**1. Flight Trajectory Calculation**

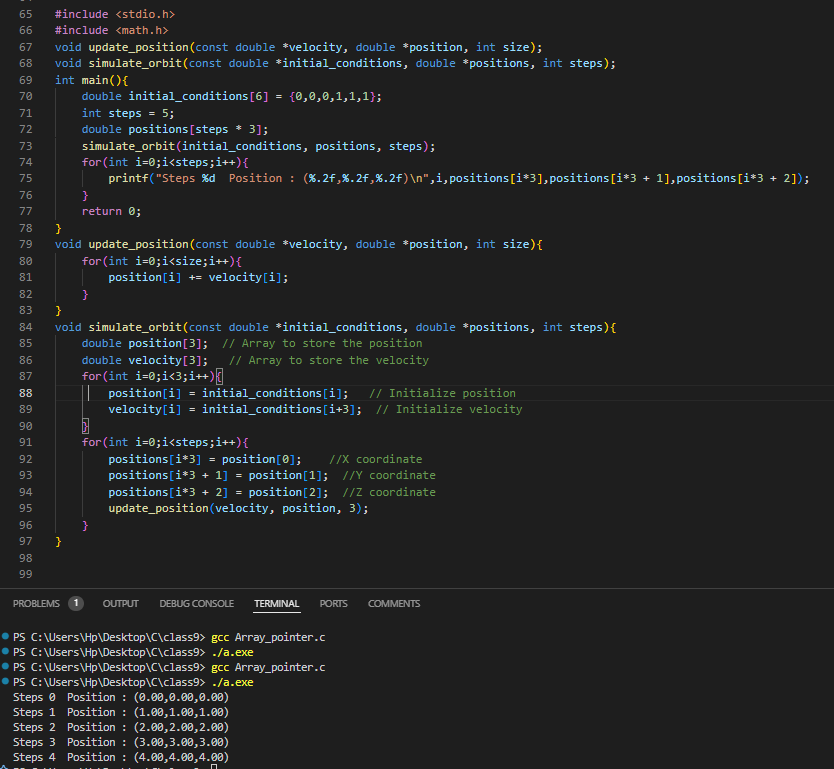
* **Pointers**: Use to traverse the trajectory array.
* **Arrays**: Store trajectory points (x, y, z) at discrete time intervals.
* **Functions**:
  + void calculate\_trajectory(const double \*parameters, double \*trajectory, int size): Takes the initial velocity, angle, and an array to store trajectory points.
  + void print\_trajectory(const double \*trajectory, int size): Prints the stored trajectory points.
* **Pass Arrays as Pointers**: Pass the trajectory array as a pointer to the calculation function.

A screen shot of a computer program

Description automatically generated

**2. Satellite Orbit Simulation**

* **Pointers**: Manipulate position and velocity vectors.
* **Arrays**: Represent the satellite's position over time as an array of 3D vectors.
* **Functions**:
  + void update\_position(const double \*velocity, double \*position, int size): Updates the position based on velocity.
  + void simulate\_orbit(const double \*initial\_conditions, double \*positions, int steps): Simulates orbit over a specified number of steps.
* **Pass Arrays as Pointers**: Use pointers for both velocity and position arrays.



**3. Weather Data Processing for Aviation**

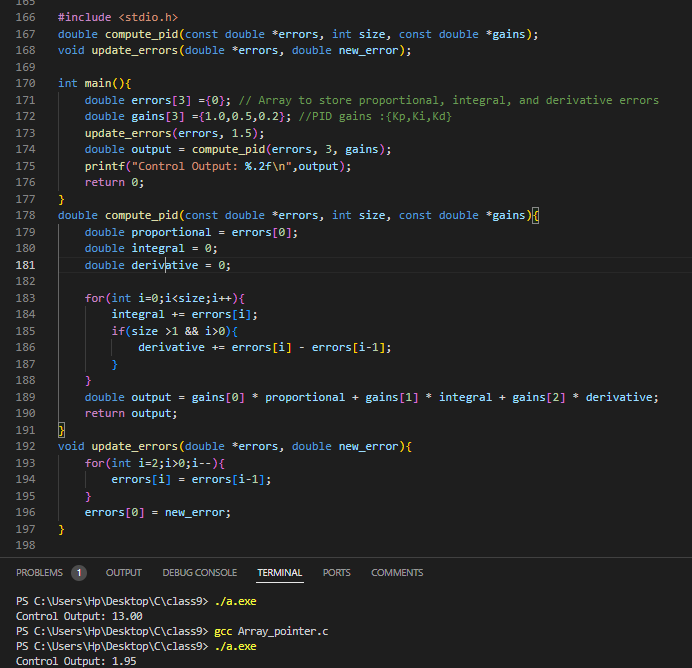
* **Pointers**: Traverse weather data arrays efficiently.
* **Arrays**: Store hourly temperature, wind speed, and pressure.
* **Functions**:
  + void calculate\_daily\_averages(const double \*data, int size, double \*averages): Computes daily averages for each parameter.
  + void display\_weather\_data(const double \*data, int size): Displays data for monitoring purposes.
* **Pass Arrays as Pointers**: Pass weather data as pointers to processing functions.

A screen shot of a computer

Description automatically generated

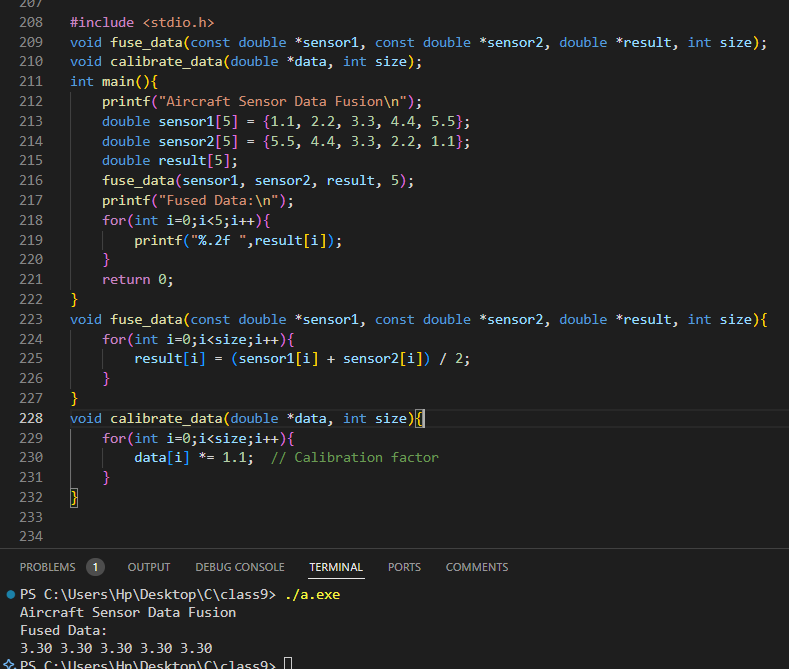
**4. Flight Control System (PID Controller)**

* **Pointers**: Traverse and manipulate error values in arrays.
* **Arrays**: Store historical error values for proportional, integral, and derivative calculations.
* **Functions**:
  + double compute\_pid(const double \*errors, int size, const double \*gains): Calculates control output using PID logic.
  + void update\_errors(double \*errors, double new\_error): Updates the error array with the latest value.
* **Pass Arrays as Pointers**: Use pointers for the errors array and the gains array.



**5. Aircraft Sensor Data Fusion**

* **Pointers**: Handle sensor readings and fusion results.
* **Arrays**: Store data from multiple sensors.
* **Functions**:
  + void fuse\_data(const double \*sensor1, const double \*sensor2, double \*result, int size): Merges two sensor datasets into a single result array.
  + void calibrate\_data(double \*data, int size): Adjusts sensor readings based on calibration data.
* **Pass Arrays as Pointers**: Pass sensor arrays as pointers to fusion and calibration functions.



**6. Air Traffic Management**

* **Pointers**: Traverse the array of flight structures.
* **Arrays**: Store details of active flights (e.g., ID, altitude, coordinates).
* **Functions**:
  + void add\_flight(flight\_t \*flights, int \*flight\_count, const flight\_t \*new\_flight): Adds a new flight to the system.
  + void remove\_flight(flight\_t \*flights, int \*flight\_count, int flight\_id): Removes a flight by ID.
* **Pass Arrays as Pointers**: Use pointers to manipulate the array of flight structures.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

**7. Satellite Telemetry Analysis**

* **Pointers**: Traverse telemetry data arrays.
* **Arrays**: Store telemetry parameters (e.g., power, temperature, voltage).
* **Functions**:
  + void analyze\_telemetry(const double \*data, int size): Computes statistical metrics for telemetry data.
  + void filter\_outliers(double \*data, int size): Removes outliers from the telemetry data array.
* **Pass Arrays as Pointers**: Pass telemetry data arrays to both functions.

A screen shot of a computer program

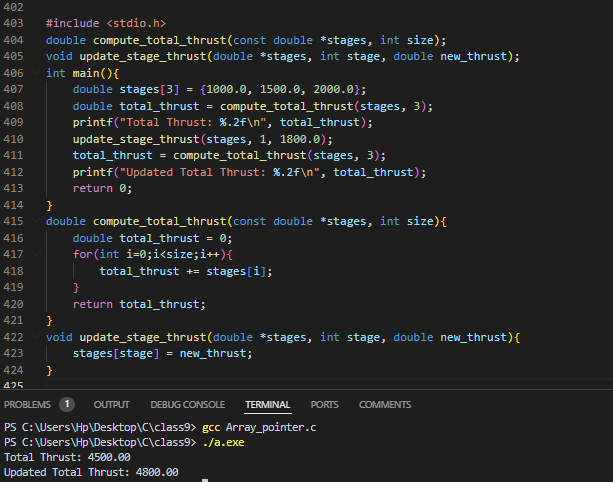
Description automatically generated

A screenshot of a computer

Description automatically generated

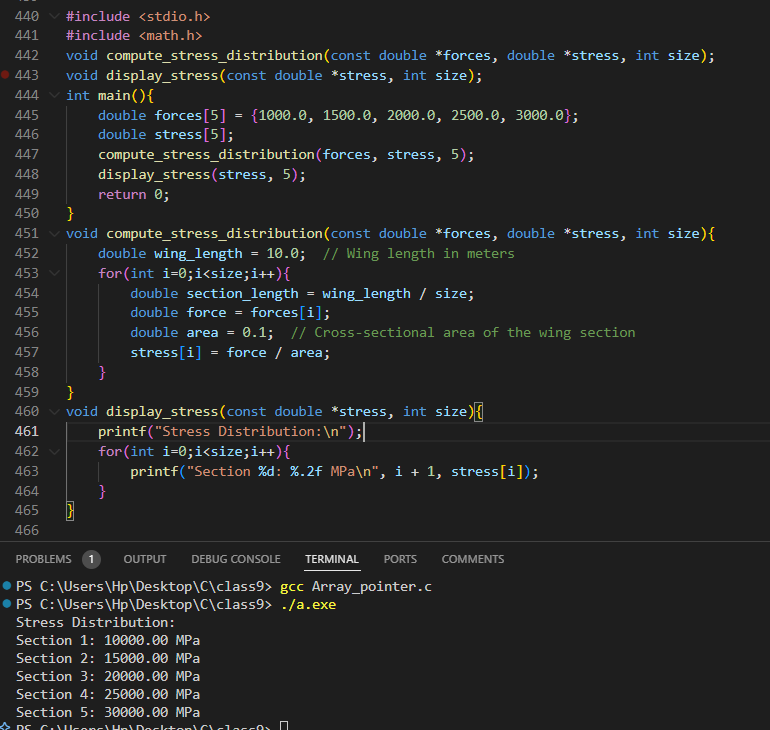
**8. Rocket Thrust Calculation**

* **Pointers**: Traverse thrust arrays.
* **Arrays**: Store thrust values for each stage of the rocket.
* **Functions**:
  + double compute\_total\_thrust(const double \*stages, int size): Calculates cumulative thrust across all stages.
  + void update\_stage\_thrust(double \*stages, int stage, double new\_thrust): Updates thrust for a specific stage.
* **Pass Arrays as Pointers**: Use pointers for thrust arrays.



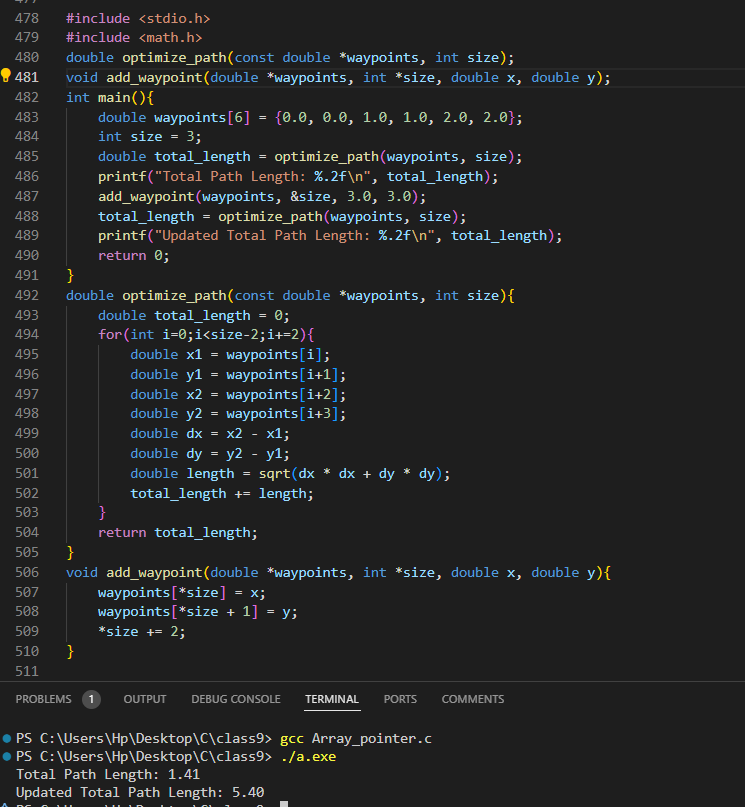
**9. Wing Stress Analysis**

* **Pointers**: Access stress values at various points.
* **Arrays**: Store stress values for discrete wing sections.
* **Functions**:
  + void compute\_stress\_distribution(const double \*forces, double \*stress, int size): Computes stress values based on applied forces.
  + void display\_stress(const double \*stress, int size): Displays the stress distribution.
* **Pass Arrays as Pointers**: Pass stress arrays to computation functions.



**10. Drone Path Optimization**

* **Pointers**: Traverse waypoint arrays.
* **Arrays**: Store coordinates of waypoints.
* **Functions**:
  + double optimize\_path(const double \*waypoints, int size): Reduces the total path length.
  + void add\_waypoint(double \*waypoints, int \*size, double x, double y): Adds a new waypoint.
* **Pass Arrays as Pointers**: Use pointers to access and modify waypoints.



**11. Satellite Attitude Control**

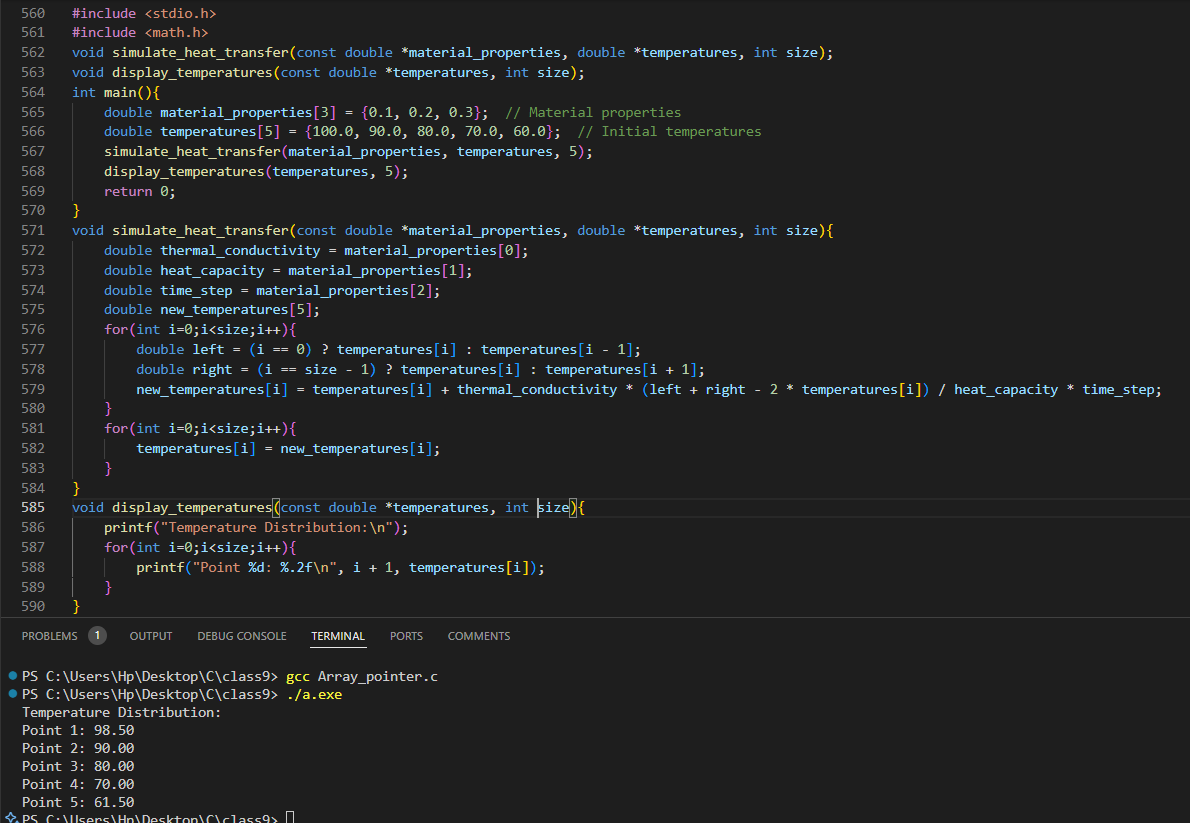
* **Pointers**: Manipulate quaternion arrays.
* **Arrays**: Store quaternion values for attitude control.
* **Functions**:
  + void update\_attitude(const double \*quaternion, double \*new\_attitude): Updates the satellite's attitude.
  + void normalize\_quaternion(double \*quaternion): Ensures quaternion normalization.
* **Pass Arrays as Pointers**: Pass quaternion arrays as pointers.

A screen shot of a computer screen

Description automatically generated

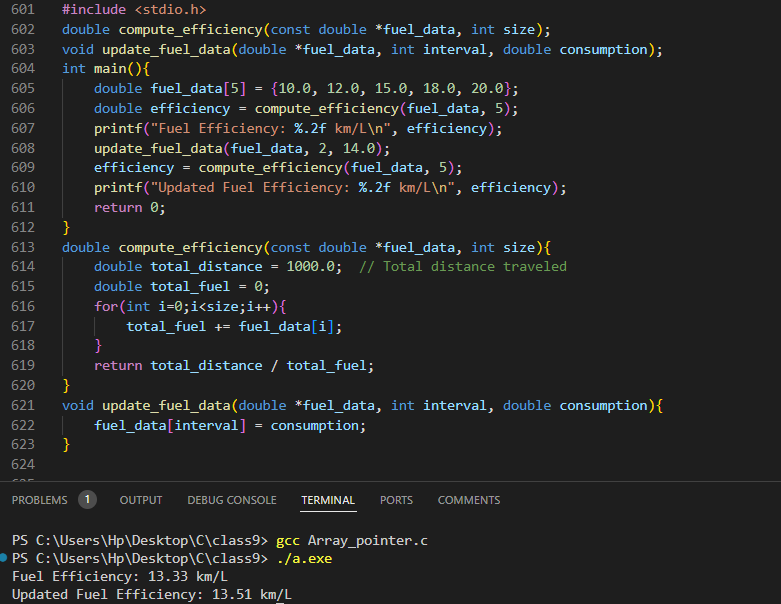
**12. Aerospace Material Thermal Analysis**

* **Pointers**: Access temperature arrays for computation.
* **Arrays**: Store temperature values at discrete points.
* **Functions**:
  + void simulate\_heat\_transfer(const double \*material\_properties, double \*temperatures, int size): Simulates heat transfer across the material.
  + void display\_temperatures(const double \*temperatures, int size): Outputs temperature distribution.
* **Pass Arrays as Pointers**: Use pointers for temperature arrays.



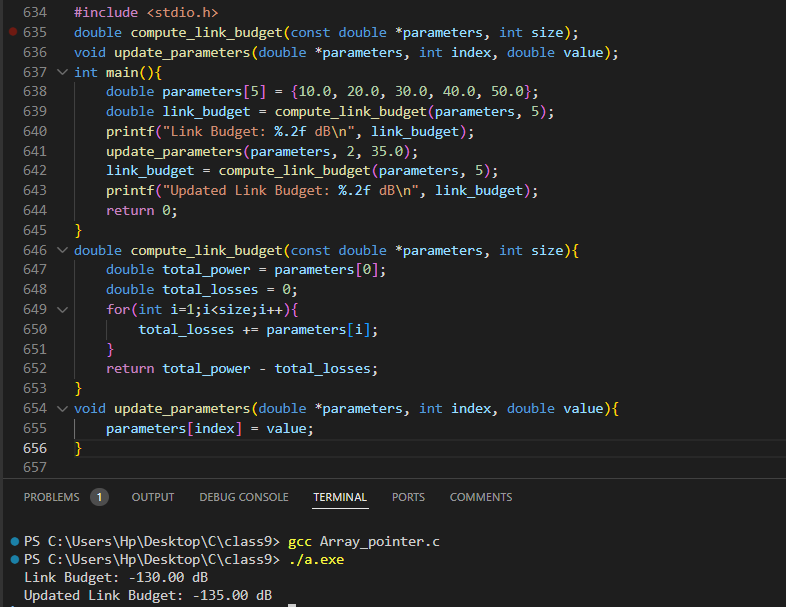
**13. Aircraft Fuel Efficiency**

* **Pointers**: Traverse fuel consumption arrays.
* **Arrays**: Store fuel consumption at different time intervals.
* **Functions**:
  + double compute\_efficiency(const double \*fuel\_data, int size): Calculates overall fuel efficiency.
  + void update\_fuel\_data(double \*fuel\_data, int interval, double consumption): Updates fuel data for a specific interval.
* **Pass Arrays as Pointers**: Pass fuel data arrays as pointers.



**14. Satellite Communication Link Budget**

* **Pointers**: Handle parameter arrays for computation.
* **Arrays**: Store communication parameters like power and losses.
* **Functions**:
  + double compute\_link\_budget(const double \*parameters, int size): Calculates the total link budget.
  + void update\_parameters(double \*parameters, int index, double value): Updates a specific parameter.
* **Pass Arrays as Pointers**: Pass parameter arrays as pointers.



**15. Turbulence Detection in Aircraft**

* **Pointers**: Traverse acceleration arrays.
* **Arrays**: Store acceleration data from sensors.
* **Functions**:
  + void detect\_turbulence(const double \*accelerations, int size, double \*output): Detects turbulence based on frequency analysis.
  + void log\_turbulence(double \*turbulence\_log, const double \*detection\_output, int size): Logs detected turbulence events.
* **Pass Arrays as Pointers**: Pass acceleration and log arrays to functions.

