Assessment

/**

Requirements

1. Define Data Types

Sensor Data Structure:

Define a structure WheelSensor to store the real-time data for each wheel.

Fields:

sensorID (integer): Unique ID for the wheel sensor.

wheelSpeed (float): Speed of the wheel (in RPM).

brakeForce (float): Force applied by the braking system on the wheel (in Newtons).

slipRatio (float): Slip ratio of the wheel (calculated field: (wheelSpeed - vehicleSpeed) / vehicleSpeed).

Union for Error/Status Reporting:

Define a union ABSStatus to store either:

statusMessage (string): A text message describing the system's current state (e.g., "ACTIVE", "INACTIVE", "ERROR").

errorCode (integer): A numerical error code indicating a malfunction.

Vehicle Data Structure:

Define a structure VehicleData to hold:

vehicleSpeed (float): Current speed of the vehicle (in km/h).

isABSActive (integer): 1 if ABS is active, 0 otherwise.

An array of WheelSensor structures for each wheel.

2. Features to Implement

Dynamic Memory Allocation:

Allocate memory dynamically to store an array of WheelSensor structures based on the number of wheels (N wheels).

Input and Output:

Input the details of each wheel sensor, including wheel speed and brake force.

Input the vehicle speed and calculate the slipRatio for each wheel.

Display the details of all wheels, including their calculated slip ratios.

ABS Control Logic:

If the slip ratio for any wheel exceeds a predefined threshold (e.g., 0.2), activate ABS for that wheel and adjust the brake force dynamically to reduce the slip ratio.

Update the isABSActive status in VehicleData.

Error and Status Reporting:

Store and display either a system status message (e.g., "ABS ACTIVE", "NORMAL") or an error code (e.g., 101 for sensor failure) using the ABSStatus union.

Sorting and Analysis:

Sort wheels by their slip ratios in descending order to prioritize adjustments.

Identify the wheel with the highest slip ratio and display its details.

Typedef Usage:

Use typedef to simplify the code for WheelSensor and ABSStatus.

Example Program Flow

Menu-Driven Interface:

Provide a user-friendly menu with options:

Input Wheel Sensor Data

Display All Wheel Data

Monitor and Adjust ABS

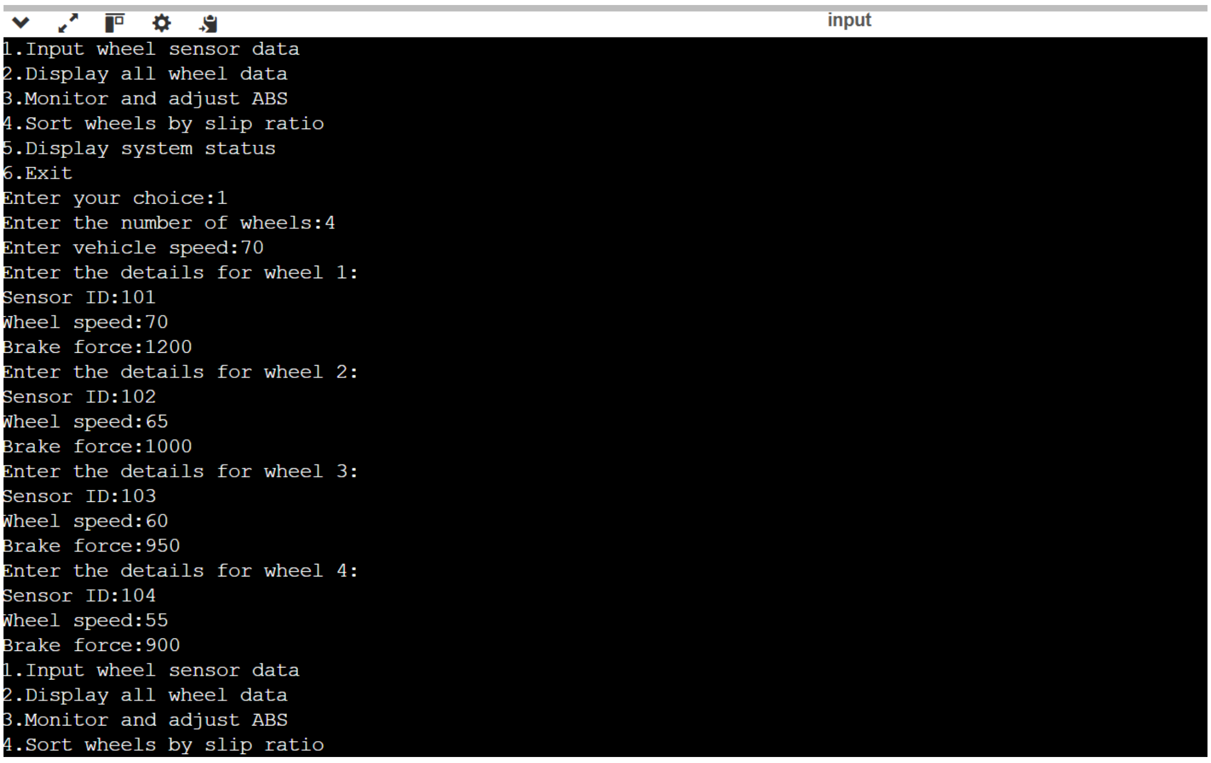
Sort Wheels by Slip Ratio

Display System Status

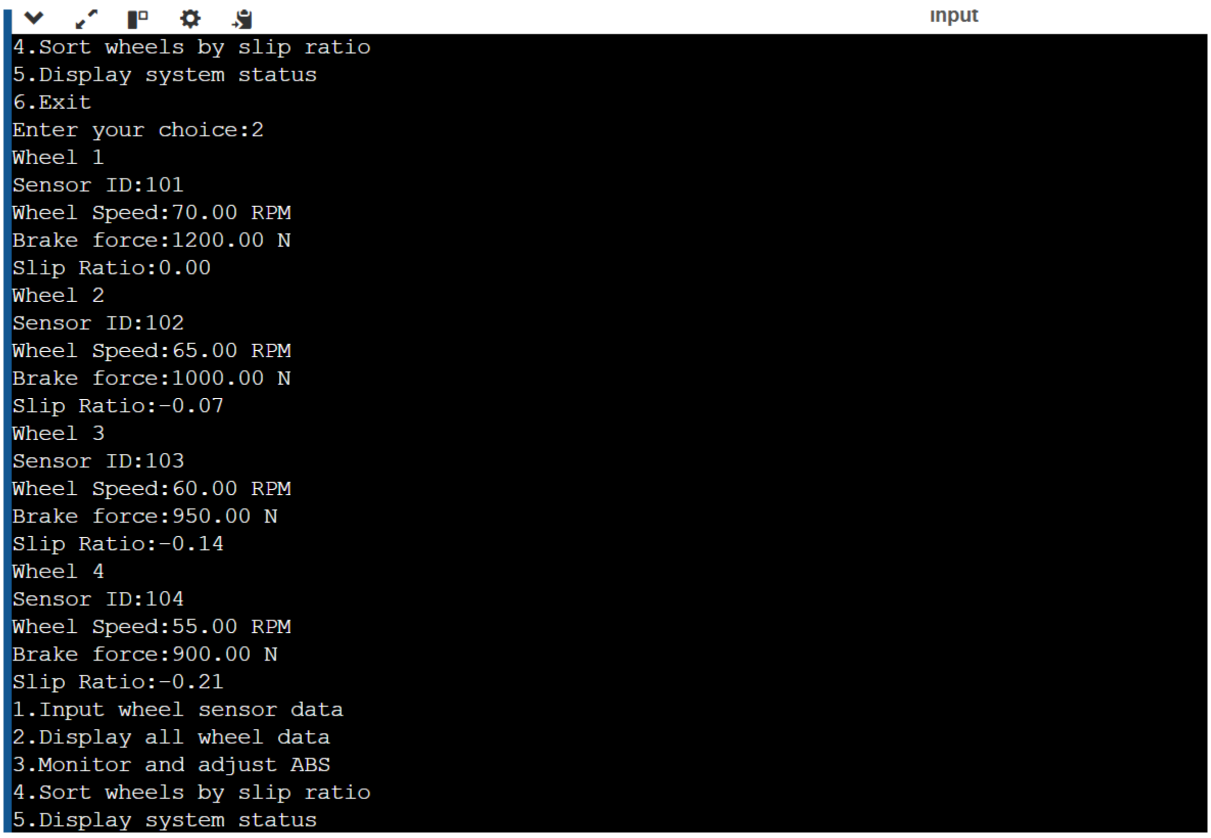
```
******************************************************************************/

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define PREDEFINED_THRESHOLD 0.2

typedef struct

{
    int sensorID;
    float wheelSpeed;
    float brakeForce;
    float slipRatio;
}WheelSensor;

typedef union{
    char statusMessage[50];
    int errorCode;
}ABSStatus;

typedef struct{
    float vehicleSpeed;
    int isABSActive;
    WheelSensor *wheels;
    int numWheels;
}VehicleData;

VehicleData inputWheelSensorData(VehicleData vehicle);
void displayWheelData(VehicleData vehicle);
void monitorAndAdjustAbs(VehicleData *vehicle);
void sortWheelsBySlipRatio(VehicleData *vehicle);
void displaySystemStatus(VehicleData vehicle);
void freeVehicleData(VehicleData *vehicle);
int main()
{
    VehicleData vehicle={0};
    vehicle.wheels=NULL;
    int op;
    while(1)
    {
```

```
printf("1.Input wheel sensor data\n");
printf("2.Display all wheel data\n");
printf("3.Monitor and adjust ABS\n");
printf("4.Sort wheels by slip ratio\n");
printf("5.Display system status\n");
printf("6.Exit\n");
printf("Enter your choice:");
scanf("%d",&op);
switch(op)
{
  case 1:
    vehicle=inputWheelSensorData(vehicle);
    break;
  case 2:
    displayWheelData(vehicle);
    break;
  case 3:
    monitorAndAdjustAbs(&vehicle);
    break;
  case 4:
    sortWheelsBySlipRatio(&vehicle);
    break;
  case 5:
    displaySystemStatus(vehicle);
    break;
  case 6:
    printf("Exit");
    freeVehicleData(&vehicle);
    return 0;
  default:
    printf("Invalid choice");
```

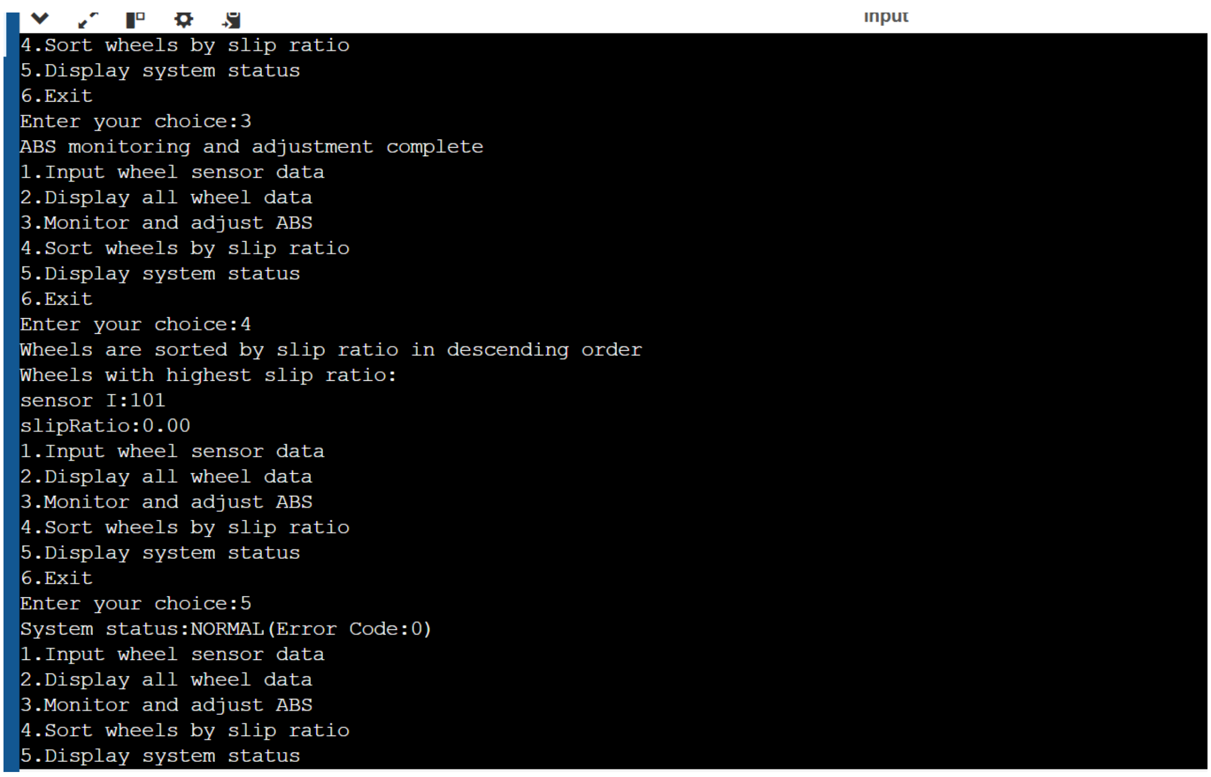
```
        }
    }
    return 0;
}
VehicleData inputWheelSensorData(VehicleData vehicle)
{
    printf("Enter the number of wheels:");
    scanf("%d",&vehicle.numWheels);
    vehicle.wheels=(WheelSensor*)malloc(vehicle.numWheels*sizeof(WheelSensor));
    if(vehicle.wheels==NULL)
    {
        printf("Memory Allocation failed\n");
        exit(1);
    }
    printf("Enter vehicle speed:");
    scanf("%f",&vehicle.vehicleSpeed);
    for(int i=0;i<vehicle.numWheels;i++)
    {
        printf("Enter the details for wheel %d:\n",i+1);
        printf("Sensor ID:");
        scanf("%d",&vehicle.wheels[i].sensorID);
        printf("Wheel speed:");
        scanf("%f",&vehicle.wheels[i].wheelSpeed);
        printf("Brake force:");
        scanf("%f",&vehicle.wheels[i].brakeForce);
        if(vehicle.vehicleSpeed>0)
        {
            vehicle.wheels[i].slipRatio=(vehicle.wheels[i].wheelSpeed-
vehicle.vehicleSpeed)/vehicle.vehicleSpeed;
        }
        else
        {
            vehicle.wheels[i].slipRatio=0;
        }
    }
    return vehicle;
}
void displayWheelData(VehicleData vehicle)
{
    if(vehicle.wheels==NULL)
    {
        printf("No wheel data available");
        return;
    }
    for(int i=0;i<vehicle.numWheels;i++)
    {
        printf("Wheel %d\n",i+1);
        printf("Sensor ID:%d\n",vehicle.wheels[i].sensorID);
        printf("Wheel Speed:%.2f RPM\n",vehicle.wheels[i].wheelSpeed);
        printf("Brake force:%.2f N\n",vehicle.wheels[i].brakeForce);
        printf("Slip Ratio:%.2f\n",vehicle.wheels[i].slipRatio);
    }
}

void monitorAndAdjustAbs(VehicleData *vehicle)
{
    if(vehicle->wheels==NULL)
    {
        printf("No wheel data available \n");
        return;
    }
    vehicle->isABSActive=0;
    for(int i=0;i<vehicle->numWheels;i++)
    {
        if(vehicle->wheels[i].slipRatio>PREDEFINED_THRESHOLD)
        {
            vehicle->isABSActive=1;
            vehicle->wheels[i].brakeForce*=0.8;
            if(vehicle->vehicleSpeed>0)
            {
                vehicle->wheels[i].slipRatio=(vehicle->wheels[i].wheelSpeed-vehicle-
>vehicleSpeed)/vehicle->vehicleSpeed;
            }
        }
    }
    printf("ABS monitoring and adjustment complete \n");
}
void sortWheelsBySlipRatio(VehicleData *vehicle)
{
    if(vehicle->wheels==NULL)
    {
        printf("NO wheel data available \n");
        return;
    }
    for(int i=0;i<vehicle->numWheels-1;i++)
    {
        for(int j=0;j<vehicle->numWheels-i-1;j++)
        {
            if(vehicle->wheels[j].slipRatio<vehicle->wheels[j+1].slipRatio)
            {
                WheelSensor temp=vehicle->wheels[j];
                vehicle->wheels[j]=vehicle->wheels[j+1];
                vehicle->wheels[j+1]=temp;
            }
        }
    }
    printf("Wheels are sorted by slip ratio in descending order \n");
    printf("Wheels with highest slip ratio:\n");
    printf("sensor I:%d\n",vehicle->wheels[0].sensorID);
    printf("slipRatio:%.2f\n",vehicle->wheels[0].slipRatio);
}

void displaySystemStatus(VehicleData vehicle)
{
    ABSStatus status;
    if(vehicle.isABSActive)
    {
        strcpy(status.statusMessage,"ABS ACTIVE");
        printf("System status:%s\n",status.statusMessage);
    }
    else{
        status.errorCode=0;
        printf("System status:NORMAL(Error Code:%d)\n",status.errorCode);
    }
}

void freeVehicleData(VehicleData *vehicle)
{
    if(vehicle->wheels!=NULL)
    {
        free(vehicle->wheels);
        vehicle->wheels=NULL;
    }
}
```



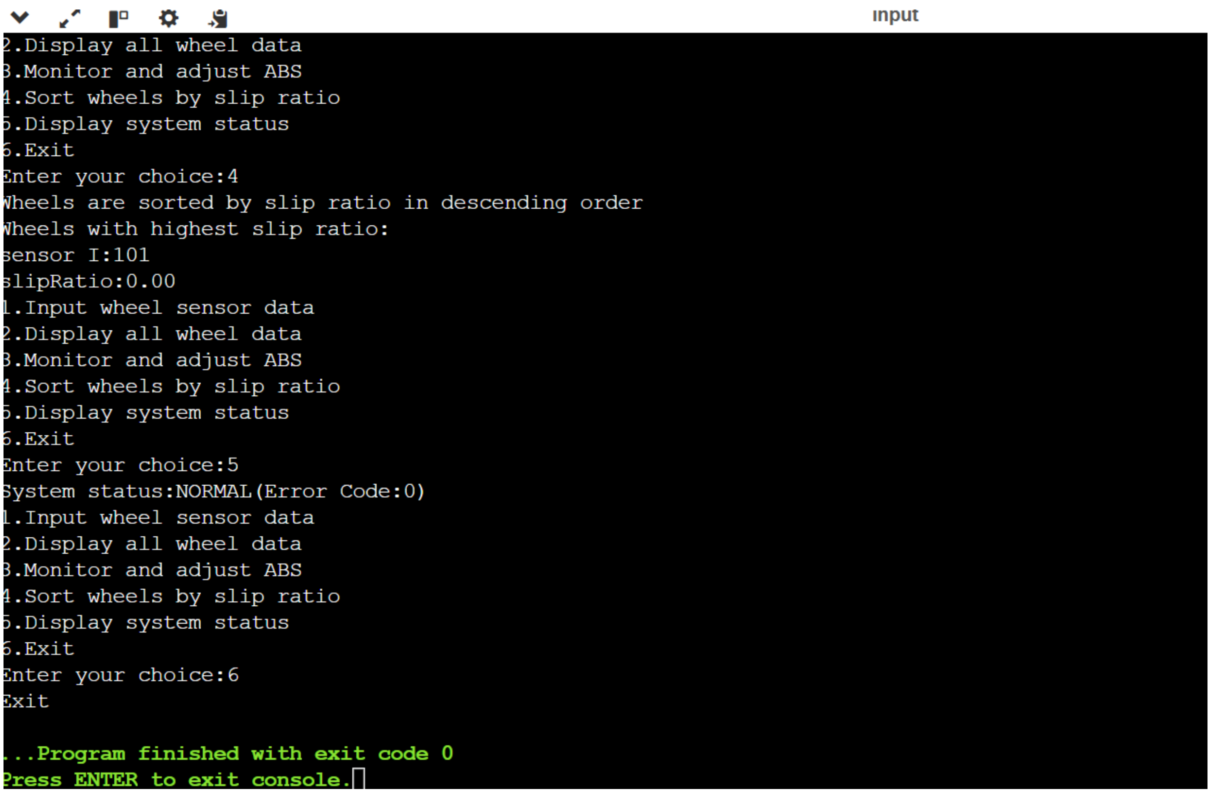
```
1.Input wheel sensor data
2.Display all wheel data
3.Monitor and adjust ABS
4.Sort wheels by slip ratio
5.Display system status
6.Exit
Enter your choice:1
Enter the number of wheels:4
Enter vehicle speed:70
Enter the details for wheel 1:
Sensor ID:101
Wheel speed:70
Brake force:1200
Enter the details for wheel 2:
Sensor ID:102
Wheel speed:65
Brake force:1000
Enter the details for wheel 3:
Sensor ID:103
Wheel speed:60
Brake force:950
Enter the details for wheel 4:
Sensor ID:104
Wheel speed:55
Brake force:900
1.Input wheel sensor data
2.Display all wheel data
3.Monitor and adjust ABS
4.Sort wheels by slip ratio
```



```
4.Sort wheels by slip ratio
5.Display system status
6.Exit
Enter your choice:2
Wheel 1
Sensor ID:101
Wheel Speed:70.00 RPM
Brake force:1200.00 N
Slip Ratio:0.00
Wheel 2
Sensor ID:102
Wheel Speed:65.00 RPM
Brake force:1000.00 N
Slip Ratio:-0.07
Wheel 3
Sensor ID:103
Wheel Speed:60.00 RPM
Brake force:950.00 N
Slip Ratio:-0.14
Wheel 4
Sensor ID:104
Wheel Speed:55.00 RPM
Brake force:900.00 N
Slip Ratio:-0.21
1.Input wheel sensor data
2.Display all wheel data
3.Monitor and adjust ABS
4.Sort wheels by slip ratio
5.Display system status
```



```
4.Sort wheels by slip ratio
5.Display system status
6.Exit
Enter your choice:3
ABS monitoring and adjustment complete
1.Input wheel sensor data
2.Display all wheel data
3.Monitor and adjust ABS
4.Sort wheels by slip ratio
5.Display system status
6.Exit
Enter your choice:4
Wheels are sorted by slip ratio in descending order
Wheels with highest slip ratio:
sensor I:101
slipRatio:0.00
1.Input wheel sensor data
2.Display all wheel data
3.Monitor and adjust ABS
4.Sort wheels by slip ratio
5.Display system status
6.Exit
Enter your choice:5
System status:NORMAL(Error Code:0)
1.Input wheel sensor data
2.Display all wheel data
3.Monitor and adjust ABS
4.Sort wheels by slip ratio
5.Display system status
```



```
2.Display all wheel data
3.Monitor and adjust ABS
4.Sort wheels by slip ratio
5.Display system status
6.Exit
Enter your choice:4
Wheels are sorted by slip ratio in descending order
Wheels with highest slip ratio:
sensor I:101
slipRatio:0.00
1.Input wheel sensor data
2.Display all wheel data
3.Monitor and adjust ABS
4.Sort wheels by slip ratio
5.Display system status
6.Exit
Enter your choice:5
System status:NORMAL(Error Code:0)
1.Input wheel sensor data
2.Display all wheel data
3.Monitor and adjust ABS
4.Sort wheels by slip ratio
5.Display system status
6.Exit
Enter your choice:6
Exit

...Program finished with exit code 0
Press ENTER to exit console.
```