

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 Володимир Миколайович

Студент Лихацький Денис Сергійович

Група КН-42/2

Варіант №8

Суми - 2025

## ЗМІСТ

<b>1</b>	<b>Постановка задачі.....</b>	<b>4</b>
1.1	Мета курсової роботи .....	4
1.2	Номер варіанта завдання, виданий викладачем .....	4
1.3	Основні дії, які виконує програма.....	5
1.4	Вхідні дані програми .....	5
1.5	Основні результати обчислень (вихідні дані) .....	6
<b>2</b>	<b>Теоретичний матеріал із теми.....</b>	<b>7</b>
2.1	Короткі відомості про мову програмування C .....	7
2.2	Основні визначення та концепції, використані при розробці програми .	7
2.3	Опис стандартних функцій, що використовувалися в програмі .....	8
<b>3</b>	<b>Опис структури даних та вимог до них .....</b>	<b>9</b>
3.1	Вхідні дані.....	9
3.2	Вимоги до вхідних даних та діапазон можливих значень.....	10
3.3	Структури даних, використані у програмі.....	11
<b>4</b>	<b>Опис функцій користувача.....</b>	<b>12</b>
4.1	Функція: calculate_S_projected_area .....	12
4.2	Функція: calculate_v_velocity .....	12
4.3	Функція: calculate_Re_reynolds_number .....	13
4.4	Функція: calculate_phi_drag_coefficient .....	13
4.5	Функція: calculate_F_drag_force .....	13
<b>5</b>	<b>Опис файлів та їх призначення .....</b>	<b>14</b>
5.1	coursera_drag.c.....	14
5.2	input.txt.....	14
5.3	output.txt.....	15
5.4	coursera_drag.exe .....	15
<b>6.</b>	<b>Список використаних бібліотек. ....</b>	<b>16</b>
<b>7.</b>	<b>Інструкція для роботи з програмою .....</b>	<b>17</b>
7.1	Необхідні ресурси .....	17
7.2	Підготовка до запуску.....	17
7.3	Компіляція програми .....	17

7.4 Запуск програми.....	18
7.5 Взаємодія з програмою та запити .....	18
7.6 Перевірка даних, передбачена програмою .....	19
7.6.1 Перевірка відкриття файлів: .....	19
7.6.2 Перевірка коректності читання даних з input.txt:.....	19
8 Приклад тестування та результати роботи програми .....	20
8.1 Вхідні дані які використовувалися для роботи програми.....	20
8.2 Процес тестування.....	20
8.3 Результати роботи програми .....	20
8.4 Приклади розрахунків (контрольна перевірка).....	22
9 Графіки .....	23
9.1 Графік 1 .....	23
9.2 Графік 2.....	23
9.3 Графік 3.....	23
10 Висновки .....	24
11. Список літератури.....	25

# 1 Постановка задачі

## 1.1 Мета курсової роботи

Метою виконання курсової роботи є закріплення та поглиблення теоретичних знань, оволодіння практичними навичками щодо розроблення алгоритмів розв'язування задач за математичним описом для вирішення конкретних завдань, а також їх реалізація мовою програмування C, закріплення навичок самостійної роботи шляхом пошуку, аналізу та активного засвоювання інформації із різних джерел[1].

## 1.2 Номер варіанта завдання, виданий викладачем

**Варіант №8:** Описати масив структур із трьох елементів. Кожна структура об'єднує дані для одного варіанта розрахунку. Необхідно для кожного варіанта на відрізку від 0 до T з кроком  $\Delta t$  побудувати графік зміни сили опору F, що чиниться середовищем до частинки, яка в ній рухається[1].

$$F = \varphi S \frac{\rho v^2}{2},$$

Де: d - діаметр частинки;

$\rho$  - густина середовища;

$\mu$  - в'язкість середовища.

$$\text{Конкретно } \varphi = \frac{24}{R_e} \text{ для } R_e \leq 2; \quad \varphi = \frac{18,5}{R_e^{0,6}} \text{ для } 2 < R_e \leq 500; \\ \varphi = 0,44 \text{ для } 500 < R_e \leq 2 \cdot 10^5.$$

Швидкість v змінюється з часом:

$$v = \begin{cases} v_0 + \frac{4A}{T}t \text{ для } t \in [0, \frac{T}{4}], \\ v_0 + A \text{ для } t \in [\frac{T}{4}, \frac{3T}{4}], \\ v_0 + A - (t - \frac{3}{4}T) \frac{4A}{T} \text{ для } t \in [\frac{3T}{4}, T], \end{cases}$$

де 0 v A, - задані константи.

Вхідні дані зчитуються з файлу. Результати обчислень занести до іншого файлу. Передбачити окремі функції для обчислення  $R_e, S, \phi, v$ .

Вхідні дані:

1.  $T = 600$  с,  $\Delta t = 20$  с,  $\rho = 1,29$  кг/м<sup>3</sup>,  $A = 4$  м/с,  $v_0 = 6$  м/с,  
 $\mu = 17,3 \cdot 10^{-6} \frac{Нс}{м^2}$ ,  $d = 1 \cdot 10^{-3}$  м.
2.  $A = 5$  м/с,  $v_0 = 8$  м/с,  $d = 2 \cdot 10^{-3}$  м. Решта даних – див. пункт 1.
3.  $A = 7$  м/с,  $v_0 = 10$  м/с,  $d = 3 \cdot 10^{-3}$  м. Решта даних – див. пункт 1.

Рисунок 1 - Вхідні дані

### 1.3 Основні дії, які виконує програма

**Розроблена програма виконує такі основні дії:**

Спочатку зчитує дані з вхідного файлу «input.txt» параметри для трьох варіантів розрахунку руху частинки в середовищі.

Для кожного варіанта розрахунку:

Ітерує по часу  $t$  від 0 до заданого кінцевого часу  $T$  з визначеним кроком  $\Delta t$ .

На кожному кроці часу обчислює: поточну швидкість частинки  $v(t)$  згідно з кусково-заданою функцією; проекційну площу частинки  $S$  (виходячи з її діаметра  $d$ , для сферичної частинки  $S = \pi d^2 / 4$ ); число Рейнольдса  $R_e$ ; коефіцієнт аеродинамічного спротиву  $\phi$  на основі розрахованого  $R_e$ ; силу опору середовища  $F$ . Потім записує пари значень (поточний час  $t$ , розрахована сила  $F$ ) у вихідний файл (output.txt) для подальшої побудови графіка. Обчислення величин  $S, v, R_e, \phi$  реалізовані в окремих функціях для модульності коду.

### 1.4 Вхідні дані програми

Вхідними даними для програми є текстовий файл «input.txt», який містить три рядки. Кожен рядок відповідає одному варіанту розрахунку та містить такі параметри, розділені пробілами[1]:

- $T$  - загальний час розрахунку, с;
- $\Delta t$  - крок інтегрування по часу, с;
- $\rho$  - густина середовища, кг/м<sup>3</sup>;
- $\mu$  - динамічна в'язкість середовища, Нс/м<sup>2</sup>;
- $A$  - константа для розрахунку швидкості, м/с;

- $v_0$  - початкова швидкість частинки, м/с;
- $d$  - діаметр частинки, м.

Згідно з моїм варіантом №8, використовуються такі набори вхідних даних:

1.  $T=600, \Delta t=20, \rho=1.29, \mu=17.3 \cdot 10^{-6}, A=4, v_0=6, d=1 \cdot 10^{-3}$ .
2.  $A=5, v_0=8, d=2 \cdot 10^{-3}$  (інші параметри як у п.1).
3.  $A=7, v_0=10, d=3 \cdot 10^{-3}$  (інші параметри як у п.1)."

### 1.5 Основні результати обчислень (вихідні дані)

**Основними результатами роботи програми є:**

- Розраховані значення сили опору середовища  $F$  для кожного кроку часу  $t$  для кожного з трьох варіантів вхідних даних.
- Проміжні величини, що обчислюються для розрахунку  $F$ : поточна швидкість  $v(t)$ , проекційна площа  $S$ , число Рейнольдса  $Re$ , коефіцієнт спротиву  $\phi$ .
- Дані для побудови графіка залежності сили опору  $F$  від часу  $t$ .

Результати (пари значень час  $t$  та сила  $F$ ) для кожного варіанта записуються у вихідний текстовий файл output.txt у форматі для побудови графіків.

## **2 Теоретичний матеріал із теми**

### **2.1 Короткі відомості про мову програмування C**

Мова C, створена Деннісом Рітчі в Bell Labs на початку 1970-х років для операційної системи UNIX, є компілювальною мовою процедурного програмування. Вона здобула широке визнання завдяки своїй ефективності, швидкодії, можливостям низькорівневого програмування та портативності[6]. Наявність розширеної стандартної бібліотеки спрощує виконання багатьох типових завдань. Ці характеристики роблять мову C придатною для розв'язання науково-технічних задач, включно із завданням даної курсової роботи, де важливі точність обчислень та контроль над процесом їх виконання[6].

### **2.2 Основні визначення та концепції, використані при розробці програми**

При розробці програми мною було застосовано ключові елементи мови C. Для зберігання даних використовувалися змінні; зокрема, тип `double` який забезпечував необхідну точність для фізичних величин, а тип `int` - для цілочисельних значень, таких як лічильники[4]. Для об'єднання пов'язаних параметрів кожного варіанта розрахунку (час, крок, густина, в'язкість тощо) було визначено структуру `VariantParameters`. Оскільки завдання передбачало обробку кількох варіантів, я використовував масив таких структур[4].

Програмний код було організовано у вигляді окремих функцій, що відповідають за конкретні обчислювальні задачі (розрахунок площі, швидкості, числа Рейнольдса, коефіцієнта спротиву та сили опору). Логіка програми реалізована за допомогою керуючих конструкцій: цикли `for` використовувалися для ітерації по варіантах та часових кроках, а умовні оператори `if-else if-else` - для реалізації розгалужень в алгоритмах обчислення швидкості та коефіцієнта спротиву. Взаємодія з файловою системою для читання вхідних параметрів та запису результатів здійснювалася через механізми файлового введення/виведення, використовуючи потоки типу `FILE*`. Також застосовувалися директиви

препроцесора `#include` для підключення стандартних бібліотек та `#define` для визначення констант[2,3,4].

### 2.3 Опис стандартних функцій, що використовувалися в програмі

Програма активно використовує функції стандартної бібліотеки C, підключені через заголовкові файли `<stdio.h>` (для введення/виведення) та `<math.h>` (для математичних обчислень)[4].

З `<stdio.h>` були задіяні такі функції для роботи з файлами: `fopen()` для відкриття файлів у потрібному режимі (читання "r" або запис "w"), `fclose()` для їх закриття, `fscanf()` для зчитування форматованих даних з вхідного файлу, та `fprintf()` для запису форматованих результатів у вихідний файл та виведення повідомлень про помилки. Для виведення інформації на консоль використовувалася функція `printf()`, а для відображення системних повідомлень про помилки файлових операцій - `perror()`.

З математичної бібліотеки `<math.h>` використовувалися: константа `M_PI` для отримання значення числа  $\pi$ , функція `pow()` для піднесення до степеня (зокрема, при обчисленні  $R_e^{0.6}$ ), та функція `fabs()` для знаходження абсолютного значення числа з плаваючою комою, що було корисним при порівнянні дійсних чисел у циклі обробки часу[4].



### 3 Опис структури даних та вимог до них

#### 3.1 Вхідні дані

Програма отримує вхідні параметри з текстового файлу input.txt. Для кожного з трьох варіантів розрахунку, зчитується набір параметрів, що характеризують умови руху частинки та властивості середовища. Ці параметри для кожного варіанту зберігаються у програмі в екземплярі структури VariantParameters

Ім'я параметра у формулі / Змістовне ім'я	Змінна у програмі (в структурі VariantParameters)	Тип змінної	Призначення
Загальний час $T$	T_total_time	double	Загальний час моделювання руху частинки
Крок часу $\Delta t$	dt_time_step	double	Крок інтегрування по часу, визначає дискретність розрахунків.
Густина середовища $\rho$	rho_density	double	Густина середовища, в якому рухається частинка.
В'язкість середовища $\mu$	mu_viscosity	double	Динамічна в'язкість середовища.
Константа $A$ (для швидкості)	A_const	double	Константа для розрахунку швидкості частинки.
Початкова швидкість $v_0$	v0_initial_velocity	double	Початкова швидкість частинки в момент часу $t=0$ .
Діаметр частинки $d$	d_diameter	double	Діаметр сферичної частинки.

### 3.2 Вимоги до вхідних даних та діапазон можливих значень

Усі вхідні параметри є дійсними числами (double). Для коректної роботи програми та фізичної осмисленості результатів, передбачаються такі обмеження на діапазони значень:

- Загальний час  $T$ : має бути позитивним ( $T > 0$ ).
- Крок часу  $\Delta t$ : має бути позитивним та, бажано, значно меншим за  $T$  ( $0 < \Delta t \ll T$ ) для забезпечення достатньої кількості точок розрахунку.
- Густина середовища  $\rho$ : має бути позитивною ( $\rho > 0$ ).
- В'язкість середовища  $\mu$ : має бути позитивною ( $\mu > 0$ ).
- Діаметр частинки  $d$ : має бути позитивним ( $d > 0$ ).
- Константа  $A$  та початкова швидкість  $v_0$ : можуть бути нульовими або позитивними. У контексті задачі (рух вперед, опір середовища) зазвичай розглядаються невід'ємні значення. Конкретні значення для варіантів наведені в постановці задачі.

Результати обчислень - пари значень (поточний час  $t$ , сила опору  $F$ ) - для кожного варіанта записуються у вихідний текстовий файл output.txt. Цей файл має структуру, де дані для кожного варіанту чітко відокремлені та супроводжуються переліком вхідних параметрів, використаних для даного розрахунку.

Ім'я параметра у формулі / Змістовне ім'я	Змінна у програмі (в структурі VariantParameters)	Тип змінної	Призначення
Поточний час $t$	current_t	double	Поточний момент часу, для якого виконуються розрахунки (змінюється від 0 до $T$ ).
Проекційна площа $S$	$S\_area$ (результат: calculate_S_projected_area)	double	Розрахована проекційна площа частинки на площину, перпендикулярну напрямку руху.
Поточна швидкість $v$	$v$ (результат: calculate_v_velocity)	double	Швидкість частинки відносно середовища в поточний момент часу $t$

Число Рейнольдса $Re$	Re (результат: calculate_Re_reynolds_number)	double	Безрозмірне число Рейнольдса, що характеризує режим потоку.
Коефіцієнт спротиву $\phi$	phi (результат: calculate_phi_drag_coefficient)	double	Розрахований коефіцієнт аеродинамічного спротиву.
Сила опору F	F (результат: calculate_F_drag_force)	double	Кінцева розрахункова величина - сила опору середовища, що діє на частинку.

### 3.3 Структури даних, використані у програмі

Основною користувацькою структурою даних у програмі є VariantParameters, оголошена за допомогою ключового слова struct. Вона призначена для зберігання повного набору вхідних параметрів для одного варіанта розрахунку:

```
typedef struct {
    double T_total_time;
    double dt_time_step;
    double rho_density;
    double mu_viscosity;
    double A_const;
    double v0_initial_velocity;
    double d_diameter;
} VariantParameters;
```

Для зберігання даних усіх трьох варіантів, передбачених завданням, у програмі використовується масив цих структур: VariantParameters variants[NUM\_VARIANTS];, де NUM\_VARIANTS є константою, що дорівнює 3. Такий підхід дозволяє зручно організувати, передавати та обробляти набори параметрів для кожного окремого розрахункового випадку.

## 4 Опис функцій користувача

### 4.1 Функція: `calculate_S_projected_area`

Функція обчислює проекційну площу частинки  $S$  на площину, перпендикулярну напрямку її руху. В рамках даної задачі передбачається, що частинка має сферичну форму.

Вхідні параметри:

- `d_diameter` (double): Діаметр сферичної частинки, для якої розраховується площа.

Вихідний параметр (значення, що повертається):

- double: Розрахована проекційна площа частинки  $S$  (площа кола з діаметром `d_diameter`).

### 4.2 Функція: `calculate_v_velocity`

Функція розраховує швидкість частинки  $v$  відносно середовища в заданий момент часу `t_current_time`. Розрахунок виконується на основі кусково-заданої функції, що залежить від загального часу моделювання  $T$  та констант  $v_0$  і  $A$ .

Вхідні параметри:

- `t_current_time` (double): Поточний момент часу, для якого необхідно обчислити швидкість.
- `v0_initial_velocity` (double): Початкова швидкість частинки  $v_0$ .
- `A_const` (double): Константа  $A$ , що використовується у формулах для розрахунку швидкості.
- `T_total_time` (double): Загальний час моделювання  $T$ , використовується для визначення часових інтервалів у кусково-заданій функції.

Вихідний параметр (значення, що повертається):

- double: Розраховане значення швидкості частинки  $v$  в момент часу `t_current_time`.

#### 4.3 Функція: `calculate_Re_reynolds_number`

Функція обчислює безрозмірне число Рейнольдса  $Re$ , яке характеризує співвідношення інерційних сил до сил в'язкості в потоці середовища навколо частинки.

Вхідні параметри:

- `v_velocity` (double): Поточна швидкість частинки відносно середовища.
- `d_diameter` (double): Характерний лінійний розмір частинки (її діаметр).
- `rho_density` (double): Густина середовища.
- `mu_viscosity` (double): Динамічна в'язкість середовища.

Вихідний параметр (значення, що повертається):

- `double`: Розраховане число Рейнольдса  $Re$ .

#### 4.4 Функція: `calculate_phi_drag_coefficient`

Функція визначає коефіцієнт аеродинамічного (або гідродинамічного) спротиву  $\phi$ . Значення коефіцієнта залежить від числа Рейнольдса  $Re$  і розраховується за різними формулами для різних діапазонів  $Re$ .

Вхідні параметри:

- `Re_reynolds_number` (double): Число Рейнольдса, для якого необхідно визначити коефіцієнт спротиву.

Вихідний параметр (значення, що повертається):

- `double`: Розрахований коефіцієнт спротиву  $\phi$

#### 4.5 Функція: `calculate_F_drag_force`

Вхідні параметри:

- `phi_drag_coefficient` (double): Коефіцієнт спротиву  $\phi$ .
- `rho_density` (double): Густина середовища  $\rho$ .
- `S_projected_area` (double): Проекційна площа частинки  $S$ .
- `v_velocity` (double): Швидкість частинки  $v$  відносно середовища.

## 5 Опис файлів та їх призначення

Для виконання та тестування розробленої програми використовуються такі файли:

### 5.1 coursera\_drag.c

Це основний файл проєкту, що містить вихідний код програми мовою C. В ньому реалізовано всю логіку обчислень: функція `main()`, функції користувача: `calculate_S_projected_area`, `calculate_v_velocity`, `calculate_Re_reynolds_number`, `calculate_phi_drag_coefficient`, `calculate_F_drag_force`, зчитування вхідних даних з файлу та запис результатів у файл.

Стандартна структура файлу з кодом на мові C, що включає директиви препроцесора (`#include`, `#define`), оголошення глобальних констант (якщо є), визначення типів (структура `VariantParameters`), прототипи та реалізації функцій.

### 5.2 input.txt

Вхідний текстовий файл, з якого програма зчитує початкові параметри для розрахунків. Він містить дані для трьох різних варіантів умов задачі.

Файл складається з трьох рядків. Кожен рядок містить сім дійсних чисел, розділених пробілами, що відповідають таким параметрам для одного варіанту:

1.  $T$  (загальний час, с)
2.  $\Delta t$  (крок часу, с)
3.  $\rho$  (густина середовища,  $\text{кг/м}^3$ )
4.  $\mu$  (в'язкість середовища,  $\text{Нс/м}^2$ )
5.  $A$  (константа для швидкості, м/с)
6.  $v_0$  (початкова швидкість, м/с)
7.  $d$  (діаметр частинки, м)

### 5.3 output.txt

Вихідний текстовий файл, у який програма записує результати обчислень. Ці дані призначені для подальшого аналізу та побудови графіків залежності сили опору від часу.

Файл містить результати для кожного з трьох варіантів. Для кожного варіанта спочатку виводяться його вхідні параметри, а потім таблиця з двома колонками: "Час (с)" та "Сила опору F (Н)". Кожен рядок таблиці відповідає одному моменту часу. Дані для різних варіантів відокремлені порожнім рядком та заголовком варіанта. Приклад фрагменту для одного варіанта(Рисунок 2):

```
1  Варіант 1:
2  Параметри: T=600.00, dt=20.00, rho=1.29, mu=1.730000e-05, A=4.00, v0=6.00, d=0.0010
3  Час (с)   Сила опору F (Н)
4  -----
5  0.00      0.000009
6  20.00     0.000010
7  40.00     0.000011
```

Рисунок 2 - Приклад фрагменту для одного варіанта

### 5.4 coursera\_drag.exe

Скомпільований виконуваний файл програми. Саме цей файл запускається для виконання обчислень. Він створюється компілятором C з вихідного коду (coursera\_drag.c).

Бінарний файл у форматі, специфічному для операційної системи.

## 6. Список використаних бібліотек.

### 1. <stdio.h>(Стандартна бібліотека введення/виведення)

Ця бібліотека надає функції для виконання операцій введення та виведення даних. У програмі вона використовувалася для роботи з файлами (читання вхідних параметрів з файлу input.txt та запис результатів у файл output.txt), а також для виведення повідомлень на консоль. Основні використані функції: fopen(), fclose(), fscanf(), fprintf(), printf(), perror()[4,3,2].

### 2. <math.h> (Математична бібліотека)

Ця бібліотека містить набір функцій для виконання математичних обчислень. У програмі вона була необхідна для розрахунку степеня (при обчисленні коефіцієнта  $\phi$ ) та для використання математичних констант (число  $\pi$ ). Основні використані функції/константи: pow(), fabs(), M\_PI (або acos() для обчислення  $\pi$ )[4].

### 3. <stdlib.h> (Стандартна загальна бібліотека)

Ця бібліотека містить функції загального призначення, включаючи управління пам'яттю, перетворення типів, генерацію випадкових чисел та інші утиліти. У даній програмі вона могла бути підключена для загальної сумісності або для функцій, які не є строго специфічними для введення/виведення чи математики, хоча її явне використання у наданому коді було мінімальним (наприклад, exit() не використовувався, але stdlib.h часто включають за замовчуванням або для майбутніх розширень)[4].

Підключення цих бібліотек здійснювалося за допомогою директиви препроцесора #include на початку файлу з вихідним кодом програми.



## 7. Інструкція для роботи з програмою

### 7.1 Необхідні ресурси

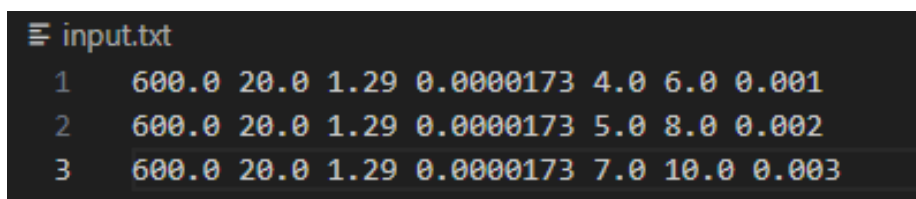
Компілятор мови C: Для компіляції вихідного коду програми необхідний C-компілятор. Рекомендується використовувати GCC, який є вільно доступним для більшості операційних систем (Linux, macOS, Windows (через MinGW або WSL)). Можливе використання інших C-компіляторів, наприклад, Clang або компілятора, вбудованого в інтегровані середовища розробки (IDE) наприклад Visual Studio Code.

Текстовий редактор або IDE: Для перегляду та, за потреби, модифікації вихідного коду (.c файлу) та вхідного файлу даних (input.txt).

### 7.2 Підготовка до запуску

Вихідний код: Сам файл з кодом (Kursova.c)

Вхідний файл input.txt: Створений файл у тій самій директорії, де буде компілюватися та запускатися програма, файл input.txt. Цей файл має містити три рядки даних, де кожен рядок відповідає одному варіанту розрахунку та містить сім дійсних чисел, розділених пробілами ( $T$ ,  $\Delta t$ ,  $\rho$ ,  $\mu$ ,  $A$ ,  $v_0$ ,  $d$ )(Рисунок 3).

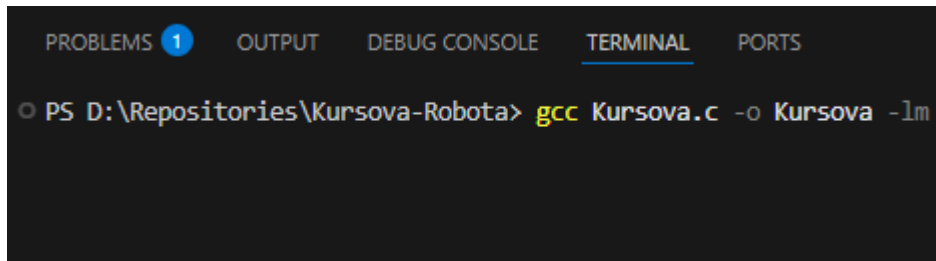


```
input.txt
1 600.0 20.0 1.29 0.0000173 4.0 6.0 0.001
2 600.0 20.0 1.29 0.0000173 5.0 8.0 0.002
3 600.0 20.0 1.29 0.0000173 7.0 10.0 0.003
```

Рисунок 3 - Приклад вмісту input.txt

### 7.3 Компіляція програми

Якщо в програмі для запису коду є встановлене розширення для компіляції коду, треба натиснути комбінацію клавіш Ctrl+Alt+N. Якщо немає встановленого розширення тоді треба відкрити термінал або командний рядок у директорії, де знаходиться файл Kursova.c. Потім ввести команду для компіляції. Для gcc команда виглядатиме так(Рисунок 4):

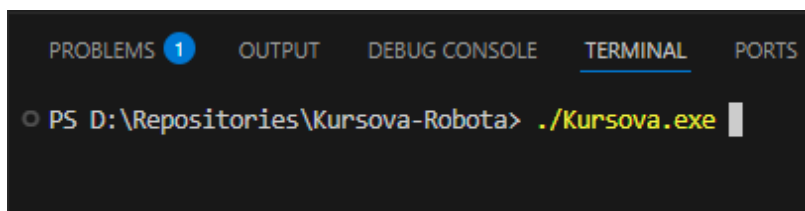


```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Repositories\Kursova-Robota> gcc Kursova.c -o Kursova -lm
```

Рисунок 4 - Вигляд команди для компіляції

## 7.4 Запуск програми

З встановленим розширенням запуск програми здійснюється автоматично з компіляцією. Без розширення, після успішної компіляції, програма запускається з терміналу або командного рядка командою(Рисунок 5):



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Repositories\Kursova-Robota> ./Kursova.exe
```

Рисунок 5 - Команда для запуску коду

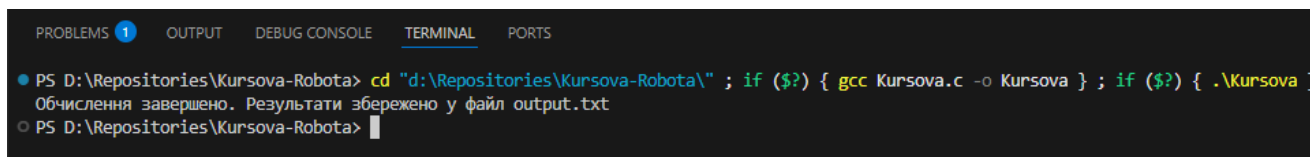
## 7.5 Взаємодія з програмою та запити

Програма не передбачає інтерактивної взаємодії з користувачем під час її виконання. Усі необхідні вхідні дані зчитуються з файлу input.txt на початку роботи.

Після запуску програма виконає обчислення та:

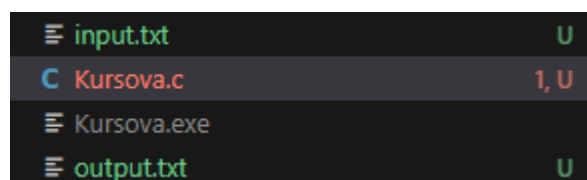
1.Створить файл output.txt у тій самій директорії, куди будуть збережені результати розрахунків(Рисунок 7).

2.Виведе на консоль повідомлення про завершення роботи та ім'я файлу з результатами(Рисунок 6).



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Repositories\Kursova-Robota> cd "d:\Repositories\Kursova-Robota\" ; if ($?) { gcc Kursova.c -o Kursova } ; if ($?) { .\Kursova.exe }
Обчислення завершено. Результати збережено у файл output.txt
PS D:\Repositories\Kursova-Robota>
```

Рисунок 6 - Повідомлення про завершення роботи



input.txt	U
Kursova.c	1, U
Kursova.exe	
output.txt	U

Рисунок 7 - створений output.txt

## 7.6 Перевірка даних, передбачена програмою

Програма виконує такі основні перевірки вхідних даних та умов виконання:

### 7.6.1 Перевірка відкриття файлів:

При спробі відкрити вхідний файл input.txt для читання: якщо файл не знайдено або доступ до нього неможливий, програма виведе повідомлення про помилку і завершить роботу.

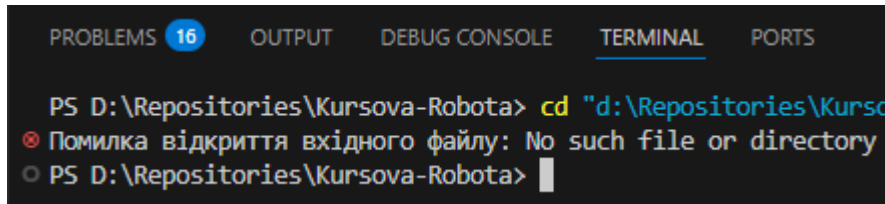
A screenshot of a terminal window with a dark background. At the top, there are tabs: 'PROBLEMS' (with a blue circle containing the number 16), 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), and 'PORTS'. The terminal shows a command prompt 'PS D:\Repositories\Kursova-Robota>' followed by the command 'cd "d:\Repositories\Kursova-Robota\input.txt"'. Below the command, a red error icon is followed by the text 'Помилка відкриття вхідного файлу: No such file or directory'. The prompt then shows 'PS D:\Repositories\Kursova-Robota>' with a cursor.

Рисунок 8 - Приклад помилки якщо вхідний файл не знайдено

### 7.6.2 Перевірка коректності читання даних з input.txt:

Програма очікує зчитати рівно 7 дійсних чисел для кожного з трьох варіантів. Якщо кількість зчитаних значень для будь-якого рядка не відповідає очікуваній (наприклад, у рядку менше чисел або дані некоректного формату), програма виведе повідомлення про помилку читання даних (наприклад, "Помилка читання даних для варіанту X з вхідного файлу.") і завершить роботу.

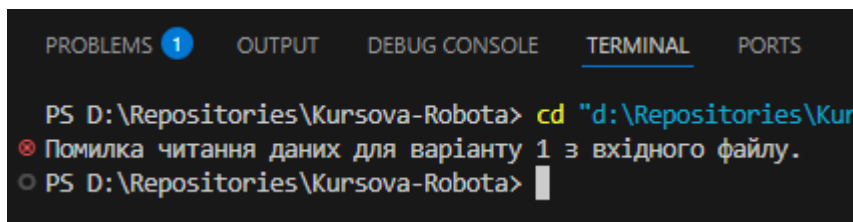
A screenshot of a terminal window with a dark background. At the top, there are tabs: 'PROBLEMS' (with a blue circle containing the number 1), 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), and 'PORTS'. The terminal shows a command prompt 'PS D:\Repositories\Kursova-Robota>' followed by the command 'cd "d:\Repositories\Kursova-Robota\input.txt"'. Below the command, a red error icon is followed by the text 'Помилка читання даних для варіанту 1 з вхідного файлу.'. The prompt then shows 'PS D:\Repositories\Kursova-Robota>' with a cursor.

Рисунок 9 - Помилка якщо файл input.txt некоректного формату

## 8 Приклад тестування та результати роботи програми

### 8.1 Вхідні дані які використовувалися для роботи програми

Тестування програми проводилося на основі трьох наборів вхідних параметрів (Рисунок 1). Ці дані були підготовлені у файлі input.txt згідно з таким форматом: T (загальний час),  $\Delta t$  (крок часу),  $\rho$  (густина середовища),  $\mu$  (в'язкість середовища), A (константа для швидкості),  $v_0$  (початкова швидкість), d (діаметр частинки).

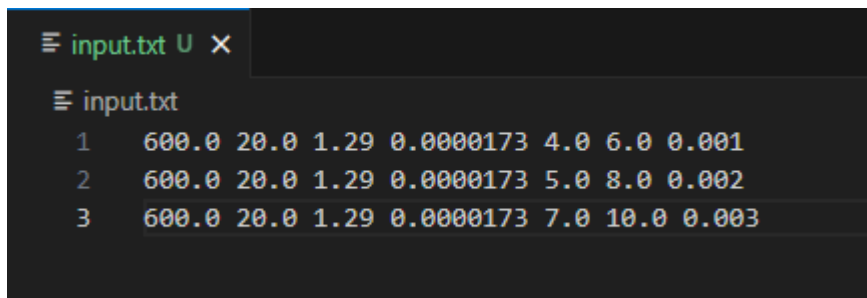


Рисунок 10 - Файл Input.txt

### 8.2 Процес тестування

Як демонструвалось раніше (Рисунок 6) програма була скомпільована за допомогою компілятора та після успішної компіляції програма буде запущена на виконання. Вхідні дані автоматично зчитувалися з файлу input.txt (Рисунок 10)

### 8.3 Результати роботи програми

Основними результатами роботи програми є розраховані значення сили опору середовища F в залежності від часу t для кожного з трьох варіантів вхідних даних. Всі розраховані значення часу та відповідної сили опору для кожного варіанта зберігаються у файлі output.txt (Рисунок 11) (Рисунок 12) (Рисунок 13)

```
output.txt x
output.txt
1  Варіант 1:
2  Параметри: T=600.00, dt=20.00, rho=1.29, mu=1.730000e-05, A=4.00, v0=6.00, d=0.0010
3  Час (с) Сила опору F (Н)
4  -----
5  0.00      0.000009
6  20.00     0.000010
7  40.00     0.000011
8  60.00     0.000013
9  80.00     0.000015
10 100.00    0.000017
11 120.00    0.000019
12 140.00    0.000021
13 160.00    0.000022
14 180.00    0.000022
15 200.00    0.000022
16 220.00    0.000022
17 240.00    0.000022
18 260.00    0.000022
19 280.00    0.000022
20 300.00    0.000022
21 320.00    0.000022
22 340.00    0.000022
23 360.00    0.000022
24 380.00    0.000022
25 400.00    0.000022
26 420.00    0.000022
27 440.00    0.000022
28 460.00    0.000021
29 480.00    0.000019
30 500.00    0.000017
31 520.00    0.000015
32 540.00    0.000013
33 560.00    0.000011
34 580.00    0.000010
35 600.00    0.000009
```

Рисунок 11 - Результат роботи коду (Варіант 1)

```
37  Варіант 2:
38  Параметри: T=600.00, dt=20.00, rho=1.29, mu=1.730000e-05, A=5.00, v0=8.00, d=0.0020
39  Час (с) Сила опору F (Н)
40  -----
41  0.00      0.000057
42  20.00     0.000067
43  40.00     0.000078
44  60.00     0.000089
45  80.00     0.000101
46  100.00    0.000115
47  120.00    0.000128
48  140.00    0.000143
49  160.00    0.000151
50  180.00    0.000151
51  200.00    0.000151
52  220.00    0.000151
53  240.00    0.000151
54  260.00    0.000151
55  280.00    0.000151
56  300.00    0.000151
57  320.00    0.000151
58  340.00    0.000151
59  360.00    0.000151
60  380.00    0.000151
61  400.00    0.000151
62  420.00    0.000151
63  440.00    0.000151
64  460.00    0.000143
65  480.00    0.000128
66  500.00    0.000115
67  520.00    0.000101
68  540.00    0.000089
69  560.00    0.000078
70  580.00    0.000067
71  600.00    0.000057
```

Рисунок 12 - Результат роботи коду (Варіант 2)

```

73  Варіант 3:
74  Параметри: T=600.00, dt=20.00, rho=1.29, mu=1.730000e-05, A=7.00, v0=10.00, d=0.0030
75  Час (с)   Сила опору F (Н)
76  -----
77  0.00      0.000201
78  20.00     0.000240
79  40.00     0.000282
80  60.00     0.000329
81  80.00     0.000378
82  100.00    0.000432
83  120.00    0.000488
84  140.00    0.000548
85  160.00    0.000580
86  180.00    0.000580
87  200.00    0.000580
88  220.00    0.000580
89  240.00    0.000580
90  260.00    0.000580
91  280.00    0.000580
92  300.00    0.000580
93  320.00    0.000580
94  340.00    0.000580
95  360.00    0.000580
96  380.00    0.000580
97  400.00    0.000580
98  420.00    0.000580
99  440.00    0.000580
100 460.00    0.000548
101 480.00    0.000488
102 500.00    0.000432
103 520.00    0.000378
104 540.00    0.000329
105 560.00    0.000282
106 580.00    0.000240
107 600.00    0.000201

```

Рисунок 13 - Результат роботи коду (Варіант 3)

#### 8.4 Приклади розрахунків (контрольна перевірка)

$$v(0)=v_0=6 \text{ м/с.}$$

$$S=\pi d^2/4 \approx 7.854 \cdot 10^{-7} \text{ м}^2.$$

$$R_e=(v d \rho)/\mu=(6 \cdot 0.001 \cdot 1.29)/(17.3 \cdot 10^{-6}) \approx 447.4.$$

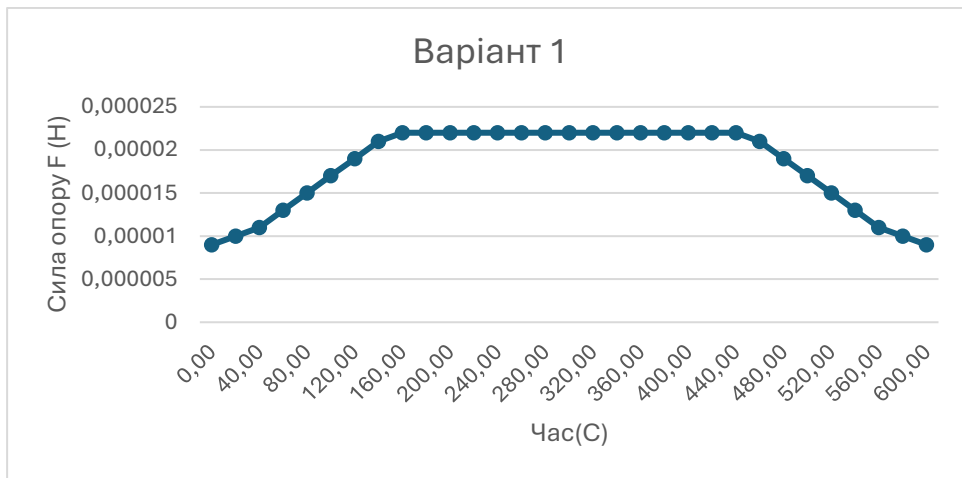
$$\text{Оскільки } 2 < R_e \leq 500, \text{ то } \phi = 18.5/R_e^{0.6} \approx 18.5/(447.4)^{0.6} \approx 0.509.$$

$$F=\phi \rho S(v^2/2) \approx 0.509 \cdot 1.29 \cdot 7.854 \cdot 10^{-7} \cdot (6^2/2) \approx 9.29 \cdot 10^{-6} \text{ Н.}$$

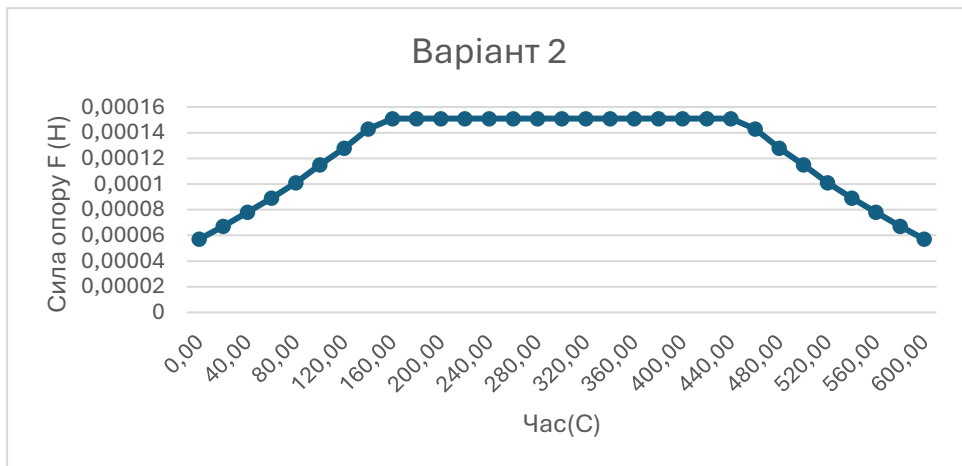
Це значення узгоджується з результатом програми (0.0000009Н у файлі output.txt )

## 9 Графіки

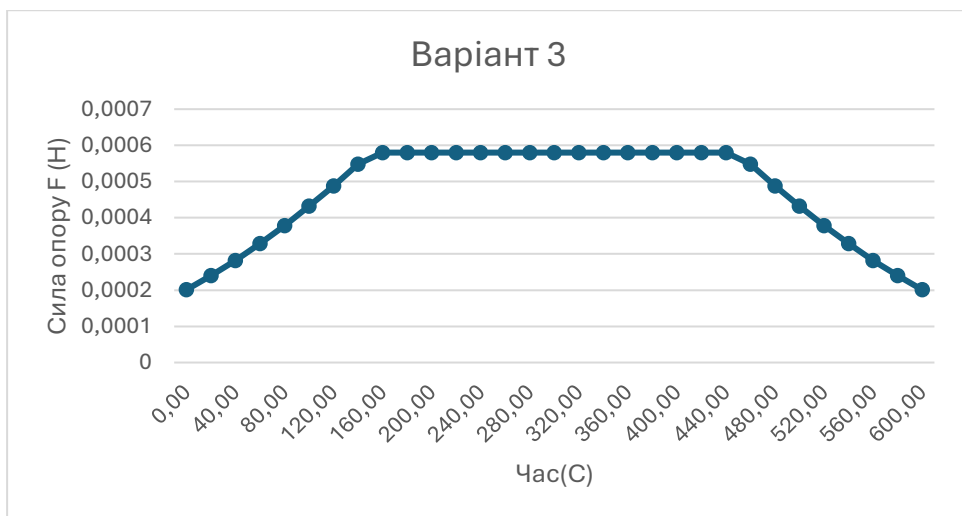
### 9.1 Графік 1



### 9.2 Графік 2



### 9.3 Графік 3



[5]

## 10 Висновки

В ході роботи було реалізовано комплексний алгоритм, що дозволяє обчислювати силу опору середовища  $F$  для трьох різних сценаріїв, які відрізняються вхідними параметрами. Ключовим аспектом програми є покроковий розрахунок у часі, де для кожного моменту  $t$  від початкового до кінцевого значення  $T$  з кроком  $\Delta t$  послідовно визначаються всі необхідні фізичні величини. Спочатку обчислюється проекційна площа частинки  $S$ , виходячи з її діаметра  $d$ . Потім, згідно з наданою кусково-заданою функцією, розраховується поточна швидкість частинки  $v(t)$ , яка залежить від початкової швидкості  $v_0$ , константи  $A$  та загального часу моделювання  $T$ . На основі цих даних визначається число Рейнольдса  $Re$ , яке, у свою чергу, використовується для знаходження коефіцієнта аеродинамічного спротиву  $\phi$  за допомогою відповідних умовних формул. Кінцевим результатом на кожному кроці є обчислення сили опору  $F$  за основною гідродинамічною формулою. Для зручності використання та гнучкості налаштування експерименту, програма зчитує всі необхідні вхідні параметри з текстового файлу `input.txt` і зберігає результати обчислень - часові ряди значень сили  $F$  - у вихідний файл `output.txt`. Це забезпечує легкість подальшого аналізу та візуалізації даних[1].

Для досягнення поставлених завдань були використані стандартні можливості мови програмування C. Розробка велася із застосуванням процедурного підходу. З метою покращення структури коду та його читабельності, основні обчислювальні блоки були реалізовані у вигляді окремих функцій користувача[1]. Для організації наборів вхідних параметрів для кожного варіанту розрахунку ефективно використовувалися структури даних (`struct`), а для їх зберігання - масиви. Керування логікою програми, зокрема ітераційні процеси та реалізація умовних залежностей, здійснювалося за допомогою циклів `for` та операторів розгалуження `if-else if-else`.



## 11. Список літератури

1. Авраменко В. В., Боровик В. О., Тиркусова Н. В. Методичні вказівки до курсової роботи з дисципліни «Програмування» Суми : Сумський державний університет, 2021. – 43 с  
URL: <https://lecturedsumdu.blob.core.windows.net/nodes/6338/90273e8c-cbf8-11ee-99fa-81ae961fb068/5028-1.pdf>
2. Авраменко В. В., Лекція 8 Функції, Суми : Сумський державний університет, 2021. URL: <https://mix.sumdu.edu.ua/textbooks/104505/1296347/index.html>
3. Авраменко В. В., Лекція 11 Стандартні функції Суми : Сумський державний університет, 2021.  
URL: <https://mix.sumdu.edu.ua/textbooks/104505/1296350/index.html>
4. Керніган Б. В., Річі Д. М. Цибуляк В., Мова програмування C / Друге видання, 2012. (с. 3-35) (с. 37-59) (с. 61-74) (с. 75-103) (с. 145-171)  
URL: [https://programming.in.ua/files/programming/book/book\\_programming\\_c\\_kernighan.rar](https://programming.in.ua/files/programming/book/book_programming_c_kernighan.rar)
5. Чибіряк Я.І., Курс Організація та обробка електронної інформації Лекція\_2 ОтаОЕІ, 2024 URL: <https://youtu.be/h0CbVNLf7aM>
6. Вікіпедія, C (мова програмування), 2025  
URL: [https://uk.wikipedia.org/wiki/C\\_\(мова\\_програмування\)](https://uk.wikipedia.org/wiki/C_(мова_програмування))