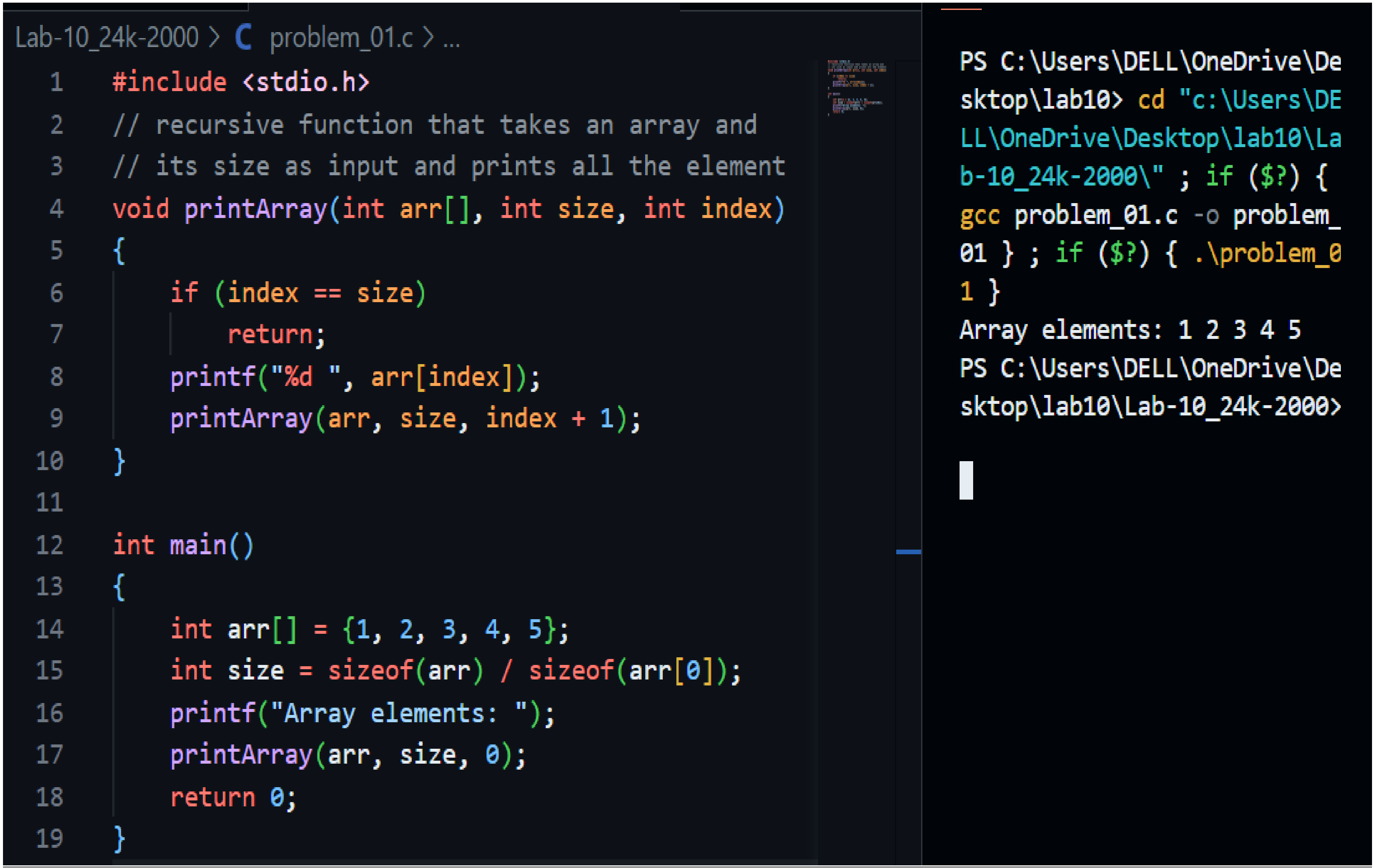
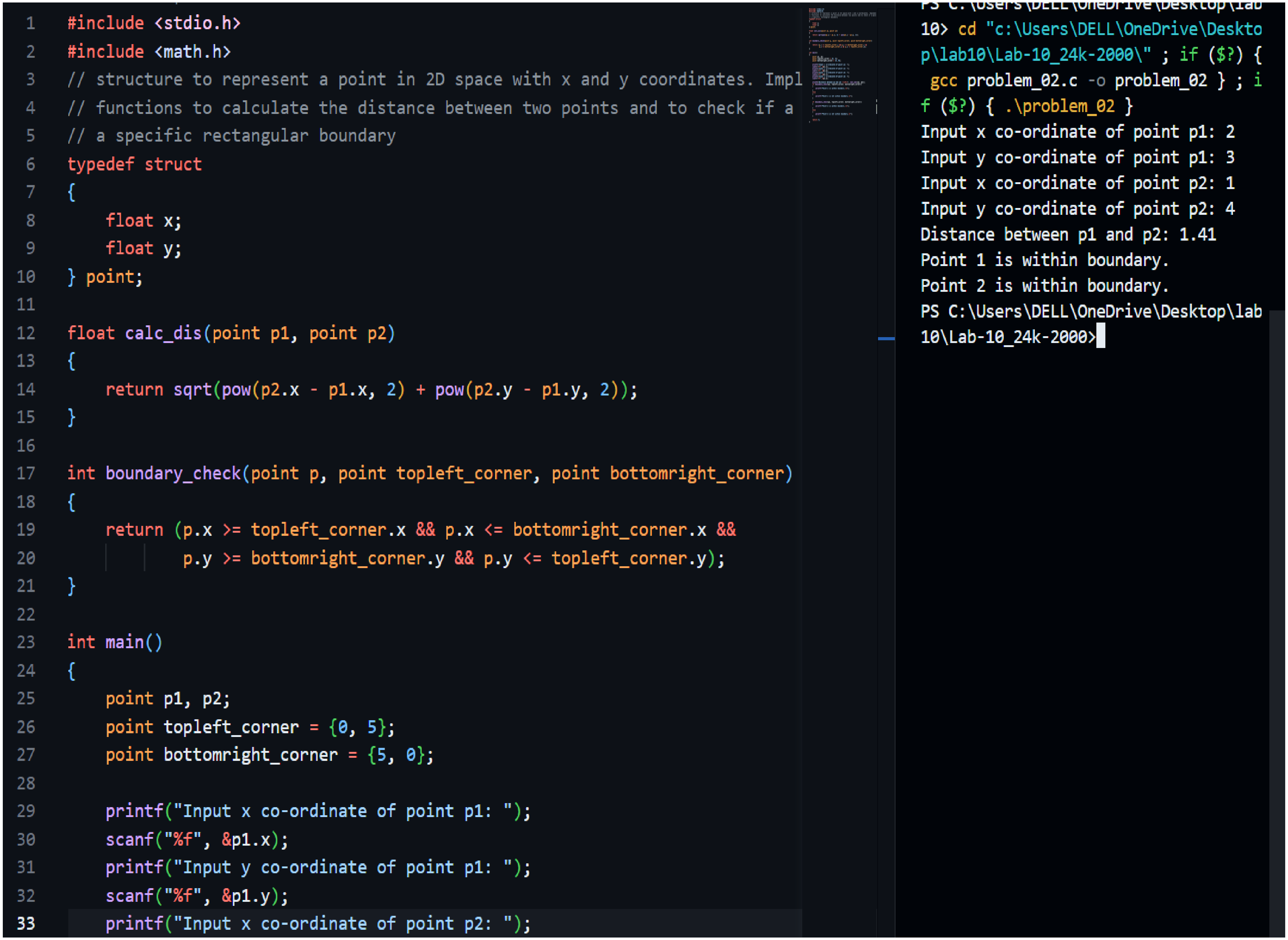
Problem 01:



```c
#include <stdio.h>
// recursive function that takes an array and
// its size as input and prints all the element
void printArray(int arr[], int size, int index)
{
    if (index == size)
        return;
    printf("%d ", arr[index]);
    printArray(arr, size, index + 1);
}

int main()
{
    int arr[] = {1, 2, 3, 4, 5};
    int size = sizeof(arr) / sizeof(arr[0]);
    printf("Array elements: ");
    printArray(arr, size, 0);
    return 0;
}
```

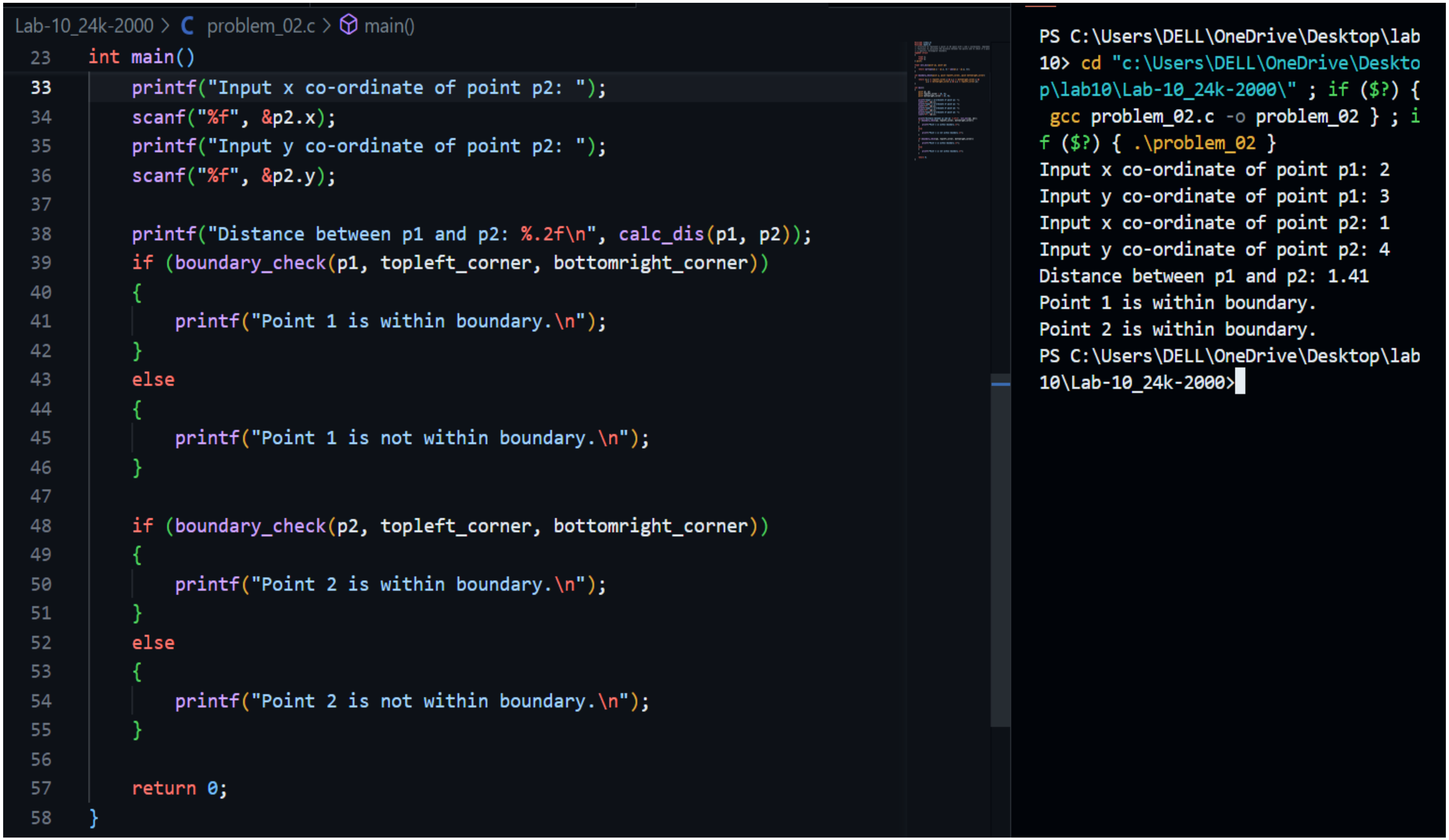
```
PS C:\Users\DELL\OneDrive\Desktop\lab10> cd "c:\Users\DELL\OneDrive\Desktop\lab10\Lab-10_24k-2000\" ; if ($?) { gcc problem_01.c -o problem_01 } ; if ($?) { .\problem_01 }
Array elements: 1 2 3 4 5
PS C:\Users\DELL\OneDrive\Desktop\lab10\Lab-10_24k-2000>
```

Problem 02:



```c
#include <stdio.h>
#include <math.h>
// structure to represent a point in 2D space with x and y coordinates. Impl
// functions to calculate the distance between two points and to check if a
// a specific rectangular boundary
typedef struct
{
    float x;
    float y;
} point;

float calc_dis(point p1, point p2)
{
    return sqrt(pow(p2.x - p1.x, 2) + pow(p2.y - p1.y, 2));
}

int boundary_check(point p, point topleft_corner, point bottomright_corner)
{
    return (p.x >= topleft_corner.x && p.x <= bottomright_corner.x &&
            p.y >= bottomright_corner.y && p.y <= topleft_corner.y);
}

int main()
{
    point p1, p2;
    point topleft_corner = {0, 5};
    point bottomright_corner = {5, 0};

    printf("Input x co-ordinate of point p1: ");
    scanf("%f", &p1.x);
    printf("Input y co-ordinate of point p1: ");
    scanf("%f", &p1.y);
    printf("Input x co-ordinate of point p2: ");
```

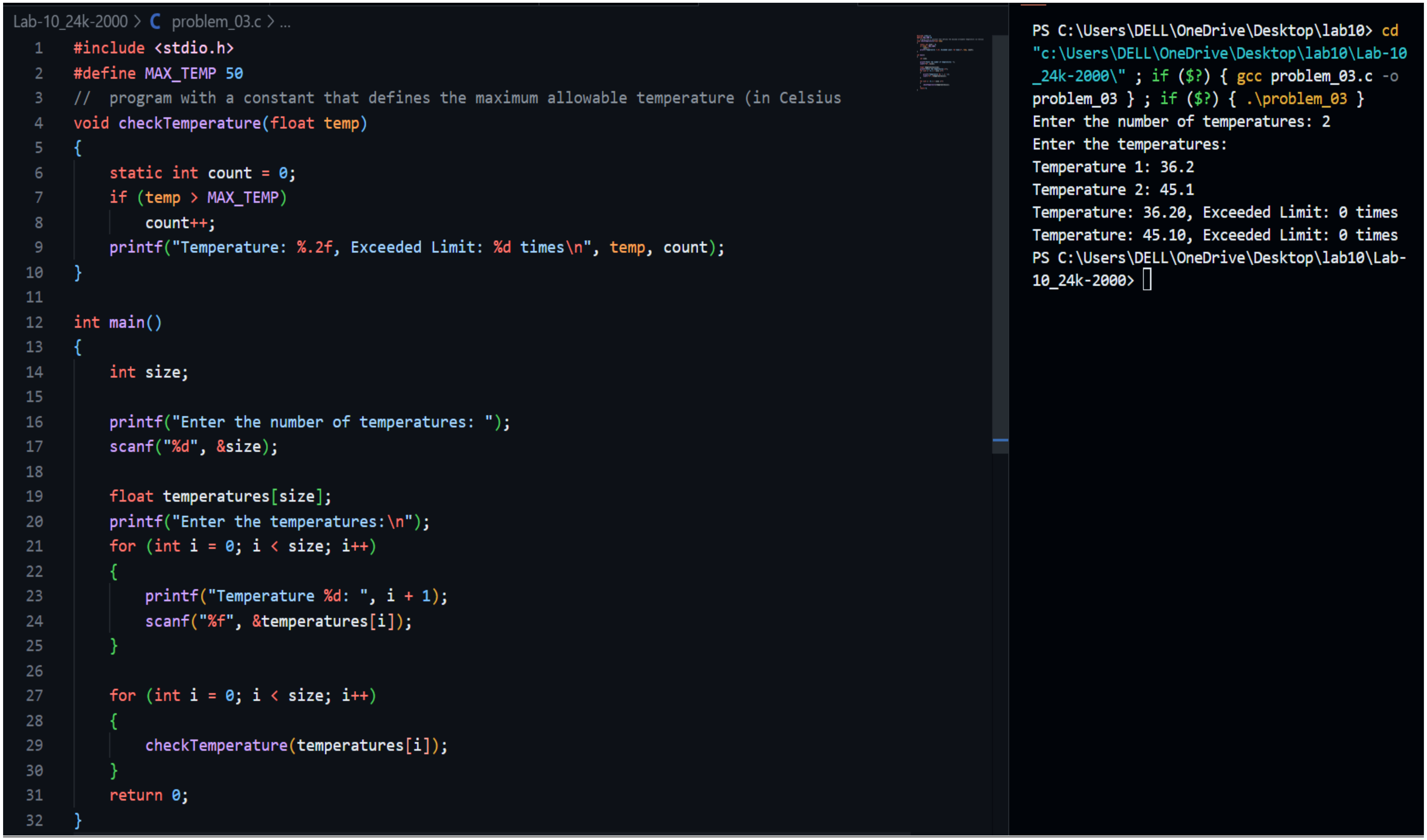
```
PS C:\Users\DELL\OneDrive\Desktop\lab
10> cd "c:\Users\DELL\OneDrive\Deskto
p\lab10\Lab-10_24k-2000\" ; if ($?) {
 gcc problem_02.c -o problem_02 } ; i
f ($?) { .\problem_02 }
Input x co-ordinate of point p1: 2
Input y co-ordinate of point p1: 3
Input x co-ordinate of point p2: 1
Input y co-ordinate of point p2: 4
Distance between p1 and p2: 1.41
Point 1 is within boundary.
Point 2 is within boundary.
PS C:\Users\DELL\OneDrive\Desktop\lab
10\Lab-10_24k-2000>
```



```c
int main()
    printf("Input x co-ordinate of point p2: ");
    scanf("%f", &p2.x);
    printf("Input y co-ordinate of point p2: ");
    scanf("%f", &p2.y);

    printf("Distance between p1 and p2: %.2f\n", calc_dis(p1, p2));
    if (boundary_check(p1, topleft_corner, bottomright_corner))
    {
        printf("Point 1 is within boundary.\n");
    }
    else
    {
        printf("Point 1 is not within boundary.\n");
    }

    if (boundary_check(p2, topleft_corner, bottomright_corner))
    {
        printf("Point 2 is within boundary.\n");
    }
    else
    {
        printf("Point 2 is not within boundary.\n");
    }

    return 0;
}
```

```
PS C:\Users\DELL\OneDrive\Desktop\lab
10> cd "c:\Users\DELL\OneDrive\Deskto
p\lab10\Lab-10_24k-2000\" ; if ($?) {
 gcc problem_02.c -o problem_02 } ; i
f ($?) { .\problem_02 }
Input x co-ordinate of point p1: 2
Input y co-ordinate of point p1: 3
Input x co-ordinate of point p2: 1
Input y co-ordinate of point p2: 4
Distance between p1 and p2: 1.41
Point 1 is within boundary.
Point 2 is within boundary.
PS C:\Users\DELL\OneDrive\Desktop\lab
10\Lab-10_24k-2000>
```

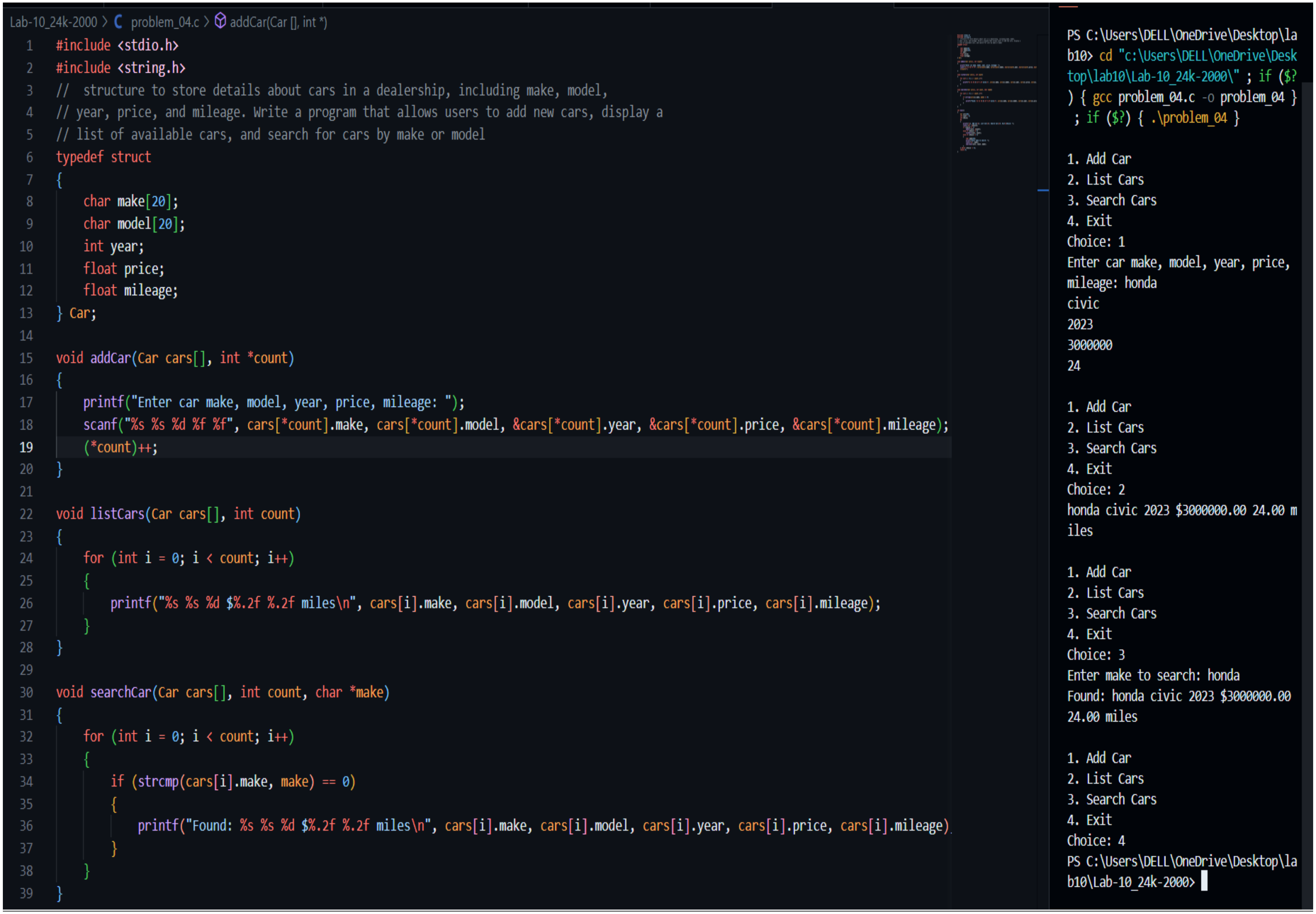
Problem 03:



```c
#include <stdio.h>
#define MAX_TEMP 50
//  program with a constant that defines the maximum allowable temperature (in Celsius
void checkTemperature(float temp)
{
    static int count = 0;
    if (temp > MAX_TEMP)
        count++;
    printf("Temperature: %.2f, Exceeded Limit: %d times\n", temp, count);
}

int main()
{
    int size;

    printf("Enter the number of temperatures: ");
    scanf("%d", &size);

    float temperatures[size];
    printf("Enter the temperatures:\n");
    for (int i = 0; i < size; i++)
    {
        printf("Temperature %d: ", i + 1);
        scanf("%f", &temperatures[i]);
    }

    for (int i = 0; i < size; i++)
    {
        checkTemperature(temperatures[i]);
    }
    return 0;
}
```

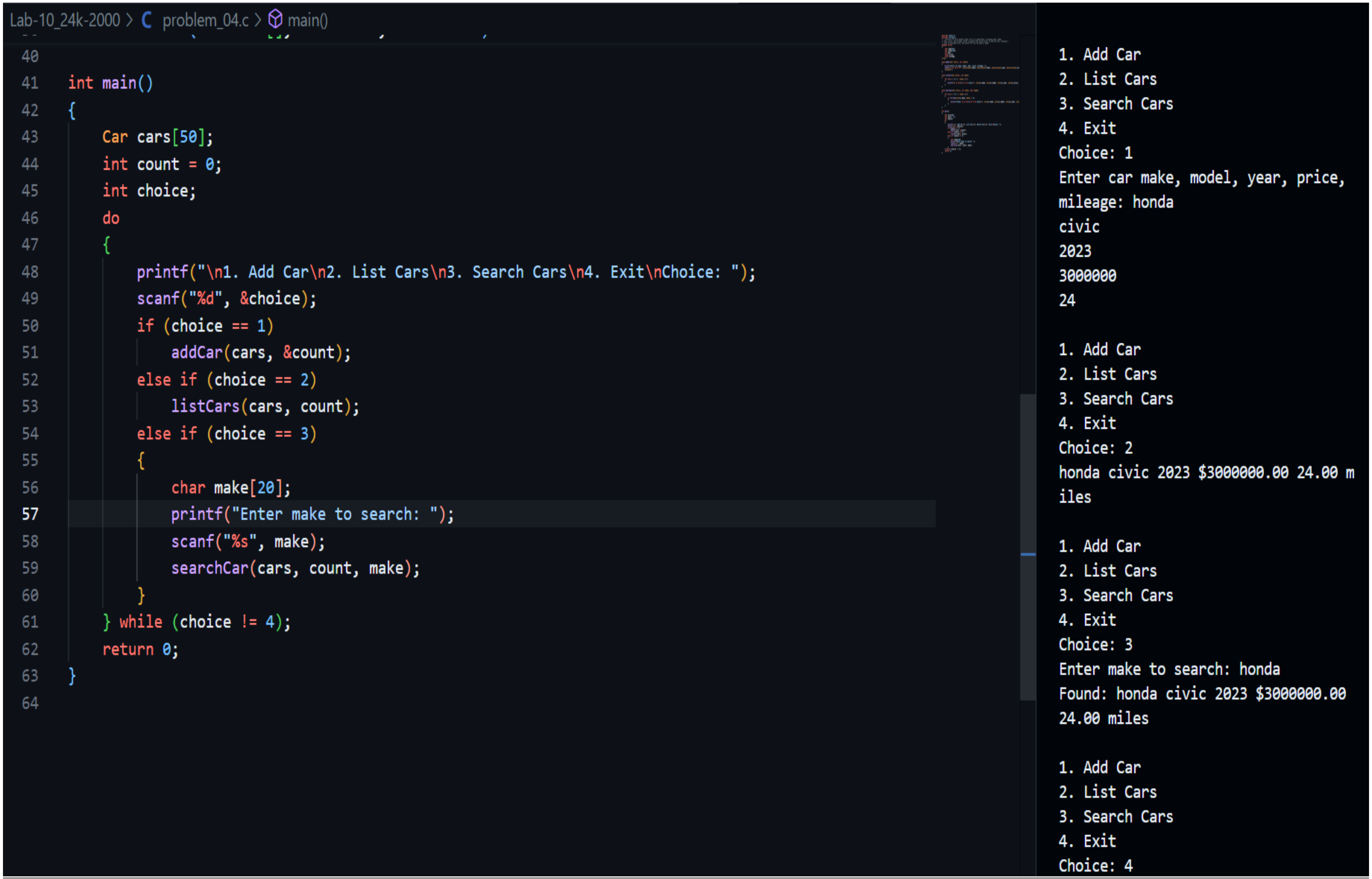
```
PS C:\Users\DELL\OneDrive\Desktop\lab10> cd
"c:\Users\DELL\OneDrive\Desktop\lab10\Lab-10
_24k-2000\" ; if ($?) { gcc problem_03.c -o
problem_03 } ; if ($?) { .\problem_03 }
Enter the number of temperatures: 2
Enter the temperatures:
Temperature 1: 36.2
Temperature 2: 45.1
Temperature: 36.20, Exceeded Limit: 0 times
Temperature: 45.10, Exceeded Limit: 0 times
PS C:\Users\DELL\OneDrive\Desktop\lab10\Lab-
10_24k-2000>
```

Problem 04:



```c
#include <stdio.h>
#include <string.h>
//  structure to store details about cars in a dealership, including make, model,
// year, price, and mileage. Write a program that allows users to add new cars, display a
// list of available cars, and search for cars by make or model
typedef struct
{
    char make[20];
    char model[20];
    int year;
    float price;
    float mileage;
} Car;

void addCar(Car cars[], int *count)
{
    printf("Enter car make, model, year, price, mileage: ");
    scanf("%s %s %d %f %f", cars[*count].make, cars[*count].model, &cars[*count].year, &cars[*count].price, &cars[*count].mileage);
    (*count)++;
}

void listCars(Car cars[], int count)
{
    for (int i = 0; i < count; i++)
    {
        printf("%s %s %d $%.2f %.2f miles\n", cars[i].make, cars[i].model, cars[i].year, cars[i].price, cars[i].mileage);
    }
}

void searchCar(Car cars[], int count, char *make)
{
    for (int i = 0; i < count; i++)
    {
        if (strcmp(cars[i].make, make) == 0)
        {
            printf("Found: %s %s %d $%.2f %.2f miles\n", cars[i].make, cars[i].model, cars[i].year, cars[i].price, cars[i].mileage)
        }
    }
}
```

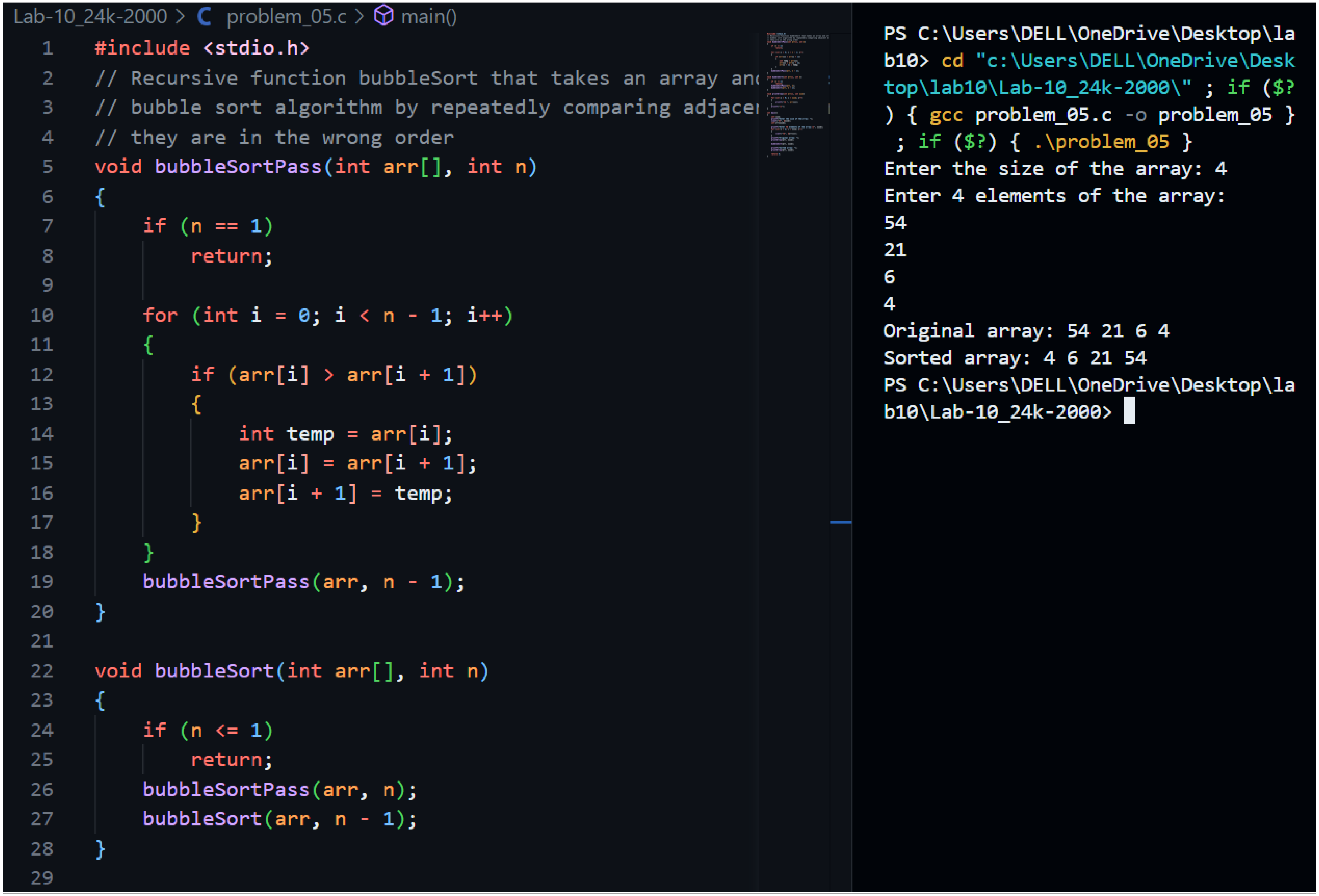
```
PS C:\Users\DELL\OneDrive\Desktop\la
b10> cd "c:\Users\DELL\OneDrive\Desk
top\lab10\Lab-10_24k-2000\" ; if ($?
) { gcc problem_04.c -o problem_04 }
 ; if ($?) { .\problem_04 }

1. Add Car
2. List Cars
3. Search Cars
4. Exit
Choice: 1
Enter car make, model, year, price,
mileage: honda
civic
2023
3000000
24

1. Add Car
2. List Cars
3. Search Cars
4. Exit
Choice: 2
honda civic 2023 $3000000.00 24.00 m
iles

1. Add Car
2. List Cars
3. Search Cars
4. Exit
Choice: 3
Enter make to search: honda
Found: honda civic 2023 $3000000.00
24.00 miles

1. Add Car
2. List Cars
3. Search Cars
4. Exit
Choice: 4
PS C:\Users\DELL\OneDrive\Desktop\la
b10\Lab-10_24k-2000>
```



```c

int main()
{
    Car cars[50];
    int count = 0;
    int choice;
    do
    {
        printf("\n1. Add Car\n2. List Cars\n3. Search Cars\n4. Exit\nChoice: ");
        scanf("%d", &choice);
        if (choice == 1)
            addCar(cars, &count);
        else if (choice == 2)
            listCars(cars, count);
        else if (choice == 3)
        {
            char make[20];
            printf("Enter make to search: ");
            scanf("%s", make);
            searchCar(cars, count, make);
        }
    } while (choice != 4);
    return 0;
}

```

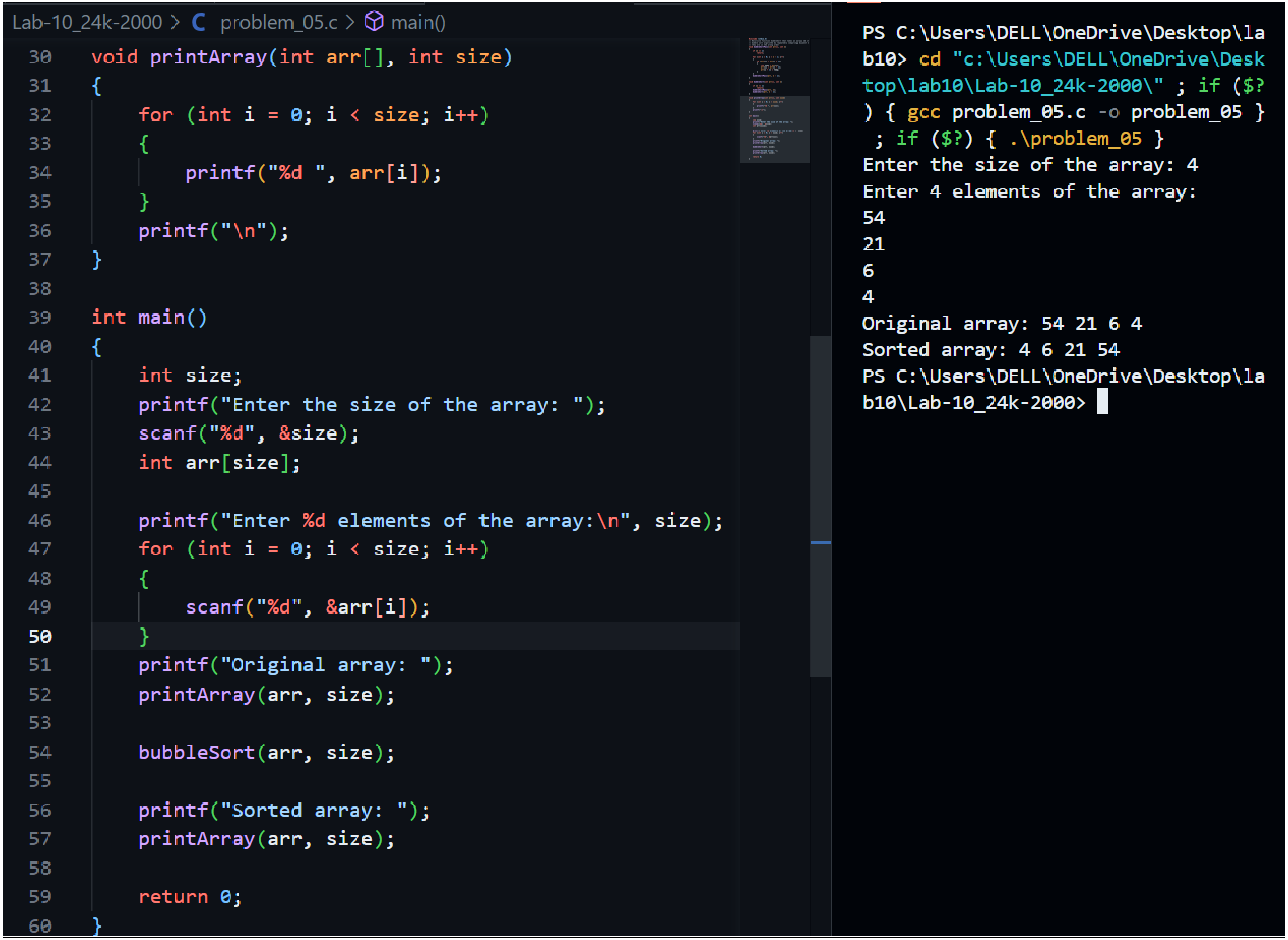
```
1. Add Car
2. List Cars
3. Search Cars
4. Exit
Choice: 1
Enter car make, model, year, price,
mileage: honda
civic
2023
3000000
24

1. Add Car
2. List Cars
3. Search Cars
4. Exit
Choice: 2
honda civic 2023 $3000000.00 24.00 m
iles

1. Add Car
2. List Cars
3. Search Cars
4. Exit
Choice: 3
Enter make to search: honda
Found: honda civic 2023 $3000000.00
24.00 miles

1. Add Car
2. List Cars
3. Search Cars
4. Exit
Choice: 4
```

Problem 05:



```c
#include <stdio.h>
// Recursive function bubbleSort that takes an array and
// bubble sort algorithm by repeatedly comparing adjacen
// they are in the wrong order
void bubbleSortPass(int arr[], int n)
{
    if (n == 1)
        return;

    for (int i = 0; i < n - 1; i++)
    {
        if (arr[i] > arr[i + 1])
        {
            int temp = arr[i];
            arr[i] = arr[i + 1];
            arr[i + 1] = temp;
        }
    }
    bubbleSortPass(arr, n - 1);
}

void bubbleSort(int arr[], int n)
{
    if (n <= 1)
        return;
    bubbleSortPass(arr, n);
    bubbleSort(arr, n - 1);
}

```

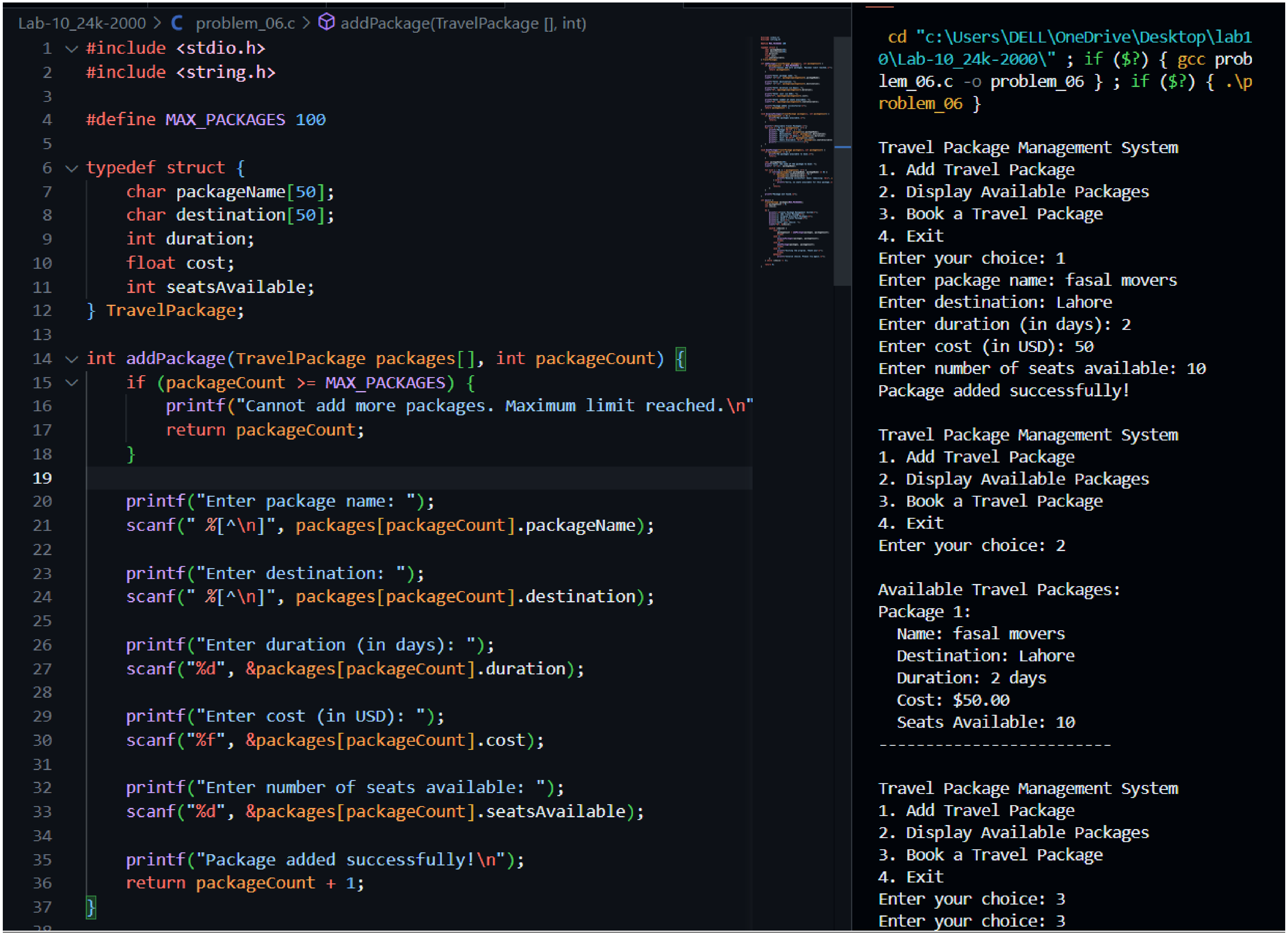
```
PS C:\Users\DELL\OneDrive\Desktop\lab10> cd "c:\Users\DELL\OneDrive\Desktop\lab10\Lab-10_24k-2000\" ; if ($?) { gcc problem_05.c -o problem_05 } ; if ($?) { .\problem_05 }
Enter the size of the array: 4
Enter 4 elements of the array:
54
21
6
4
Original array: 54 21 6 4
Sorted array: 4 6 21 54
PS C:\Users\DELL\OneDrive\Desktop\lab10\Lab-10_24k-2000>
```



```c
void printArray(int arr[], int size)
{
    for (int i = 0; i < size; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main()
{
    int size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int arr[size];

    printf("Enter %d elements of the array:\n", size);
    for (int i = 0; i < size; i++)
    {
        scanf("%d", &arr[i]);
    }
    printf("Original array: ");
    printArray(arr, size);

    bubbleSort(arr, size);

    printf("Sorted array: ");
    printArray(arr, size);

    return 0;
}
```

```
PS C:\Users\DELL\OneDrive\Desktop\la
b10> cd "c:\Users\DELL\OneDrive\Desk
top\lab10\Lab-10_24k-2000\" ; if ($?
) { gcc problem_05.c -o problem_05 }
 ; if ($?) { .\problem_05 }
Enter the size of the array: 4
Enter 4 elements of the array:
54
21
6
4
Original array: 54 21 6 4
Sorted array: 4 6 21 54
PS C:\Users\DELL\OneDrive\Desktop\la
b10\Lab-10_24k-2000>
```

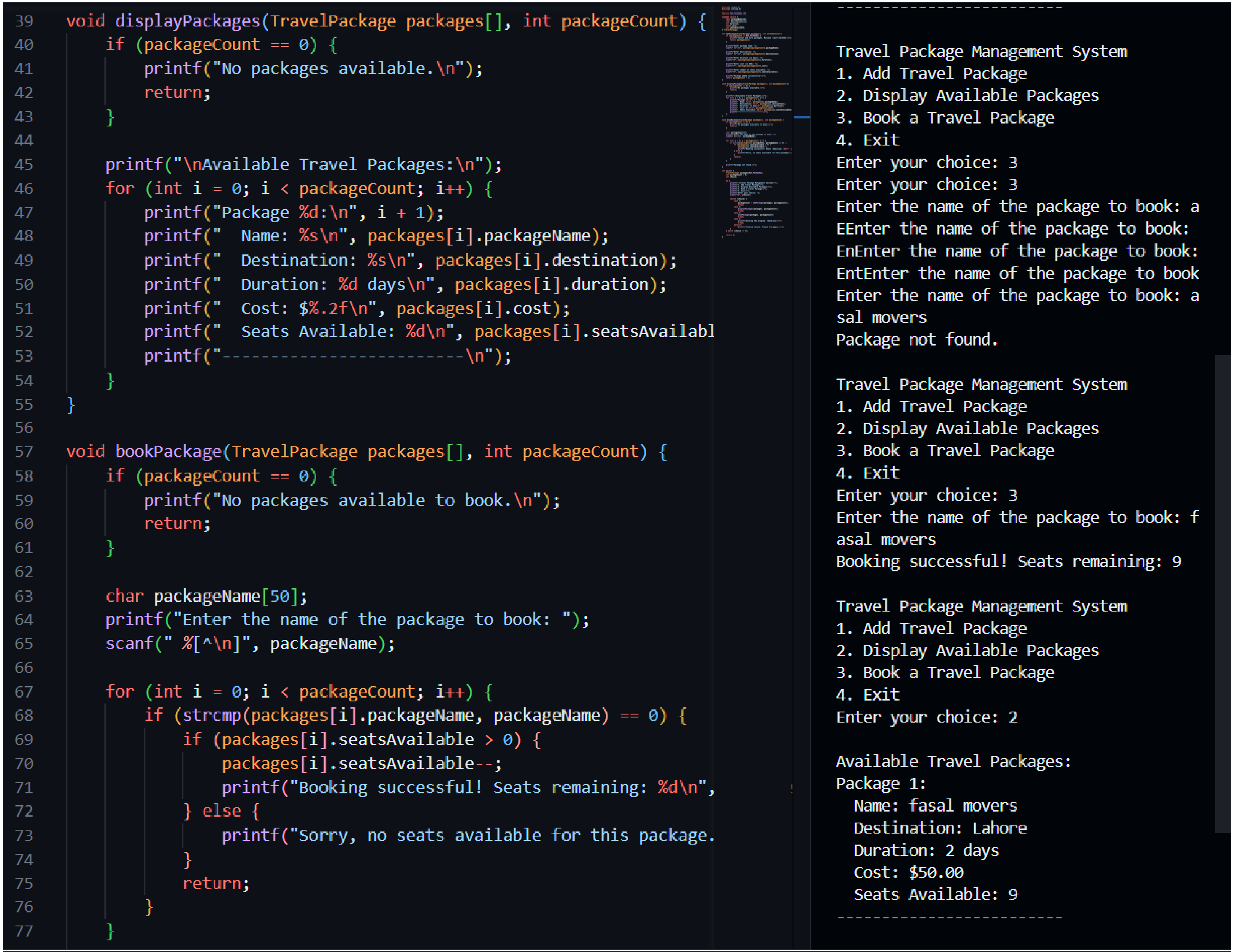
Problem 06:



Lab-10_24k-2000 > C problem_06.c > addPackage(TravelPackage [], int)

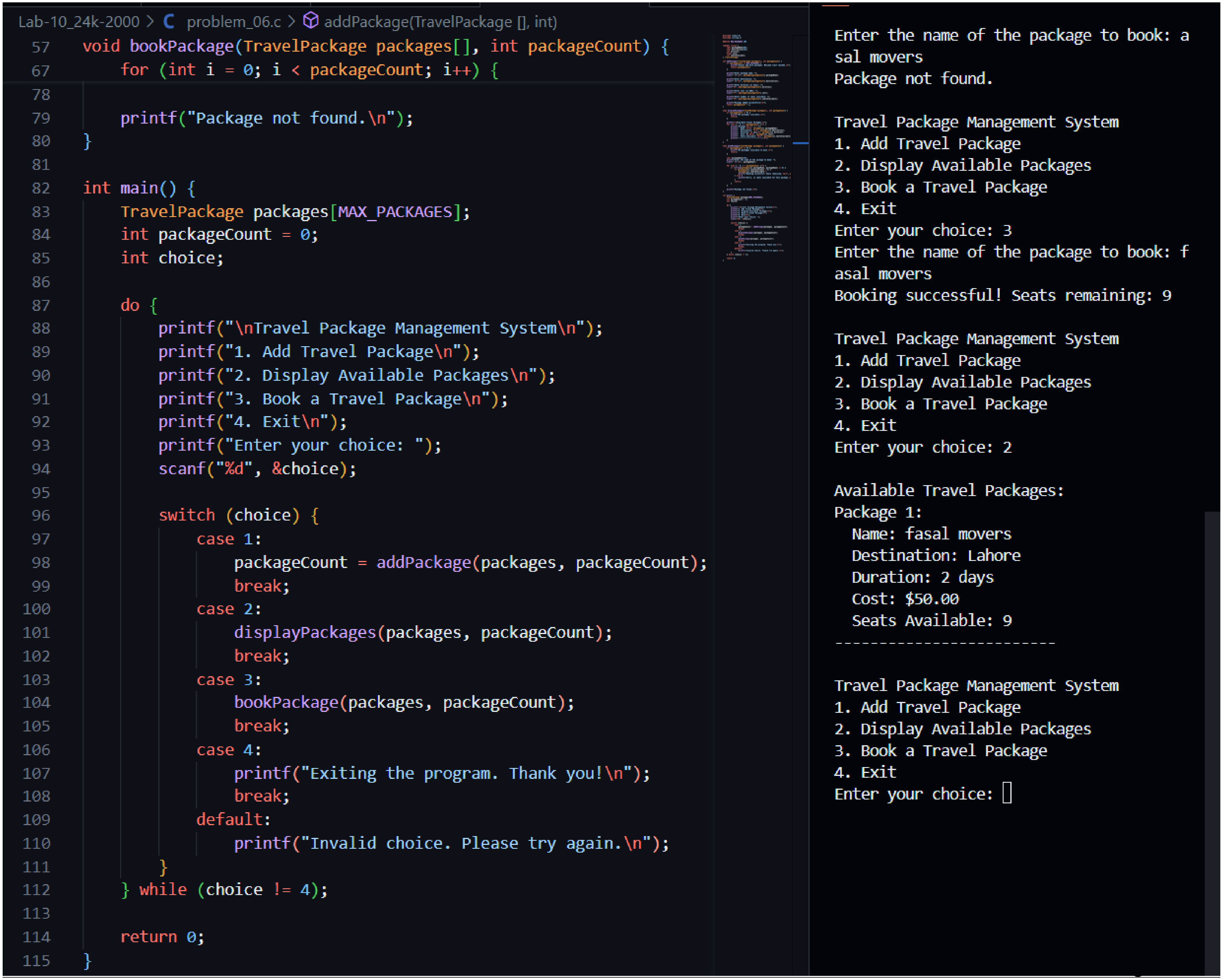
```c
#include <stdio.h>
#include <string.h>

#define MAX_PACKAGES 100

typedef struct {
    char packageName[50];
    char destination[50];
    int duration;
    float cost;
    int seatsAvailable;
} TravelPackage;

int addPackage(TravelPackage packages[], int packageCount) {
    if (packageCount >= MAX_PACKAGES) {
        printf("Cannot add more packages. Maximum limit reached.\n"
        return packageCount;
    }

    printf("Enter package name: ");
    scanf(" %[^\n]", packages[packageCount].packageName);

    printf("Enter destination: ");
    scanf(" %[^\n]", packages[packageCount].destination);

    printf("Enter duration (in days): ");
    scanf("%d", &packages[packageCount].duration);

    printf("Enter cost (in USD): ");
    scanf("%f", &packages[packageCount].cost);

    printf("Enter number of seats available: ");
    scanf("%d", &packages[packageCount].seatsAvailable);

    printf("Package added successfully!\n");
    return packageCount + 1;
}
```

```
 cd "c:\Users\DELL\OneDrive\Desktop\lab1
0\Lab-10_24k-2000\" ; if ($?) { gcc prob
lem_06.c -o problem_06 } ; if ($?) { .\p
roblem_06 }

Travel Package Management System
1. Add Travel Package
2. Display Available Packages
3. Book a Travel Package
4. Exit
Enter your choice: 1
Enter package name: fasal movers
Enter destination: Lahore
Enter duration (in days): 2
Enter cost (in USD): 50
Enter number of seats available: 10
Package added successfully!

Travel Package Management System
1. Add Travel Package
2. Display Available Packages
3. Book a Travel Package
4. Exit
Enter your choice: 2

Available Travel Packages:
Package 1:
  Name: fasal movers
  Destination: Lahore
  Duration: 2 days
  Cost: $50.00
  Seats Available: 10
-------------------------

Travel Package Management System
1. Add Travel Package
2. Display Available Packages
3. Book a Travel Package
4. Exit
Enter your choice: 3
Enter your choice: 3
```



```c
void displayPackages(TravelPackage packages[], int packageCount) {
    if (packageCount == 0) {
        printf("No packages available.\n");
        return;
    }

    printf("\nAvailable Travel Packages:\n");
    for (int i = 0; i < packageCount; i++) {
        printf("Package %d:\n", i + 1);
        printf("  Name: %s\n", packages[i].packageName);
        printf("  Destination: %s\n", packages[i].destination);
        printf("  Duration: %d days\n", packages[i].duration);
        printf("  Cost: $%.2f\n", packages[i].cost);
        printf("  Seats Available: %d\n", packages[i].seatsAvailabl
        printf("------------------------\n");
    }
}

void bookPackage(TravelPackage packages[], int packageCount) {
    if (packageCount == 0) {
        printf("No packages available to book.\n");
        return;
    }

    char packageName[50];
    printf("Enter the name of the package to book: ");
    scanf(" %[^\n]", packageName);

    for (int i = 0; i < packageCount; i++) {
        if (strcmp(packages[i].packageName, packageName) == 0) {
            if (packages[i].seatsAvailable > 0) {
                packages[i].seatsAvailable--;
                printf("Booking successful! Seats remaining: %d\n",
            } else {
                printf("Sorry, no seats available for this package.
            }
            return;
        }
    }
```

```
------------------------

Travel Package Management System
1. Add Travel Package
2. Display Available Packages
3. Book a Travel Package
4. Exit
Enter your choice: 3
Enter your choice: 3
Enter the name of the package to book: a
EEnter the name of the package to book:
EnEnter the name of the package to book:
EntEnter the name of the package to book
Enter the name of the package to book: a
sal movers
Package not found.

Travel Package Management System
1. Add Travel Package
2. Display Available Packages
3. Book a Travel Package
4. Exit
Enter your choice: 3
Enter the name of the package to book: f
asal movers
Booking successful! Seats remaining: 9

Travel Package Management System
1. Add Travel Package
2. Display Available Packages
3. Book a Travel Package
4. Exit
Enter your choice: 2

Available Travel Packages:
Package 1:
  Name: fasal movers
  Destination: Lahore
  Duration: 2 days
  Cost: $50.00
  Seats Available: 9
------------------------
```

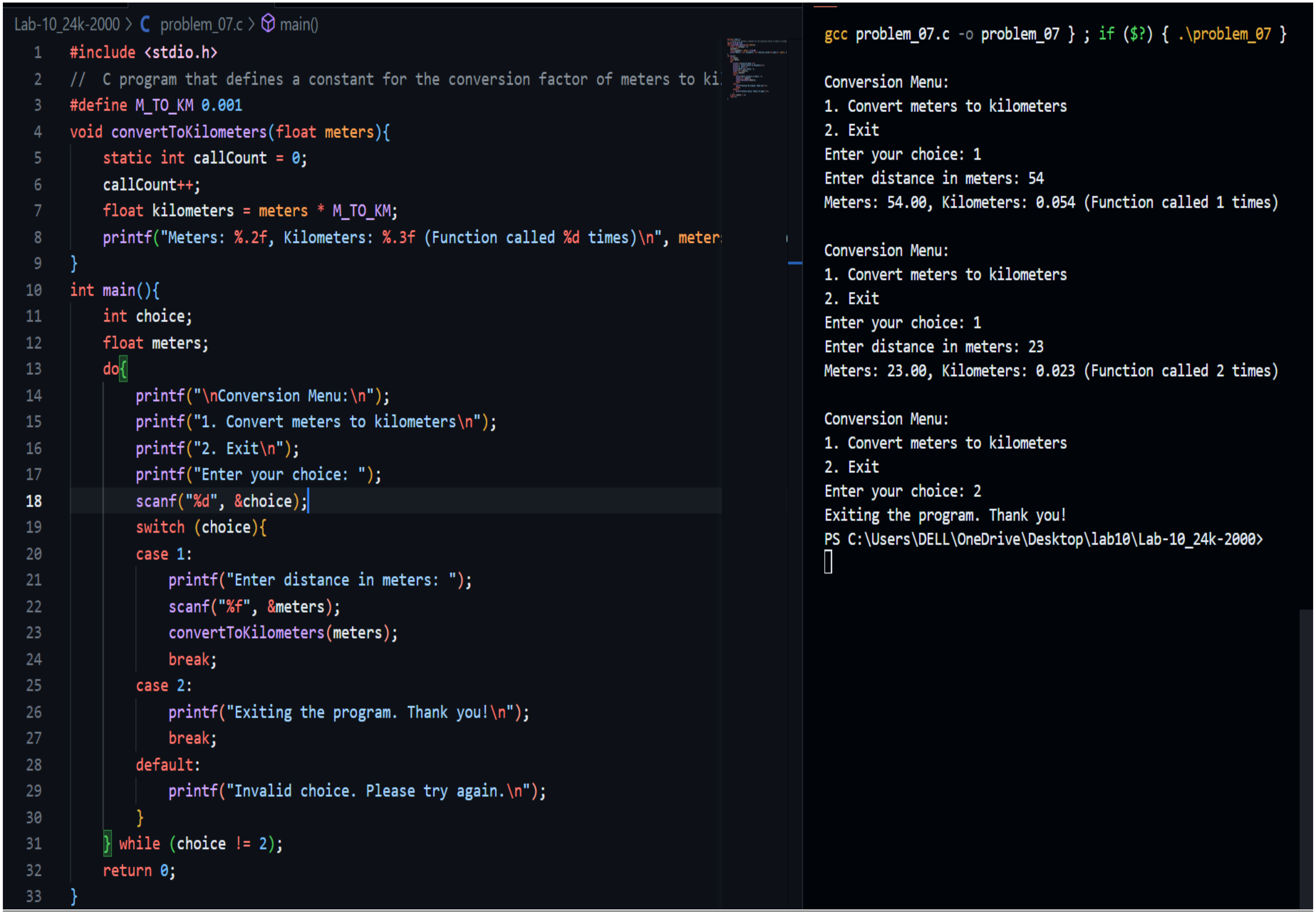


Lab-10_24k-2000 > C problem_06.c > addPackage(TravelPackage [], int)

```c
void bookPackage(TravelPackage packages[], int packageCount) {
    for (int i = 0; i < packageCount; i++) {

    printf("Package not found.\n");
}

int main() {
    TravelPackage packages[MAX_PACKAGES];
    int packageCount = 0;
    int choice;

    do {
        printf("\nTravel Package Management System\n");
        printf("1. Add Travel Package\n");
        printf("2. Display Available Packages\n");
        printf("3. Book a Travel Package\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                packageCount = addPackage(packages, packageCount);
                break;
            case 2:
                displayPackages(packages, packageCount);
                break;
            case 3:
                bookPackage(packages, packageCount);
                break;
            case 4:
                printf("Exiting the program. Thank you!\n");
                break;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    } while (choice != 4);

    return 0;
}
```

```
Enter the name of the package to book: a
sal movers
Package not found.

Travel Package Management System
1. Add Travel Package
2. Display Available Packages
3. Book a Travel Package
4. Exit
Enter your choice: 3
Enter the name of the package to book: f
asal movers
Booking successful! Seats remaining: 9

Travel Package Management System
1. Add Travel Package
2. Display Available Packages
3. Book a Travel Package
4. Exit
Enter your choice: 2

Available Travel Packages:
Package 1:
  Name: fasal movers
  Destination: Lahore
  Duration: 2 days
  Cost: $50.00
  Seats Available: 9
-------------------------

Travel Package Management System
1. Add Travel Package
2. Display Available Packages
3. Book a Travel Package
4. Exit
Enter your choice: 
```

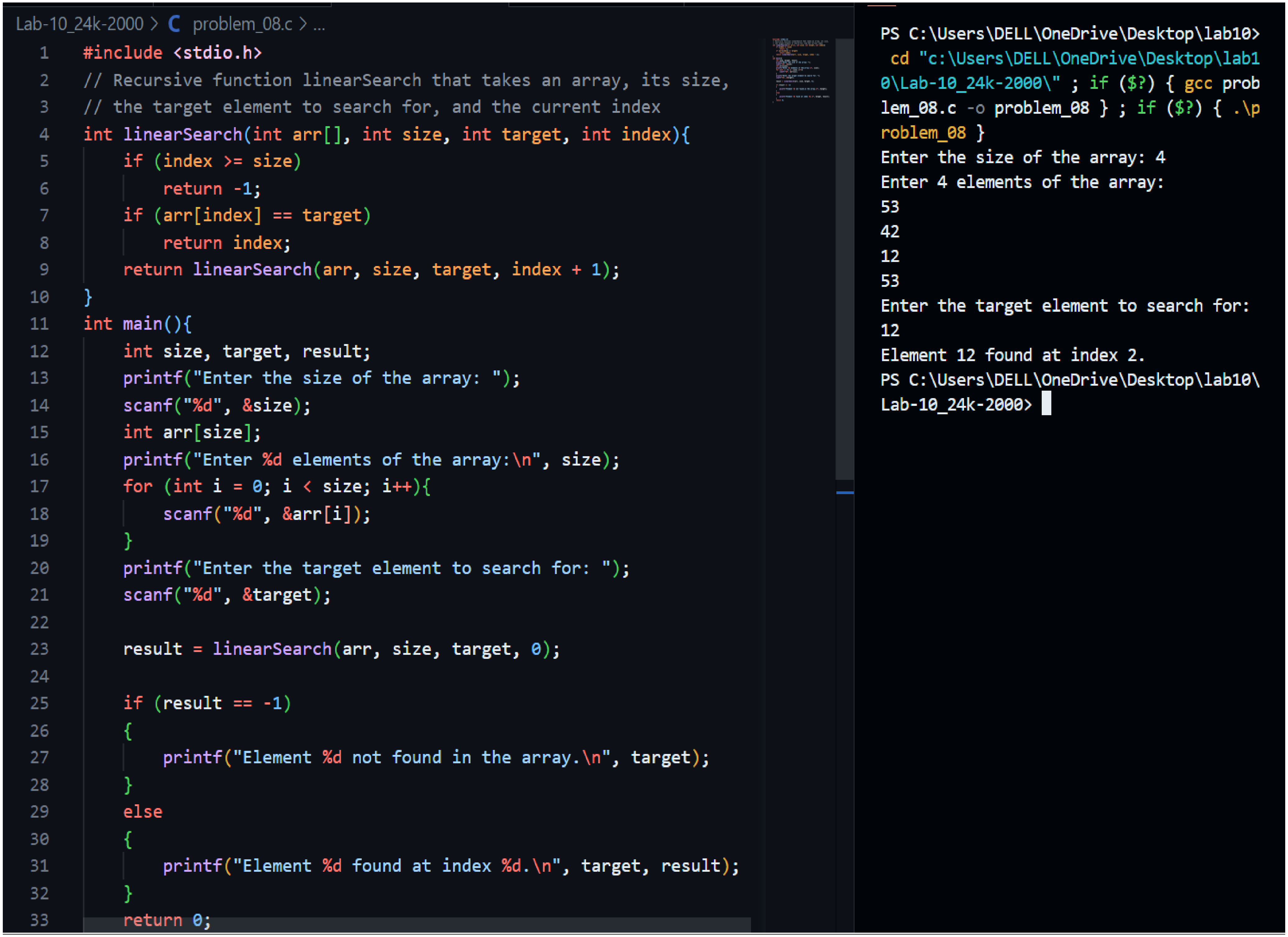
Problem 07:



```c
#include <stdio.h>
//  C program that defines a constant for the conversion factor of meters to ki
#define M_TO_KM 0.001
void convertToKilometers(float meters){
    static int callCount = 0;
    callCount++;
    float kilometers = meters * M_TO_KM;
    printf("Meters: %.2f, Kilometers: %.3f (Function called %d times)\n", meter
}
int main(){
    int choice;
    float meters;
    do{
        printf("\nConversion Menu:\n");
        printf("1. Convert meters to kilometers\n");
        printf("2. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice){
        case 1:
            printf("Enter distance in meters: ");
            scanf("%f", &meters);
            convertToKilometers(meters);
            break;
        case 2:
            printf("Exiting the program. Thank you!\n");
            break;
        default:
            printf("Invalid choice. Please try again.\n");
        }
    } while (choice != 2);
    return 0;
}
```

```
gcc problem_07.c -o problem_07 } ; if ($?) { .\problem_07 }

Conversion Menu:
1. Convert meters to kilometers
2. Exit
Enter your choice: 1
Enter distance in meters: 54
Meters: 54.00, Kilometers: 0.054 (Function called 1 times)

Conversion Menu:
1. Convert meters to kilometers
2. Exit
Enter your choice: 1
Enter distance in meters: 23
Meters: 23.00, Kilometers: 0.023 (Function called 2 times)

Conversion Menu:
1. Convert meters to kilometers
2. Exit
Enter your choice: 2
Exiting the program. Thank you!
PS C:\Users\DELL\OneDrive\Desktop\lab10\Lab-10_24k-2000>
```

Problem 08:



```c
#include <stdio.h>
// Recursive function linearSearch that takes an array, its size,
// the target element to search for, and the current index
int linearSearch(int arr[], int size, int target, int index){
    if (index >= size)
        return -1;
    if (arr[index] == target)
        return index;
    return linearSearch(arr, size, target, index + 1);
}
int main(){
    int size, target, result;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int arr[size];
    printf("Enter %d elements of the array:\n", size);
    for (int i = 0; i < size; i++){
        scanf("%d", &arr[i]);
    }
    printf("Enter the target element to search for: ");
    scanf("%d", &target);

    result = linearSearch(arr, size, target, 0);

    if (result == -1)
    {
        printf("Element %d not found in the array.\n", target);
    }
    else
    {
        printf("Element %d found at index %d.\n", target, result);
    }
    return 0;
```

```
PS C:\Users\DELL\OneDrive\Desktop\lab10>
 cd "c:\Users\DELL\OneDrive\Desktop\lab1
0\Lab-10_24k-2000\" ; if ($?) { gcc prob
lem_08.c -o problem_08 } ; if ($?) { .\p
roblem_08 }
Enter the size of the array: 4
Enter 4 elements of the array:
53
42
12
53
Enter the target element to search for:
12
Element 12 found at index 2.
PS C:\Users\DELL\OneDrive\Desktop\lab10\
Lab-10_24k-2000>
```