Сумський державний університет

Кафедра комп’ютерних наук

Секція інформаційно-комунікаційних технологій

**Пояснювальна записка**

**до курсової роботи**

з дисципліни

«Програмування»

Викладач [Прокопенко В. М.](https://mix.sumdu.edu.ua/info/userinfo/profcompanysumy)

Студент Єлек Я. Ю.

Група: КН-41

Варіант 13

Суми ˗ 2025

ЗМІСТ

1. Постановка задачі...............................................................3
2. Теоретичні відомості..........................................................5
3. Опис структури даних та вимоги до них..........................7
4. Алгоритм роботи програми...............................................9
5. Опис функцій користувача................................................12
6. Опис файлів та їх призначення.......................................14
7. Список використаних бібліотек......................................16
8. Інструкція для роботи з програмою...............................17
9. Приклад тестування та результати роботи програми…19
10. Графіки..............................................................................22
11. Висновки...........................................................................24
12. Список використаної літератури....................................26
13. Додаток: програмний код................................................27

# ПОСТАНОВКА ЗАДАЧІ

**Варіант: 13**

Метою курсової роботи є розробка програми мовою програмування С, яка обчислює силу, що діє на заряджені частинки у магнітному полі згідно із законом Лоренца. Програма має забезпечити можливість обчислення сили FFF для трьох незалежних варіантів набору вхідних параметрів у заданому часовому проміжку з фіксованим кроком інтегрування.

Завдання передбачає:

* описати масив структур, що складається з трьох елементів, де кожна структура містить набір параметрів: T — тривалість моделювання, dt — крок часу, k0 — початковий коефіцієнт, γ і r (в програмі позначено як y) — зміннs коефіцієнти, e0 — початкове значення заряду, gamma — кут між напрямком швидкості та індукцією магнітного поля В (в умові значення не було надано тому взято за π/2, 90 градусів) , v — швидкість частинки;
* для кожного варіанта обчислити магнітну індукцію B за поданими в умові формулами та силу Лоренца F () у проміжку часу від 0 до T з кроком dt;
* результати обчислень відобразити на екрані та зберегти у текстовий файл output.txt;

Вхідні дані повинні зчитуватися з файлу input.txt. Оскільки умова завдання містила не всі числові значення параметрів для розрахунку, вони були підібрані самостійно в межах фізично коректних та навчально обґрунтованих значень. Значення параметрів були задані у файлі вручну у вигляді трьох рядків, що відповідають кожному варіанту. Це дозволило зберегти універсальність програми та можливість змінювати вхідні дані без переписування коду.

Для реалізації поставленої задачі було використано структурне програмування — одна з основоположних парадигм мови С. Програма побудована як набір окремих функцій, кожна з яких виконує обчислення конкретного параметра (коефіцієнта k, заряду e, індукції B, сили F), що дозволяє зробити код гнучким, зрозумілим і зручним для масштабування.

Окрема увага приділена обробці помилок: перевіряється можливість відкриття файлів, а також правильність структури вхідних даних. У разі виникнення помилки користувач отримує відповідне повідомлення, і програма завершує роботу. Таким чином, реалізований застосунок є стійким до базових помилок введення/виведення та готовий до практичного застосування у навчальних цілях або для демонстрації дії сили Лоренца в рамках симуляційних фізичних експериментів.

# ТЕОРЕТИЧНІ ВІДОМОСТІ

Під час розроблення програми для розрахунку сили, що діє на заряджені частинки в магнітному полі за законом Лоренца, було використано мову програмування С. Ця мова є однією з найефективніших і найпоширеніших мов низького рівня, яка широко застосовується в системному програмуванні, наукових розрахунках та технічних задачах. Вона забезпечує високий рівень контролю над апаратними ресурсами комп’ютера, дозволяє працювати з пам’яттю, файлами, масивами та математичними операціями.

У створеній програмі реалізовано низку ключових понять і конструкцій мови C:

1. **Структура (struct)**

**Структура** — це спеціальний тип даних, який дозволяє об'єднати декілька змінних різного типу в одну логічну одиницю. У нашій програмі створено структуру InputData, яка містить набір параметрів, необхідних для обчислень:

* тривалість моделювання T,
* крок інтегрування dt,
* коефіцієнт k0,
* параметр γ (y),
* початковий заряд e0,
* параметр експоненційного згасання r,
* швидкість частинки v.

Це дозволяє зручно зберігати і передавати весь набір даних одного варіанта розрахунку як одну змінну.

1. **Масив структур**

Для підтримки кількох варіантів розрахунку використовується масив структур. У програмі задано NUM\_VARIANTS = 3, тобто обробляється три незалежні набори вхідних параметрів. Це дозволяє провести моделювання для різних умов в межах одного запуску програми.

1. **Цикл for**

Основна частина обчислень організована за допомогою циклів for, які забезпечують прохід по часовій шкалі — від t = 0 до t = T з фіксованим кроком dt. У кожній ітерації обчислюються поточні значення змінних (індукції, заряду, сили тощо) для відповідного моменту часу.

1. **Функції**

Програма розділена на **логічно ізольовані функції**, кожна з яких відповідає за окремий розрахунок:

calculate\_k() — обчислює змінний коефіцієнт k залежно від часу;

calculate\_e() — обчислює заряд частинки з урахуванням експоненційного згасання;

calculate\_B() — визначає значення магнітної індукції у різні періоди часу;

calculate\_LorentzForce() — реалізує формулу сили Лоренца: **F=q⋅v⋅B⋅sin(α)**де α — кут між напрямком руху частинки та вектором магнітного поля (в нашому випадку α = π/2, бо в умові нам не було дано значення α тому я сам вибрав дане значення).

1. **Робота з файлами**

Програма реалізує повний цикл роботи з файлами:

Зчитування вхідних параметрів з файлу input.txt;

Форматований вивід результатів у файл output.txt;

Паралельно результати також виводяться у консоль для зручності користувача.

1. **Використані бібліотеки**

У програмі використовуються лише необхідні стандартні бібліотеки мови С:

**<stdio.h>**  
Забезпечує функції для роботи з файлами та виводу: fopen, fclose, fscanf, fprintf, printf, fgets.

**<stdlib.h>**  
Використовується для базових операцій, таких як повернення коду завершення return, а також для коректного оброблення помилок відкриття файлів.

**<math.h>**  
Додає доступ до математичних функцій, необхідних для фізичних розрахунків:

exp() — експонента (застосовується в обчисленні заряду);

sin() — синус (для сили Лоренца).

# ОПИС СТРУКТУРИ ДАНИХ ТА ВИМОГИ ДО НИХ

Для розв’язання задачі обчислення сили, що діє на заряджену частинку в магнітному полі за законом Лоренца, в програмі використовується структура InputData. Ця структура містить вхідні параметри, необхідні для обчислень. Дані зберігаються у вигляді масиву структур, кожен елемент якого відповідає окремому варіанту розрахунку.

У таблиці нижче наведено відповідність між фізичними параметрами задачі, змінними в програмі, їхніми типами та призначенням:

**Таблиця змінних:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ім’я параметра у формулі** | **Змінна у програмі** | **Тип змінної** | **Призначення** |
| T | T | double/float | Кінцевий момент часу, до якого ведуться розрахунки |
| Δt | dt | double/float | Крок інтегрування часу |
| k₀ | k0 | double/float | Коефіцієнт, що входить у формулу розрахунку індукції |
| γ | y | double | Кут між напрямком швидкості та вектором магнітної індукції |
| ε₀ | e0 | double | Заряд частинки |
| r | r | double | Відстань від центра магнітного поля |
| v | v | double | Швидкість зарядженої частинки |
| B | B | double | Магнітна індукція (вихідний параметр) |
| F | F | double | Сила, що діє на заряд (вихідний параметр) |

# АЛГОРИТМ РОБОТИ ПРОГРАМИ

Зображення, що містить текст, знімок екрана, графічний дизайн, чорно-білий

Вміст, створений ШІ, може бути неправильним.  
Рисунок 1 – Блок-схема main() функції

Зображення, що містить знімок екрана, чорний, чорно-білий, Графіка

Вміст, створений ШІ, може бути неправильним.

Рисунок 2 – Блок-схема calculate\_k() функції

Зображення, що містить текст, знімок екрана, Шрифт, схема

Вміст, створений ШІ, може бути неправильним.

Рисунок 3 – Блок-схема calculate\_e() функції

Зображення, що містить текст, знімок екрана, Шрифт, схема

Вміст, створений ШІ, може бути неправильним.

Рисунок 4 – Блок-схема calculate\_B() функції

# **Опис функцій користувача**

У процесі виконання курсової роботи було розроблено декілька користувацьких функцій, які забезпечують обчислення необхідних фізичних величин згідно з поставленою задачею. Нижче наведено опис кожної функції, її призначення, вхідні та вихідні параметр

**1. double calculate\_k (double t, double T, double k0, double gamma)**

**Призначення**: Обчислює коефіцієнт **k** у заданий момент часу.

**Вхідні параметри**:

1. t – поточний час (тип double)
2. T – кінцевий момент часу (тип double)
3. k0 – початкове значення коефіцієнта (тип double)
4. gamma – коефіцієнт затухання (тип double)

**Вихідне значення**:

Значення **k** у момент часу t (тип double)

**2. double calculate\_e (double t, double T, double e0, double r)**

**Призначення**: Обчислює заряд **e** у момент часу.

**Вхідні параметри**:

1. t – поточний час (тип double)
2. T – загальна тривалість (тип double)
3. e0 – початкове значення заряду (тип double)
4. r – параметр, що визначає зміну заряду з часом (тип double)

**Вихідне значення**:

Значення заряду **e** у момент часу t (тип double)

**3. double calculate\_B (float t, float T, float k)**

**Призначення**: Обчислює магнітну індукцію **B**.

**Вхідні параметри**:

1. t – поточний час (тип float)
2. T – тривалість процесу (тип float)
3. k – коефіцієнт, що залежить від часу (тип float)

**Вихідне значення**:

Магнітна індукція B (тип double)

**4. double calculate\_LorentzForce (double e, double v, double B, double alpha)**

**Призначення**: Обчислює силу Лоренца F за формулою **F=q⋅v⋅B⋅sin(α)**

**Вхідні параметри**:

1. e – заряд (тип double)
2. v – швидкість частинки (тип double)
3. B – магнітна індукція (тип double)
4. alpha – кут між напрямком швидкості та вектором індукції (тип double)

**Вихідне значення**:

Сила Лоренца F (тип double)

# Опис файлів та їх призначення

У процесі виконання курсової роботи було використано три основні файли, кожен з яких виконує конкретну функціональну роль у роботі програми.

**Файл code.c** — це основний вихідний файл програми, написаний мовою С. Він містить повний програмний код, що реалізує розв’язання поставленої задачі. У ньому здійснено:

* опис структури InputData, яка зберігає вхідні параметри для кожного варіанта розрахунку;
* реалізацію користувацьких функцій calculate\_k(), calculate\_e(), calculate\_B() та calculate\_LorentzForce(), які відповідають за фізичні обчислення згідно з математичними моделями;
* зчитування вхідних даних із файлу input.txt;
* проведення циклічних обчислень у заданому діапазоні часу;
* виведення результатів у файл і на екран;
* обробку помилок при відкритті файлів або некоректних даних.

**Файл input.txt** використовується для зберігання вхідних даних, необхідних для виконання обчислень. Його структура є строго визначеною: перший рядок може містити заголовок (назви параметрів), після чого йдуть три рядки з числовими значеннями параметрів для кожного варіанта розрахунку. Для кожного варіанта задається набір наступних параметрів:

T—тривалість моделювання,

dt—крок часу,

k0—базовий коефіцієнт,

γ—параметр зміни коефіцієнта,

e0—початкове значення заряду,

r— параметр зміни заряду,

v — швидкість частинки.

Важливо дотримуватися точної структури файлу input.txt, інакше зчитування даних може завершитися помилкою. Формат має відповідати прикладу, наведеному на рисунку 5. Усі значення повинні бути записані через пробіл.

Зображення, що містить текст, знімок екрана, Шрифт, число

Вміст, створений ШІ, може бути неправильним.

Рисунок 5 – структура Файлуinput.txt

**Файл output.txt** створюється програмою автоматично під час виконання. У нього виводяться результати розрахунків для кожного з трьох варіантів: значення часу t, магнітної індукції B та сили Лоренца F. Ці дані також дублюються у вікно консолі для зручності користувача, що дозволяє оперативно оцінити результати без потреби відкривати додаткові файли. Формат виводу є стандартизованим: для кожного варіанта виводиться заголовок "Variant N", після чого йдуть рядки з обчисленими значеннями фізичних параметрів у відповідному часовому діапазоні.

# Список використаних бібліотек

1. **#include <stdio.h>**

**Призначення:**

Бібліотека стандартного введення/виведення. Забезпечує доступ до функцій роботи з файлами та виведенням у консоль, зокрема:

fopen(), fclose() — відкриття/закриття файлів;

fscanf(), fprintf() — зчитування/запис даних із/у файли;

printf(), perror() — виведення інформації та повідомлень про помилки в консоль.

1. **#include <stdlib.h>**

**Призначення:**

Містить функції для роботи з пам’яттю, перетворення типів та керування виконанням програми. У даній програмі використовується для:

визначення типу exit code у функції return 1; — завершення програми у випадку помилки.

1. **#include <math.h>**

**Призначення:**

Забезпечує математичні функції. У програмі використовується для:

sin() — обчислення синуса в розрахунку сили Лоренца;

exp() — експоненційна функція в розрахунку заряду.

# Інструкція для роботи з програмою

**Для роботи програми необхідно мати:**

Систему з компілятором мови С, наприклад GCC або середовище розробки на кшталт Code::Blocks, Visual Studio, тощо.

Три файли:

code.c — вихідний файл із програмним кодом.

input.txt — текстовий файл із вхідними даними.

output.txt — файл, у який програма виведе результати.

**Кроки для запуску програми:**

1. Створіть файл input.txt у тій самій директорії, що й програма. Його структура обов’язково повинна відповідати формату:

T dt k0 y e0 r v

1 0.05 0.01 0.01 1e-9 0.01 1000

2 0.10 0.04 0.03 2e-9 0.02 1500

3 0.15 0.06 0.02 3e-9 0.03 2000

Перший рядок — заголовок (для зручності читання), далі — три варіанти вхідних даних.

1. Відкриття та запуск програми

Для запуску програми необхідно відкрити файл code.c у Visual Studio Code (разом із файлом input.txt у цій самій папці) або скористатися будь-яким іншим зручним способом, після чого відкрити термінал і запустити програму (у Visual Studio Code це можна зробити комбінацією клавіш **Ctrl + Alt + N** через розширення Code Runner); після запуску програма зчитає вхідні дані з файлу input.txt, виконає обчислення сили Лоренца для трьох варіантів і виведе результати в консоль, а також збереже їх у файл output.txt.

**Результат роботи:**

Після запуску програми вона зчитує вхідні параметри з файлу input.txt, де задано дані для трьох варіантів розрахунку. Для кожного з варіантів у часовому проміжку від t = 0 до t = T з кроком dt обчислюються:

* коефіцієнт k за заданою формулою;
* заряд e, який змінюється з часом;
* магнітна індукція B;
* сила Лоренца F, яка діє на заряджену частинку в магнітному полі.

Результати обчислень для кожного моменту часу виводяться одночасно на екран (консоль) і зберігаються у файл output.txt. Для кожного варіанта обчислень виводиться заголовок **Variant 1**, **Variant 2**, **Variant 3**, а нижче — відповідні значення часу t, магнітної індукції B та сили Лоренца F. Таким чином, користувач отримує повний звіт про динаміку сили Лоренца в кожному з трьох заданих випадків, як у візуальному (на екрані), так і в текстовому (у файлі) вигляді.

# Приклад тестування та результати роботи програми

Для тестування було використано вхідний файл input.txt з трьома варіантами параметрів. Його вміст:

T dt k0 y e0 r v

1 0.05 0.01 0.01 1e-9 0.01 1000

2 0.10 0.04 0.03 2e-9 0.02 1500

3 0.15 0.06 0.02 3e-9 0.03 2000

Ці дані відповідають трьом окремим випадкам із різними значеннями часу T, кроку dt та фізичних параметрів.

Приклад розрахунку (для першого варіанту)

Для варіанта 1:

T = 1 с, dt = 0.05 с, k0 = 0.01, y = 0.01, e0 = 1e-9, r = 0.01, v = 1000

Формули:

1. k(t) = k0 + y \* t
2. e(t) = e0 \* (1 + t)
3. B(t) = k(t) \* r
4. F(t) = e(t) \* v \* B(t)

Приклад для t = 0.00 с:

* k = 0.01 + 0.01 \* 0 = 0.01
* e = 1e-9 \* (1 + 0) = 1e-9
* B = 0.01 \* 0.01 = 0.0001
* F = 1e-9 \* 1000 \* 0.0001 = 1e-10 Н

Для t = 0.05 с:

* k = 0.01 + 0.01 \* 0.05 = 0.0105
* e = 1e-9 \* (1 + 0.05) = 1.05e-9
* B = 0.0105 \* 0.01 = 0.000105
* F = 1.05e-9 \* 1000 \* 0.000105 = 1.1025e-10 Н

Ці значення формують перші рядки виводу.

Результат роботи програми

Після запуску програми в терміналі (консолі) відображається результат:

Зображення, що містить текст, знімок екрана

Вміст, створений ШІ, може бути неправильним.

Рисунок 6 – вивід результату в консолі

**Обробка помилок**

У програмі передбачено базову систему перевірки помилок, яка дозволяє виявити критичні ситуації ще до початку або під час виконання розрахунків. Зокрема, реалізовано такі перевірки:

**Помилка відкриття файлу input.txt** — на початку виконання програма намагається відкрити файл з вхідними даними. Якщо файл відсутній у поточній директорії або з якоїсь причини не може бути відкритий, на екран виводиться повідомлення:

Зображення, що містить текст, Шрифт, знімок екрана, Графіка

Вміст, створений ШІ, може бути неправильним.

Рисунок 7 – вивід помилки відкриття файлу input.txt

Після цього програма завершує виконання з кодом помилки 1.

**Недостатня кількість варіантів у вхідному файлі** — програма розрахована на обробку трьох варіантів даних. Якщо вхідний файл містить менше трьох повних наборів чисел, програма виводить повідомлення:

Зображення, що містить текст, Шрифт, знімок екрана

Вміст, створений ШІ, може бути неправильним.

Рисунок 8 – вивід помилки недостатня кількість варіантів у вхідному файлі

І завершує роботу. Це запобігає помилкам при доступі до неініціалізованих елементів масиву.

**Помилка відкриття файлу output.txt** — перед записом результатів програма намагається створити або відкрити файл output.txt для запису. Якщо з якихось причин файл не може бути відкритий для запису (наприклад, через відсутність дозволів), на екран виводиться повідомлення:



Рисунок 9 – вивід помилки відкриття файлу output.txt

І програма завершить роботу з кодом помилки 1.

Ці перевірки дозволяють зробити програму більш надійною та зрозумілою для користувача, забезпечуючи контроль над ситуаціями, які можуть призвести до некоректного виконання або втрати даних.

# Графіки

На основі отриманих результатів обчислень сили Лоренца для трьох варіантів було побудовано графіки залежності сили F від часу t. Для побудови графіків використовувалося програмне забезпечення Microsoft Excel, яке дає змогу візуалізувати зміни фізичних величин у зручному вигляді.

Для кожного варіанта з файлу output.txt було окремо взято значення часу та відповідної сили, після чого побудовано лінійні графіки. Вони демонструють характер змін сили Лоренца впродовж усього часового інтервалу для заданих параметрів.

Графіки дають змогу наочно побачити, як змінюється сила під дією змінної магнітної індукції та інших параметрів, що впливають на неї. Це дозволяє краще зрозуміти фізичну суть процесу та підтверджує правильність роботи програми.

Нижче наведено графіки для кожного з трьох варіантів:

**Графік залежності сили Лоренца від часу для 1 варіанту:**

**Графік залежності сили Лоренца від часу для 2 варіанту:**

**Графік залежності сили Лоренца від часу для 1 варіанту:**

**Висновки**

У процесі виконання курсової роботи було досягнуто поставленої мети — розроблено програму мовою С для розрахунку сили, що діє на заряджені частинки в магнітному полі за законом Лоренца. Програму реалізовано з використанням структурованого підходу до обчислень, обробки вхідних даних і збереження результатів. Це дозволило на практиці закріпити знання з мови С, зокрема в частині роботи з масивами структур, файлами та математичними функціями.

Отримані результати:

* Обробка вхідних даних: Забезпечено зчитування параметрів із текстового файлу input.txt для трьох незалежних варіантів експерименту, що імітує реальне завантаження даних у системи обробки.
* Розрахунки за фізичними формулами: Реалізовано функції для обчислення коефіцієнта **k**, електричного заряду **e**, магнітної індукції **B** та сили Лоренца **F** (F=q⋅v⋅B⋅sin(α)). Обчислення проводяться у часовому циклі від 0 до T з кроком, заданим параметром dt, що дозволяє моделювати процес у динаміці.
* Виведення результатів: Результати для кожного моменту часу виводяться одночасно на екран (у консоль) та записуються у файл output.txt. Це дозволяє зберігати дані для подальшого аналізу або побудови графіків.
* Обробка помилок: Реалізовано базову обробку помилок при відкритті файлів і перевірку достатності даних у вхідному файлі. У разі виникнення проблем користувачу надається відповідне повідомлення.
* Візуалізація: На основі отриманих числових даних побудовано графіки залежності сили Лоренца від часу для кожного з варіантів у середовищі Microsoft Excel, що дає змогу наочно оцінити зміну сили в залежності від параметрів.

Практична значущість роботи полягає в можливості використання програми для моделювання дії магнітного поля на заряджені частинки, що є актуальним у фізичних дослідженнях, електродинаміці та інженерних застосуваннях. Запропоновану програму можна вдосконалити шляхом розширення кількості варіантів, урахування додаткових параметрів або візуалізації результатів безпосередньо в середовищі С.

Ця курсова робота стала основою для поглиблення знань у галузі прикладного програмування на мові С та набула навичок розробки науково-технічного програмного забезпечення.

# Список літературних джерел

1. Шпак З. Я. Програмування мовою С : навчальний посібник / З. Я. Шпак. — Львів : Оріяна-Нова, 2011. — 432 с. — Режим доступу: <https://catalog.lounb.org.ua/bib/392358>
2. Методичні вказівки до курсової роботи з дисципліни «Програмування» / уклад.: В. В. Авраменко, В. О. Боровик, Н. В. Тиркусова. — Суми : Сумський державний університет, 2021. — 43 с.
3. Керниган Б., Рітчі Д. Мова програмування С. — М. : Вільямс, 2001. — 272 с. — Режим доступу: <http://programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
4. Ришковець Ю. В., Висоцька В. А. Алгоритмізація та програмування. Ч. 1 [Електронний ресурс] : навч. посіб. — Львів : Новий Світ-2000, 2018. — 337 с. — Режим доступу: <http://lib.sumdu.edu.ua>

# Додаток

**Код:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define PI 3.141592653589793

#define NUM\_VARIANTS 3

#define MAX\_LINE\_LENGTH 256

typedef *struct* {

*double* T;

*double* dt;

*double* k0;

*double* y;

*double* e0;

*double* r;

*double* v;

} InputData;

*double* calculate\_k(*double* *t*, *double* *T*, *double* *k0*, *double* *y*) {

    if (*t* <= *T* / 2.0)

        return *k0* \* (1 + *y*);

    else if (*t* <= *T*)

        return *k0* \* (1 - *y*);

    else {

        return 0.0;

    }

}

*double* calculate\_e(*double* *t*, *double* *T*, *double* *e0*, *double* *r*) {

    if (*t* <= *T* / 2.0)

        return *e0* \* (1 + exp(-*r* \* *t*));

    else if (*t* <= *T*)

        return *e0* \* (1 + exp(-*r* \* *T* / 2.0));

    else {

        return 0.0;

    }

}

*double* calculate\_B(*float* *t*, *float* *T*, *float* *k0*) {

    if (*t* <= *T* / 4.0)

        return *k0* \* *t*;

    else if (*t* <= 3 \* *T* / 4.0)

        return (*k0* \* *T* / 4.0) - *k0* \* (*t* - *T* / 4.0);

    else {

        return 0.0;

    }

}

*double* calculate\_LorentzForce(*double* *e*, *double* *v*, *double* *B*, *double* *a*) {

    return *e* \* *v* \* *B* \* sin(*a*);

}

*int* main() {

    FILE \*fin = fopen("input.txt", "r");

    if (!fin) {

        perror("Failed to open input.txt");

        return 1;

    }

*char* line[MAX\_LINE\_LENGTH];

    fgets(line, sizeof(line), fin);

    InputData data[NUM\_VARIANTS];

*int* count = 0;

    while (count < NUM\_VARIANTS &&

           fscanf(fin, "%lf %lf %lf %lf %lf %lf %lf",

                  &data[count].T,

                  &data[count].dt,

                  &data[count].k0,

                  &data[count].y,

                  &data[count].e0,

                  &data[count].r,

                  &data[count].v) == 7) {

        count++;

    }

    fclose(fin);

    if (count != NUM\_VARIANTS) {

        fprintf(stderr, "Error: exactly %d variants are required.\n", NUM\_VARIANTS);

        return 1;

    }

    FILE \*fout = fopen("output.txt", "w");

    if (!fout) {

        perror("Failed to open output.txt");

        return 1;

    }

    for (*int* i = 0; i < NUM\_VARIANTS; i++) {

        fprintf(fout, "Variant %d:\n", i + 1);

        printf("Variant %d:\n", i + 1);

        for (*double* t = 0; t <= data[i].T; t += data[i].dt) {

*double* k = calculate\_k(t, data[i].T, data[i].k0, data[i].y);

*double* e = calculate\_e(t, data[i].T, data[i].e0, data[i].r);

*double* B = calculate\_B(t, data[i].T, k);

*double* alpha = PI / 2.0;

*double* F = calculate\_LorentzForce(e, data[i].v, B, alpha);

            fprintf(fout, "t = %.2f, B = %.6f, F = %.12f\n", t, B, F);

            printf("t = %.2f, B = %.6f, F = %.12f\n", t, B, F);

        }

        fprintf(fout, "\n");

        printf("\n");

    }

    fclose(fout);

    return 0;

}