DAY 2

1. EXC1:

SOURCE:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DAY2

{

internal class Car

{

string Model;

int EngineSpeed;

int EnginePower;

double CargoWeight;

string CargoType;

double Tire1Pressure;

int Tire1Age;

double Tire2Pressure;

int Tire2Age;

double Tire3Pressure;

int Tire3Age;

double Tire4Pressure;

int Tire4Age;

public Car() { }

public Car(string model,

int engineSpeed,

int enginePower,

double cargoWeight,

string cargoType,

double tire1Pressure,

int tire1Age,

double tire2Pressure, int tire2Age, double tire3Pressure,

int tire3Age, double tire4Pressure, int tire4Age)

{

Model = model;

EngineSpeed = engineSpeed;

EnginePower = enginePower;

CargoWeight = cargoWeight;

CargoType = cargoType;

Tire1Pressure = tire1Pressure;

Tire1Age = tire1Age;

Tire2Pressure = tire2Pressure;

Tire2Age = tire2Age;

Tire3Pressure = tire3Pressure;

Tire3Age = tire3Age;

Tire4Pressure = tire4Pressure;

Tire4Age = tire4Age;

}

public string CargoTypeGet()

{

return this.CargoType;

}

public int EnginePowerGet()

{

return this.EnginePower;

}

public double Tire1PressureGet()

{

return this.Tire1Pressure;

}

public double Tire2PressureGet()

{

return this.Tire2Pressure;

}

public double Tire3PressureGet()

{

return this.Tire3Pressure;

}

public double Tire4PressureGet()

{

return this.Tire4Pressure;

}

public override string ToString()

{

return this.Model;

}

public void checkCar(string check)

{

if (check.Equals(this.CargoTypeGet()))

{

switch (check)

{

case "fragile":

if ((this.Tire1Pressure < 1)

|| (this.Tire2Pressure < 1)

|| (this.Tire3Pressure < 1)

|| (this.Tire4Pressure < 1)) Console.WriteLine(this.ToString());

break;

case "flamable":

if (this.EnginePower > 250) Console.WriteLine(this.ToString());

break;

}

}

}

public Car Add(string temp)

{

Car car = new Car();

string[] temp2 = new string[100];

temp2 = temp.Split(' ');

car = new Car(

temp2[0],

int.Parse(temp2[1]),

int.Parse(temp2[2]),

double.Parse(temp2[3]),

temp2[4],

double.Parse(temp2[5]),

int.Parse(temp2[6]),

double.Parse(temp2[7]),

int.Parse(temp2[8]),

double.Parse(temp2[9]),

int.Parse(temp2[10]),

double.Parse(temp2[11]),

int.Parse(temp2[12]));

return car;

}

}

}

static void Problem1()

{

int n = int.Parse(Console.ReadLine());

Car[] Car = new Car[n];

for (int i = 0; i < n; i++)

{

string temp = Console.ReadLine();

Car[i] = Car[i].Add(temp);

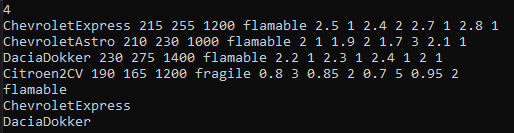
}

string check = Console.ReadLine();

for (int i = 0; i < n; i++) Car[i].checkCar(check);

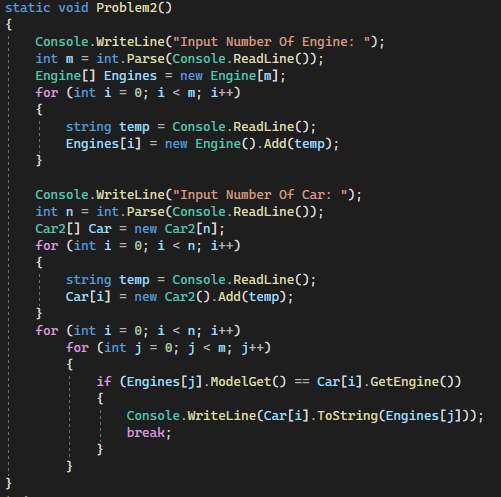
}

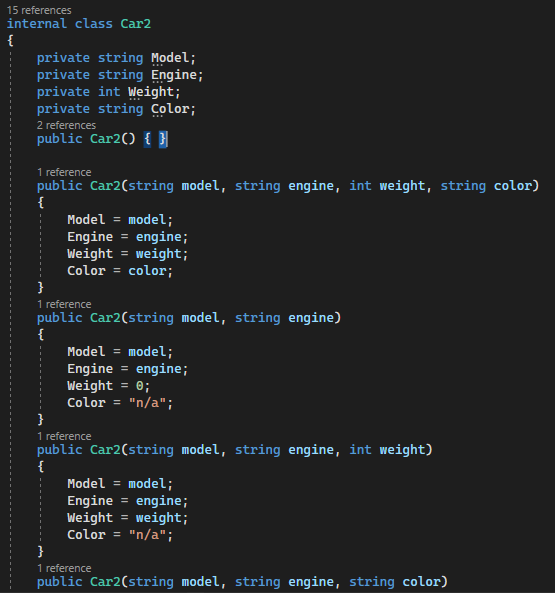
RESULT:



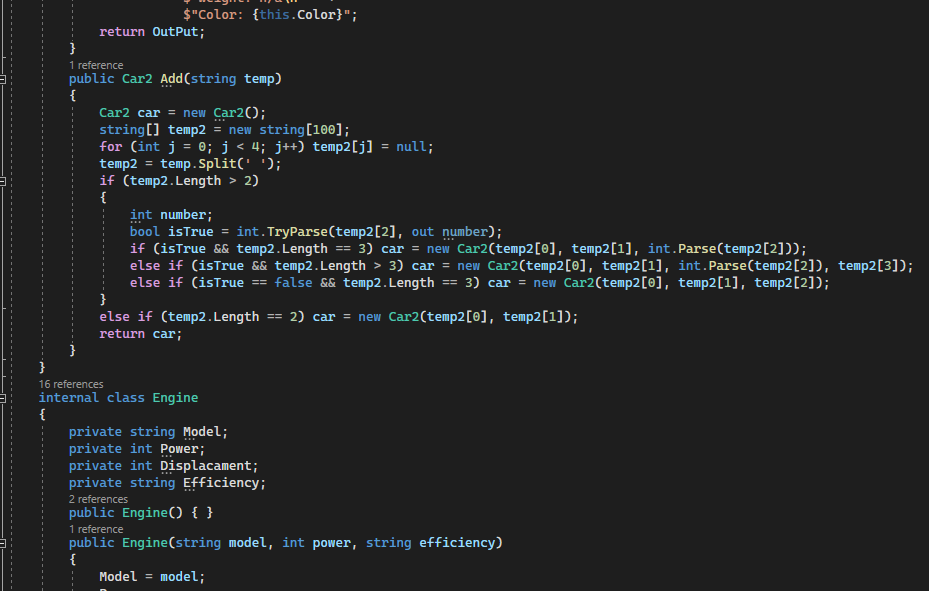
1. EXC2:

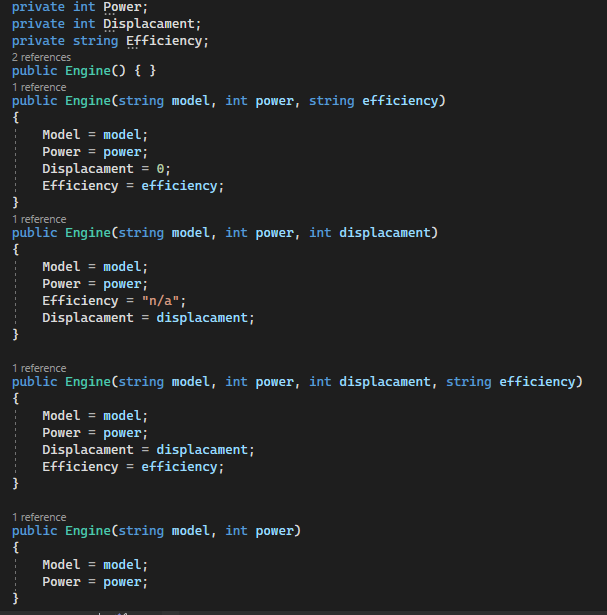
SOURCE:

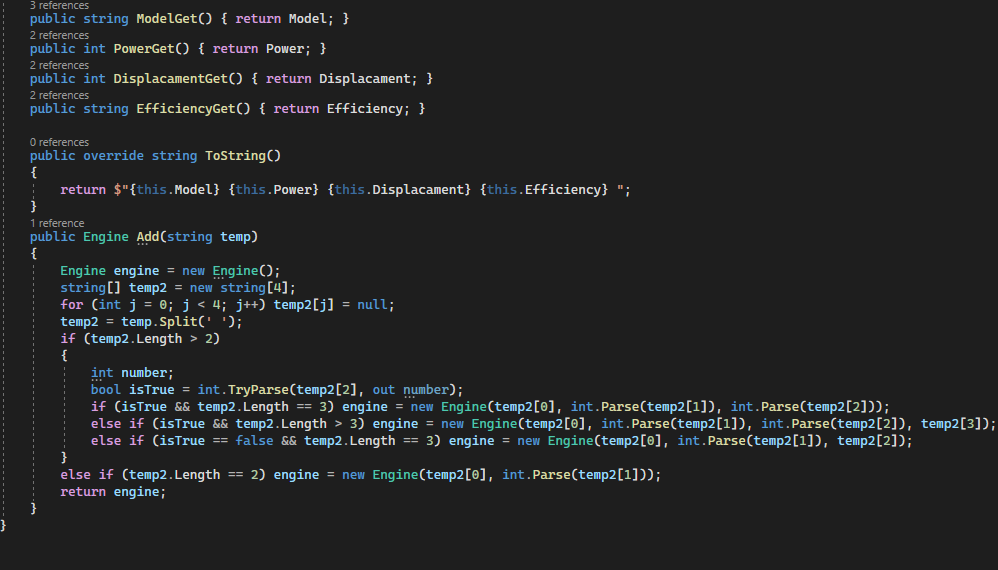




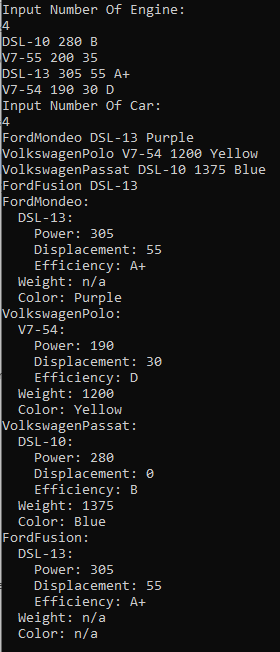






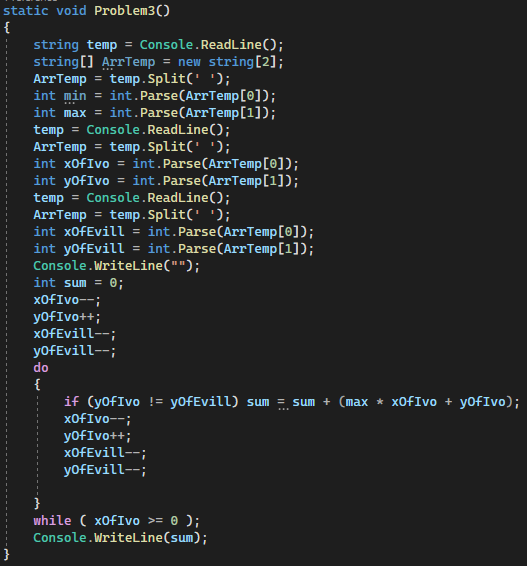


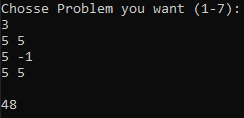
RESULT:



1. EXC3:

SOURCE:



RESULT: 

1. EXC4:

SOURCE:

RESULT:

1. EXC5:

SOURCE:

RESULT:

1. EXC6:

SOURCE:

RESULT:

1. EXC7:

SOURCE:

RESULT: