Task 4: Strings,2d Arrays, user defined functions,Hashmap

Parcel Tracking: Create a program that allows users to input a parcel tracking number.Store the tracking number and Status in 2d String Array. Initialize the array with values. Then, simulate the tracking process by displaying messages like "Parcel in transit," "Parcel out for delivery," or "Parcel delivered" based on the tracking number's status.

Customer Data Validation: Write a function which takes 2 parameters, data-denotes the data and detail-denotes if it is name addtress or phone number.Validate customer information based on following critirea. Ensure that names contain only letters and are properly capitalized, addresses do not contain special characters, and phone numbers follow a specific format (e.g., ###-###-####).

Address Formatting: Develop a function that takes an address as input (street, city, state, zip code) and formats it correctly, including capitalizing the first letter of each word and properly formatting the zip code.

Order Confirmation Email: Create a program that generates an order confirmation email. The email should include details such as the customer's name, order number, delivery address, and expected delivery date.

Calculate Shipping Costs: Develop a function that calculates the shipping cost based on the distance between two locations and the weight of the parcel. You can use string inputs for the source and destination addresses.

Password Generator: Create a function that generates secure passwords for courier system accounts. Ensure the passwords contain a mix of uppercase letters, lowercase letters, numbers, and special characters.

Find Similar Addresses: Implement a function that finds similar addresses in the system. This can be useful for identifying duplicate customer entries or optimizing delivery routes.Use string functions to implement this. Following tasks are incremental stages to build an application and should be done in a single project

Code:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;


namespace TaskFour

{

   internal class Program

   {
```

```
static void Main(string[] args)
{
   while (true)
   {
      Console.WriteLine("\nCourier Management System\n");
      Console.WriteLine("1. Track Parcel");
      Console.WriteLine("2. Validate Customer Data");
      Console.WriteLine("3. Format Address");
      Console.WriteLine("4. Generate Order Confirmation Email");
      Console.WriteLine("5. Calculate Shipping Cost");
      Console.WriteLine("6. Generate Secure Password");
      Console.WriteLine("7. Find Similar Addresses");
      Console.WriteLine("8. Exit");
      Console.WriteLine("Enter your choice: ");

      int choice = int.Parse(Console.ReadLine());
     switch (choice)
      {
         case 1:
            TrackParcel();
            break;
         case 2:
            ValidateCustomerData();
            break;
         case 3:
            FormatAddress();
            break;
         case 4:
            GenerateOrderConfirmation();
            break;
         case 5:
            CalculateShippingCost();
```

```
                break;
            case 6:
                GeneratePassword();
                break;
            case 7:
                FindSimilarAddresses();
                break;
            case 8:
                return;
            default:
                Console.WriteLine("Invalid choice");
                break;
        }
    }
}

static void TrackParcel()
{
    string[,] trackingData = {
{"12345", "In Transit"},
{"67890", "Out for Delivery"},
{"54321", "Delivered"},
{"98765", "Processing"}
};

    Console.Write("Enter tracking number: ");
    string trackingNumber = Console.ReadLine();

    for (int i = 0; i < trackingData.Length; i++)
    {
        if (trackingData[i, 0] == trackingNumber)
        {
```

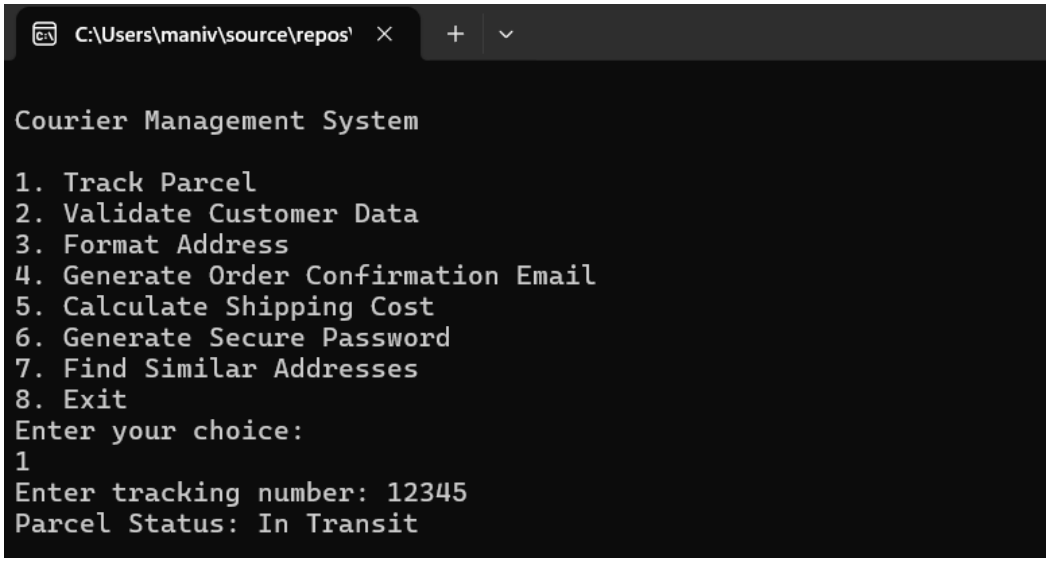
```
            Console.WriteLine($"Parcel Status: {trackingData[i, 1]}");
            return;
        }
    }
    Console.WriteLine("Tracking number not found!");
}
static void ValidateCustomerData()
{
    Console.Write("Enter data type (name, address, phone): ");
    string detail = Console.ReadLine().ToLower();

    Console.Write("Enter the data: ");
    string data = Console.ReadLine();

    bool isValid = false;

    switch (detail)
    {
        case "name":
            isValid = Regex.IsMatch(data, @"^[A-Z][a-zA-Z\s]+$");
            break;
        case "address":
            isValid = !Regex.IsMatch(data, @"[!@#$%^&*(),.?\:{ }|<>]");
            break;
        case "phone":
            isValid = Regex.IsMatch(data, @"^\d{10}$");
            break;
        default:
            Console.WriteLine("Invalid data type!");
            return;
    }
```

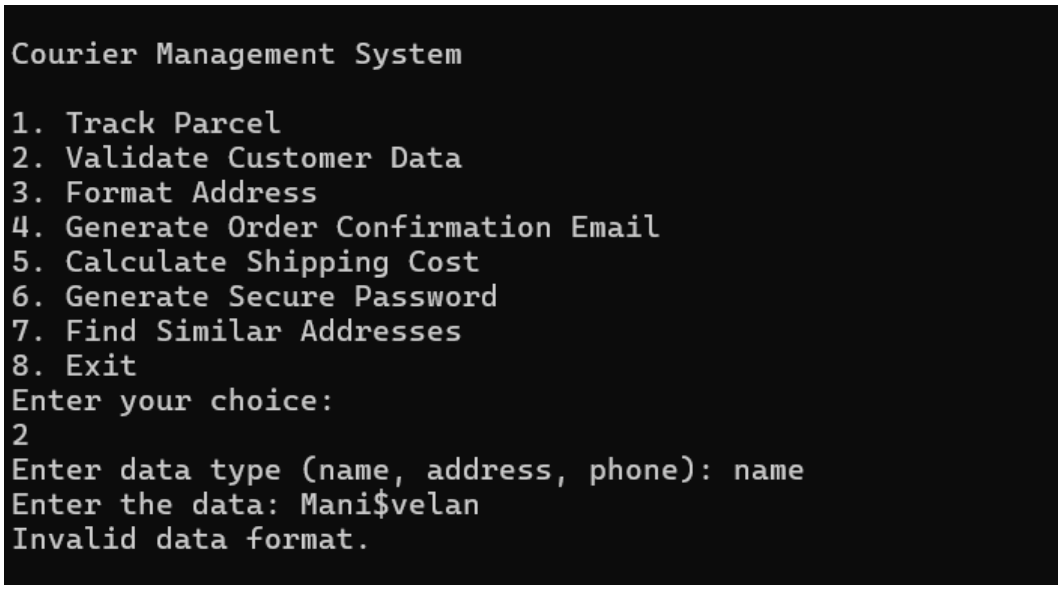
```
    Console.WriteLine(isValid ? "Valid data." : "Invalid data format.");
}
static void FormatAddress()
{
    Console.Write("Enter Street: ");
    string street = Console.ReadLine();
    Console.Write("Enter City: ");
    string city = Console.ReadLine();
    Console.Write("Enter State: ");
    string state = Console.ReadLine();
    Console.Write("Enter Zip Code: ");
    string zip = Console.ReadLine();

    string formattedAddress =
        $"{char.ToUpper(street[0])}{street.Substring(1)}, " +
         $"{char.ToUpper(city[0])}{city.Substring(1)}, " +
         $"{char.ToUpper(state[0])}{state.Substring(1)} - {zip}";
    Console.WriteLine("Formatted Address: " + formattedAddress);
}

static void GenerateOrderConfirmation()
{
    Console.Write("Enter Customer Name: ");
    string name = Console.ReadLine();
    Console.Write("Enter Order Number: ");
    string orderNumber = Console.ReadLine();
    Console.Write("Enter Delivery Address: ");
    string address = Console.ReadLine();
    Console.Write("Enter Expected Delivery Date: ");
    string deliveryDate = Console.ReadLine();

    Console.WriteLine("\nOrder Confirmation Email:");
```

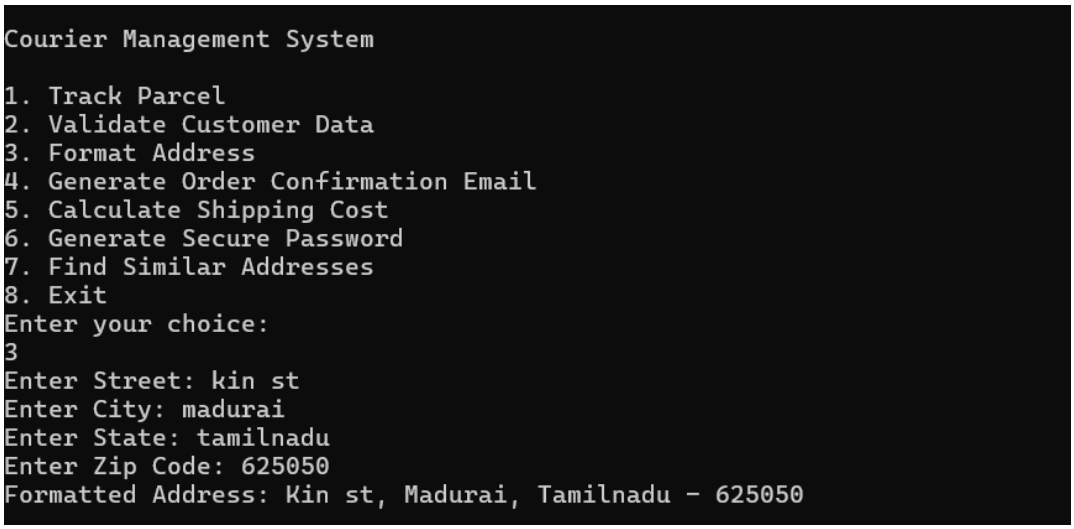
```
    Console.WriteLine($"Dear {name},\nYour order {orderNumber} is confirmed.");
    Console.WriteLine($"Delivery Address: {address}");
    Console.WriteLine($"Expected Delivery Date: {deliveryDate}");
    Console.WriteLine("Thank you for choosing our service!");
}

static void CalculateShippingCost()
{
    Console.Write("Enter Source Location: ");
    string source =Console.ReadLine();
    Console.Write("Enter Destination Location: ");
    string destination =Console.ReadLine();
    Console.Write("Enter Parcel Weight (kg): ");
    double weight= int.Parse( Console.ReadLine() );
    Random rnd = new Random();
    double distance = rnd.Next(10, 1000);
    double cost = (distance * 0.5) + (weight * 2);

    Console.WriteLine($"Shipping cost from {source} to {destination}: {cost}");
}

static void GeneratePassword()
{
    return;
}
static void FindSimilarAddresses()
{

    string[,] customerAddresses = {
{"Manivelan", "123,Mullai St,Madurai-610001"},
{"reshmika", "456,king colonytrichy-690001"},
{"abdul", "3A,raj apartment,Madurai-660601"},
```

```
        {"robert", "101,D street,Chenani-677001"}

        };
            Console.Write("Enter Address to Search: ");
            string searchAddress = Console.ReadLine().ToLower();

            Console.WriteLine("Matching Addresses:");
            bool found = false;

            for (int i = 0; i < customerAddresses.GetLength(0); i++)
            {
                if (customerAddresses[i, 1].ToLower().Contains(searchAddress))
                {
                    Console.WriteLine($"{customerAddresses[i, 0]}: {customerAddresses[i, 1]}");
                    found = true;
                }
            }

            if (!found)
            {
                Console.WriteLine("No matching addresses found.");
            }
        }


    }
}
```

Output:


C:\Users\maniv\source\repos
Courier Management System
1. Track Parcel
2. Validate Customer Data
3. Format Address
4. Generate Order Confirmation Email
5. Calculate Shipping Cost
6. Generate Secure Password
7. Find Similar Addresses
8. Exit
Enter your choice:
1
Enter tracking number: 12345
Parcel Status: In Transit


Courier Management System
1. Track Parcel
2. Validate Customer Data
3. Format Address
4. Generate Order Confirmation Email
5. Calculate Shipping Cost
6. Generate Secure Password
7. Find Similar Addresses
8. Exit
Enter your choice:
2
Enter data type (name, address, phone): name
Enter the data: Mani$velan
Invalid data format.


Courier Management System
1. Track Parcel
2. Validate Customer Data
3. Format Address
4. Generate Order Confirmation Email
5. Calculate Shipping Cost
6. Generate Secure Password
7. Find Similar Addresses
8. Exit
Enter your choice:
3
Enter Street: kin st
Enter City: madurai
Enter State: tamilnadu
Enter Zip Code: 625050
Formatted Address: Kin st, Madurai, Tamilnadu - 625050

```
Courier Management System

1. Track Parcel
2. Validate Customer Data
3. Format Address
4. Generate Order Confirmation Email
5. Calculate Shipping Cost
6. Generate Secure Password
7. Find Similar Addresses
8. Exit
Enter your choice:
4
Enter Customer Name: Manivelan
Enter Order Number: 1234
Enter Delivery Address: 4/271 sourastrapuram , Madurai
Enter Expected Delivery Date: 04/04/2025

Order Confirmation Email:
Dear Manivelan,
Your order 1234 is confirmed.
Delivery Address: 4/271 sourastrapuram , Madurai
Expected Delivery Date: 04/04/2025
Thank you for choosing our service!
```

```
Courier Management System

1. Track Parcel
2. Validate Customer Data
3. Format Address
4. Generate Order Confirmation Email
5. Calculate Shipping Cost
6. Generate Secure Password
7. Find Similar Addresses
8. Exit
Enter your choice:
5
Enter Source Location: Madurai
Enter Destination Location: Chennai
Enter Parcel Weight (kg): 5
Shipping cost from Madurai to Chennai: 129.5
```

```
Courier Management System

1. Track Parcel
2. Validate Customer Data
3. Format Address
4. Generate Order Confirmation Email
5. Calculate Shipping Cost
6. Generate Secure Password
7. Find Similar Addresses
8. Exit
Enter your choice:
7
Enter Address to Search: 123,Mullai st,Madurai-610001
Matching Addresses:
Manivelan: 123,Mullai St,Madurai-610001
```