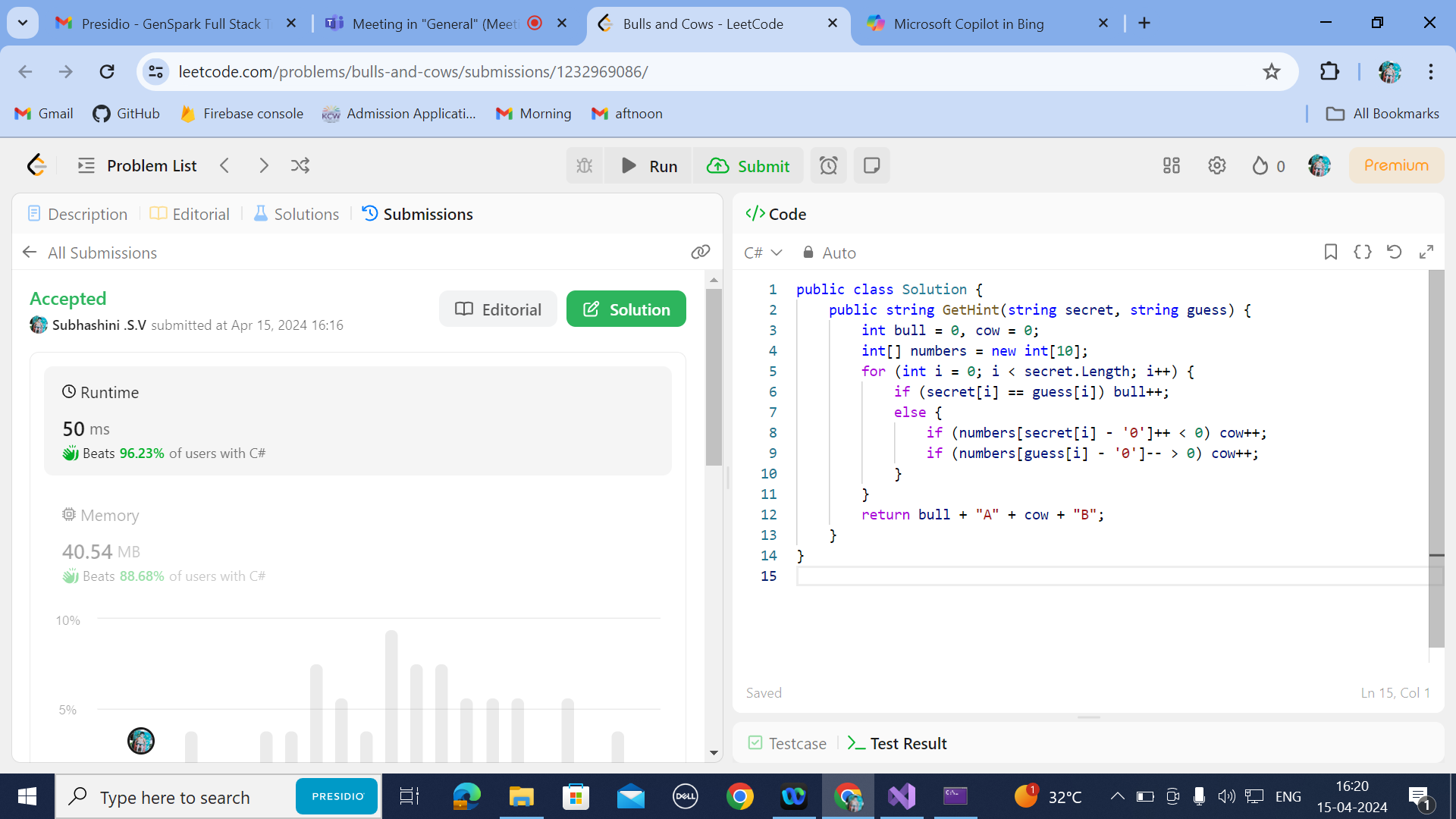
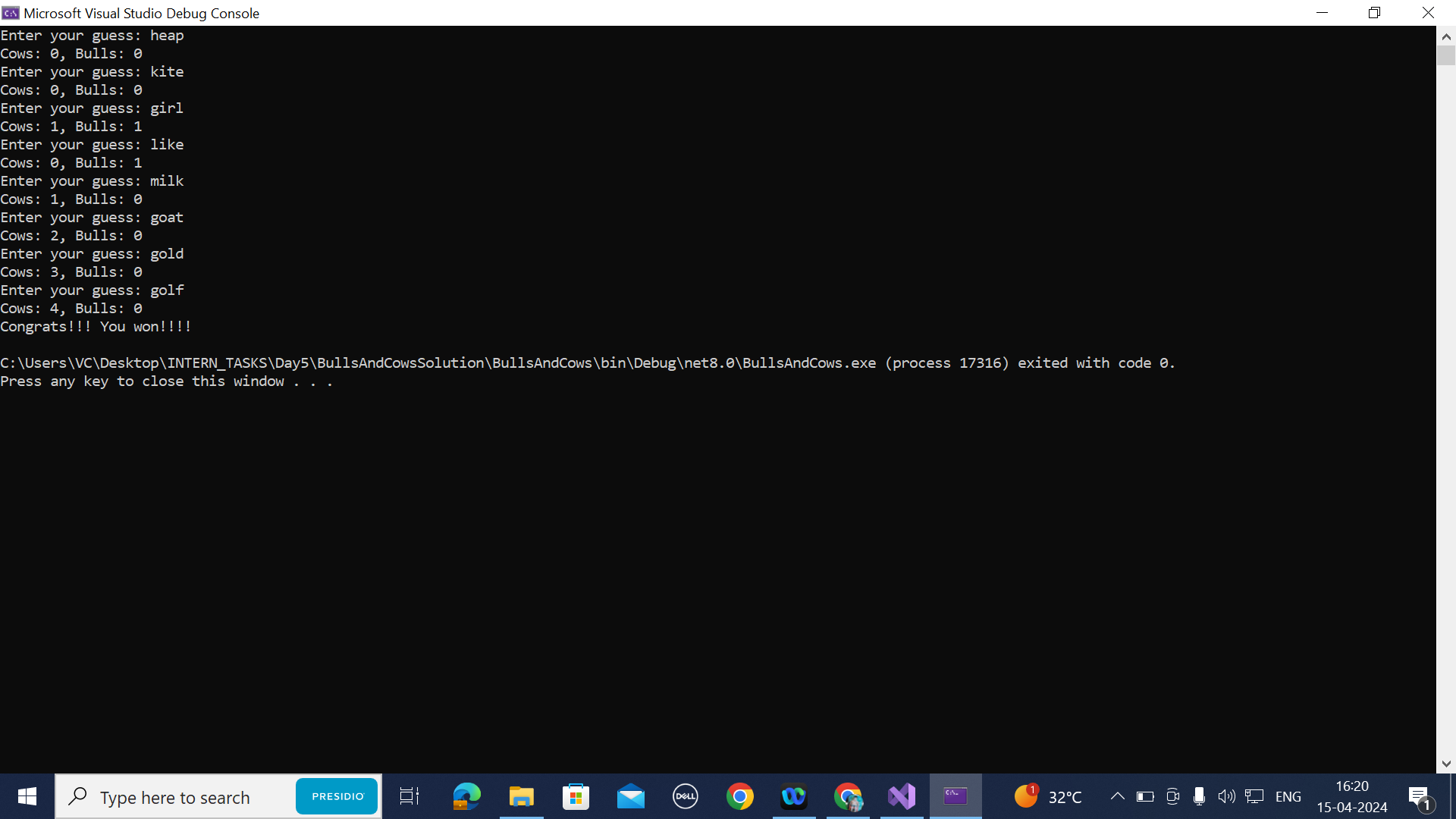
DAY 5  
COWS AND BULLS LEETCODE



COWS AND BULL USING VS

OUTPUT:



CODING:

namespace BullsAndCows

{

internal class Program

{

static void Main(string[] args)

{

string targetWord = "golf";

string guessWord;

int cows, bulls;

while (true)

{

Console.Write("Enter your guess: ");

guessWord = Console.ReadLine();

if (guessWord.Length != 4)

{

Console.WriteLine("Please enter a 4-letter word.");

continue;

}

cows = bulls = 0;

for (int i = 0; i < 4; i++)

{

if (guessWord[i] == targetWord[i])

{

cows++;

}

else

{

for (int j = 0; j < 4; j++)

{

if (i != j && guessWord[i] == targetWord[j])

{

bulls++;

break;

}

}

}

}

if (cows == 4)

{

Console.WriteLine($"Cows: {cows}, Bulls: {bulls}");

Console.WriteLine("Congrats!!! You won!!!!");

break;

}

else

{

Console.WriteLine($"Cows: {cows}, Bulls: {bulls}");

}

}

}

}

}

**UNDERSTANDING LOOPS,SWITCHCASE,ARRAY:**

**CODING:**

using RequestTrackerAppModel;

namespace RequestTrackerApp

{

internal class Program

{

void UnderstandingArrays()

{

int[] numbers = { 200, 666, 960, 777, 665, 681 };

int countOfRepeatingNumbers = 0;

for (int i = 0; i < numbers.Length; i++)

{

int firstNumber, secondNumber,thirdNumber;

firstNumber = numbers[i] / 100;

secondNumber = (numbers[i] /10)%10;

thirdNumber = numbers[i] % 10;

if (firstNumber == secondNumber&&secondNumber==thirdNumber)

countOfRepeatingNumbers++;

}

Console.WriteLine("The number of repeating numbers is " + countOfRepeatingNumbers);

}

void UnderstandingSequenceStatments()

{

int num1, num2;

num1 = 100;

num2 = 20;

int num3 = num1 / num2;

Console.WriteLine($"The result of {num1} divided by {num2} is {num3}");

}

void UnderstandingForLoops()

{ for(int i=0;i<5;i++)

{

Console.WriteLine("Hello"+i);

}

}

void UnderstandingDoWhileLoops()

{ int count=-1;

do {

Console.WriteLine("In Do while the value is " + count);

Console.WriteLine("Please enter the number");

count = Convert.ToInt32(Console.ReadLine());

} while(count>0);

}

void UnderstandingWhileLoops()

{

int count = 10;

while (count > 0)

{

count--;

if (count == 7)

continue;

Console.WriteLine("Thje value of count is " + count);

if (count == 4)

break;

}

}

void UnderstandingIf()

{

string str = "Ramu";

if (str == "Ramu")

{

Console.WriteLine("Welcome");

}

else if (str == "Somu")

{

Console.WriteLine("You have basic access");

}

else

{

Console.WriteLine("You are not allowed");

}

}

void UnderstandingSwitchCase() {

Console.WriteLine("Please enter a number for day");

int choice = Convert.ToInt32(Console.ReadLine());

switch (choice)

{

case 1:

Console.WriteLine("Monday");

break;

case 2:

Console.WriteLine("Tuesday");

break;

case 3:

Console.WriteLine("Wednesday");

break;

case 4:

Console.WriteLine("Thursday");

break;

case 5:

Console.WriteLine("Friday");

break;

case 6:

Console.WriteLine("Saturday");

break;

case 7:

Console.WriteLine("Sunday");

break;

default:

Console.WriteLine("Try again");

break;

}

}

static void Main(string[] args)

{

Program program = new Program();

program.UnderstandingSequenceStatments();

program.UnderstandingIf();

program.UnderstandingSwitchCase();

program.UnderstandingForLoops();

program.UnderstandingWhileLoops();

program.UnderstandingDoWhileLoops();

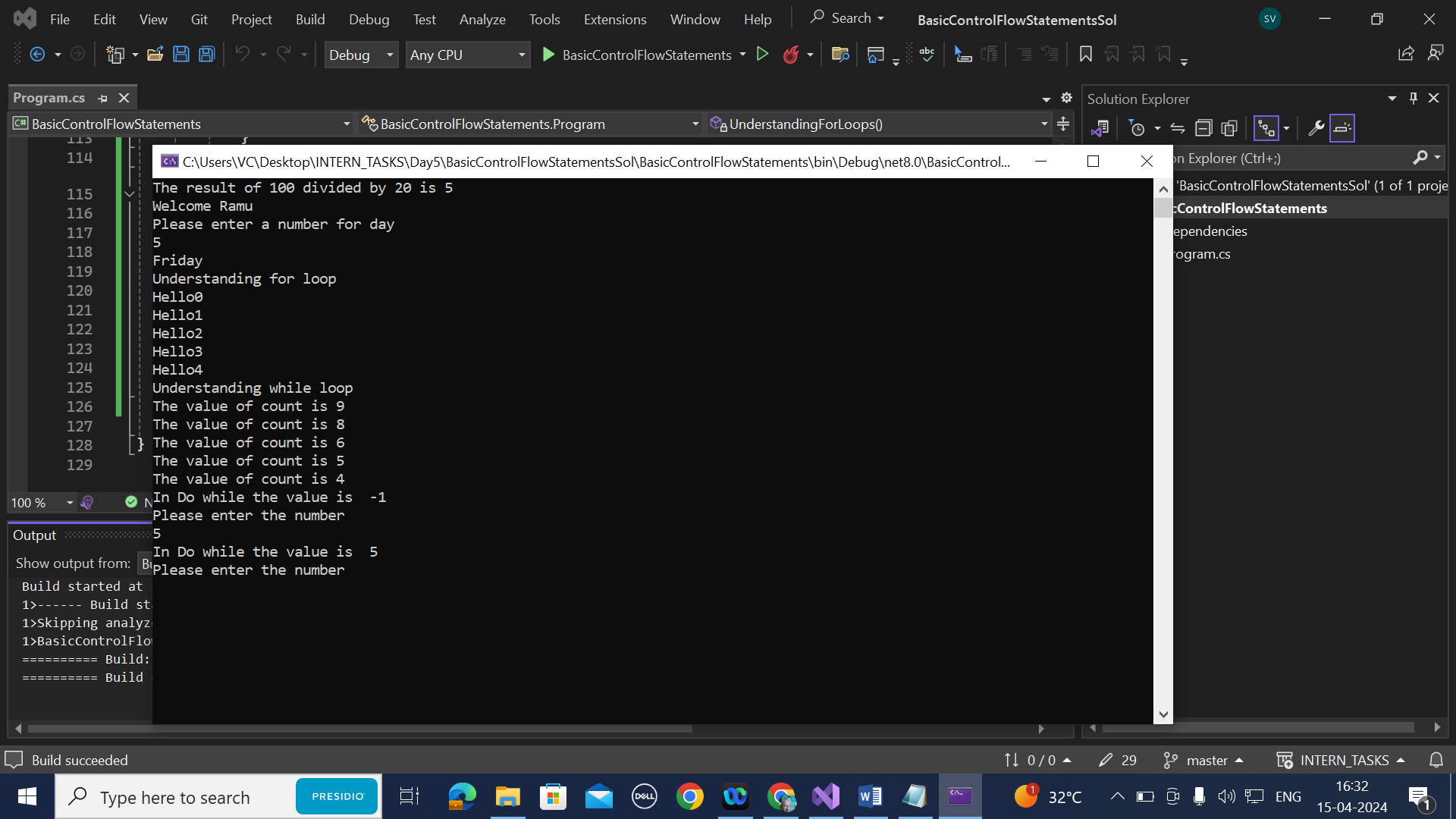
program.UnderstandingArrays();

}

}

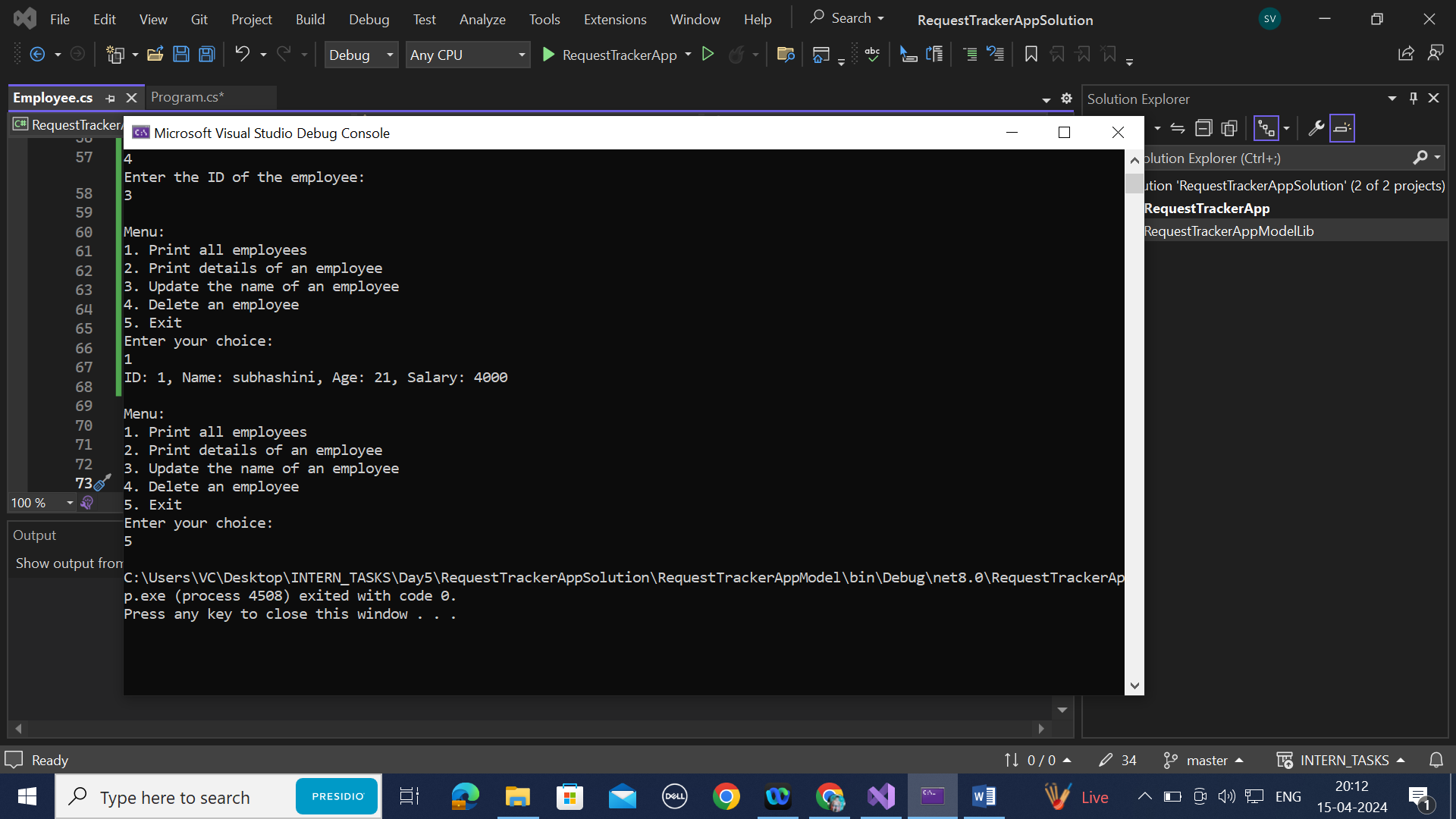
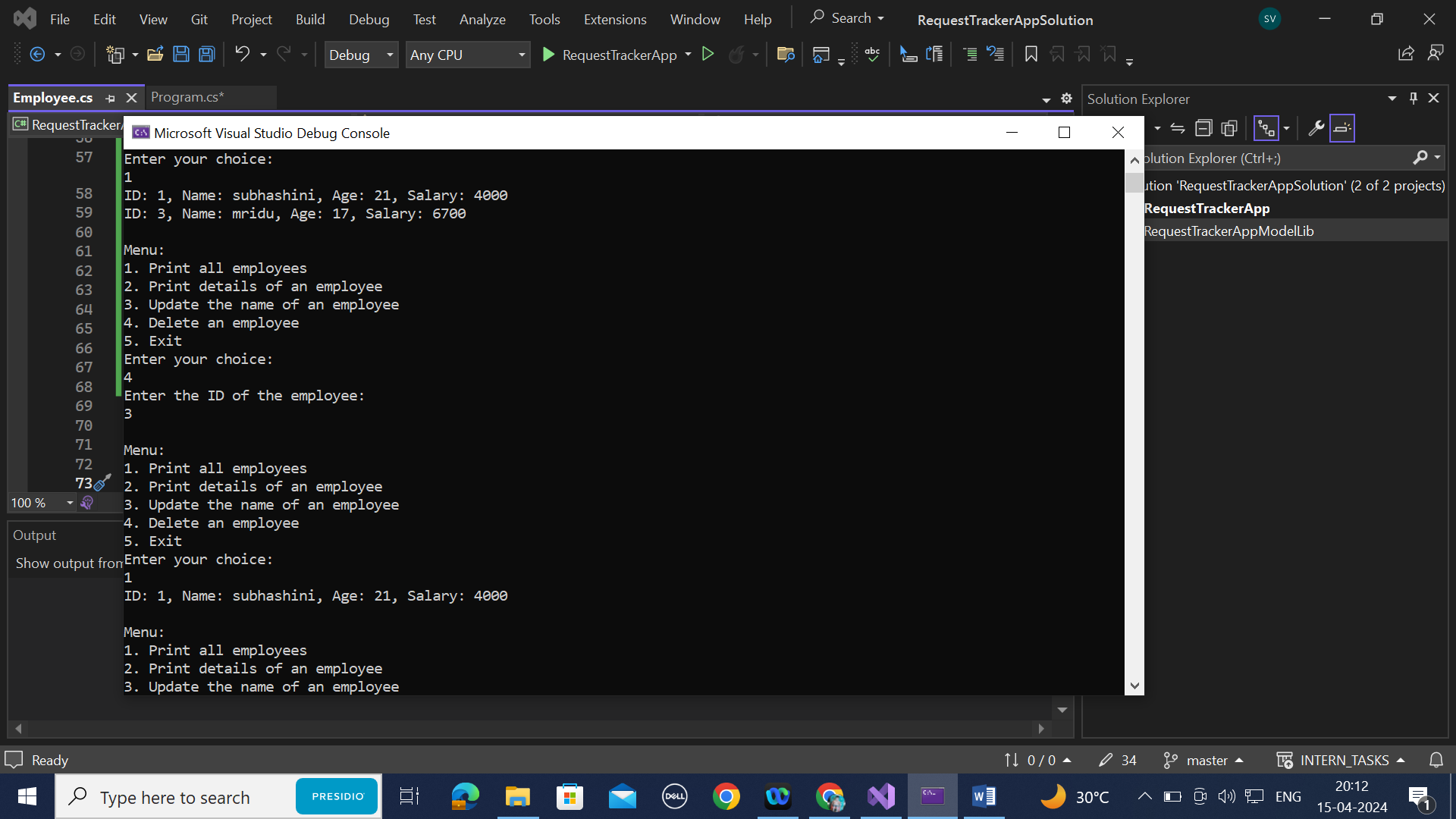
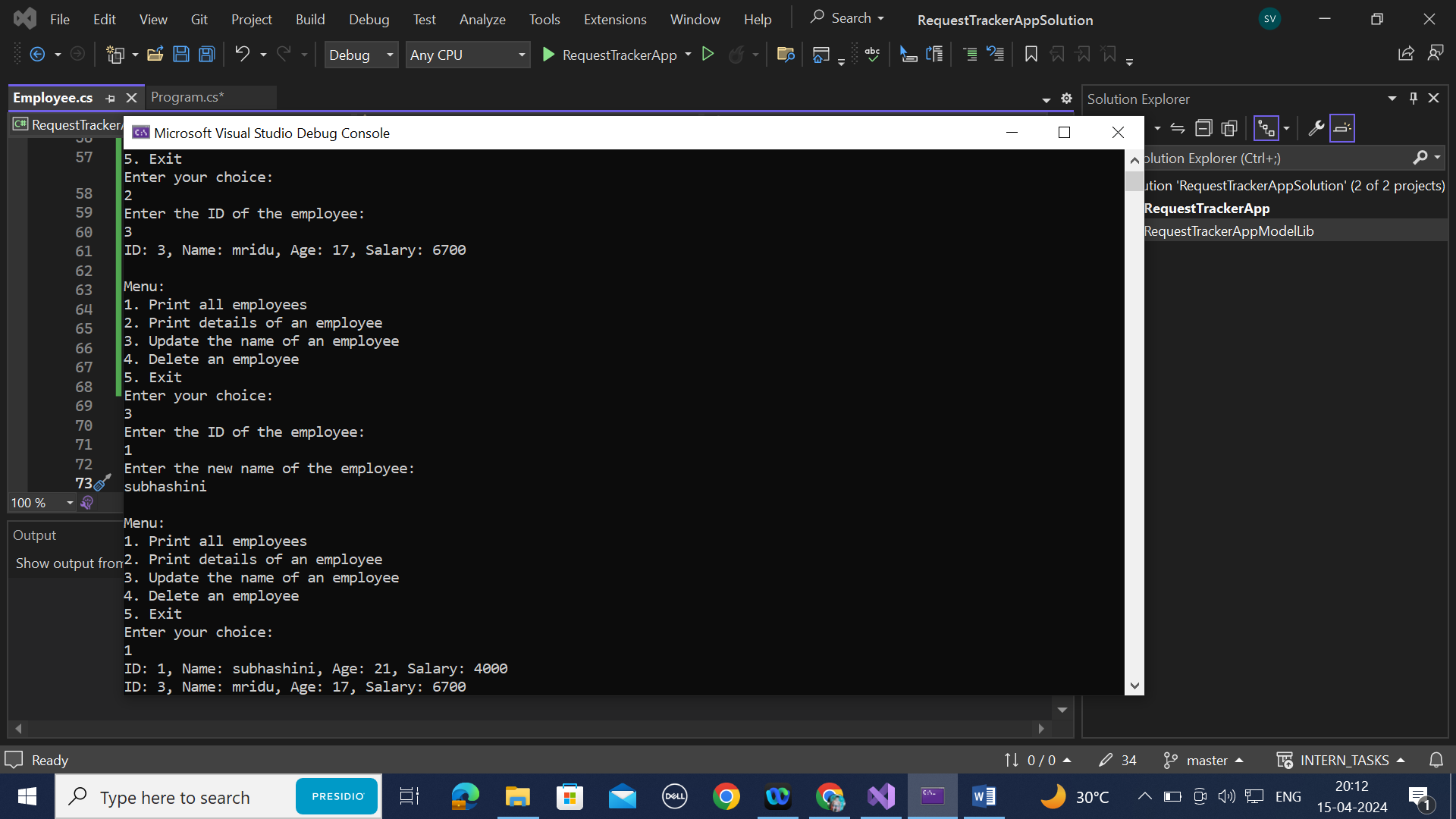
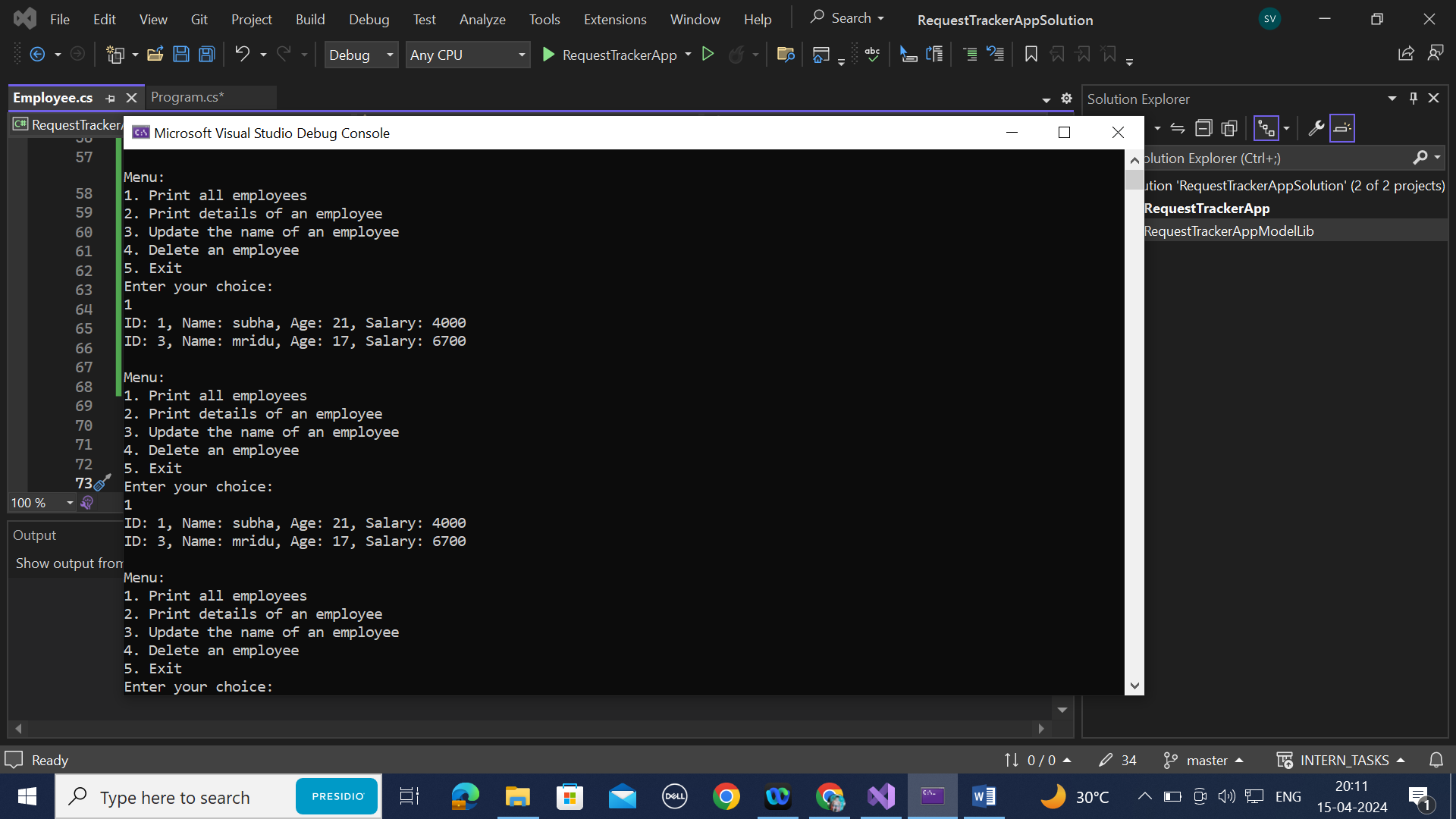
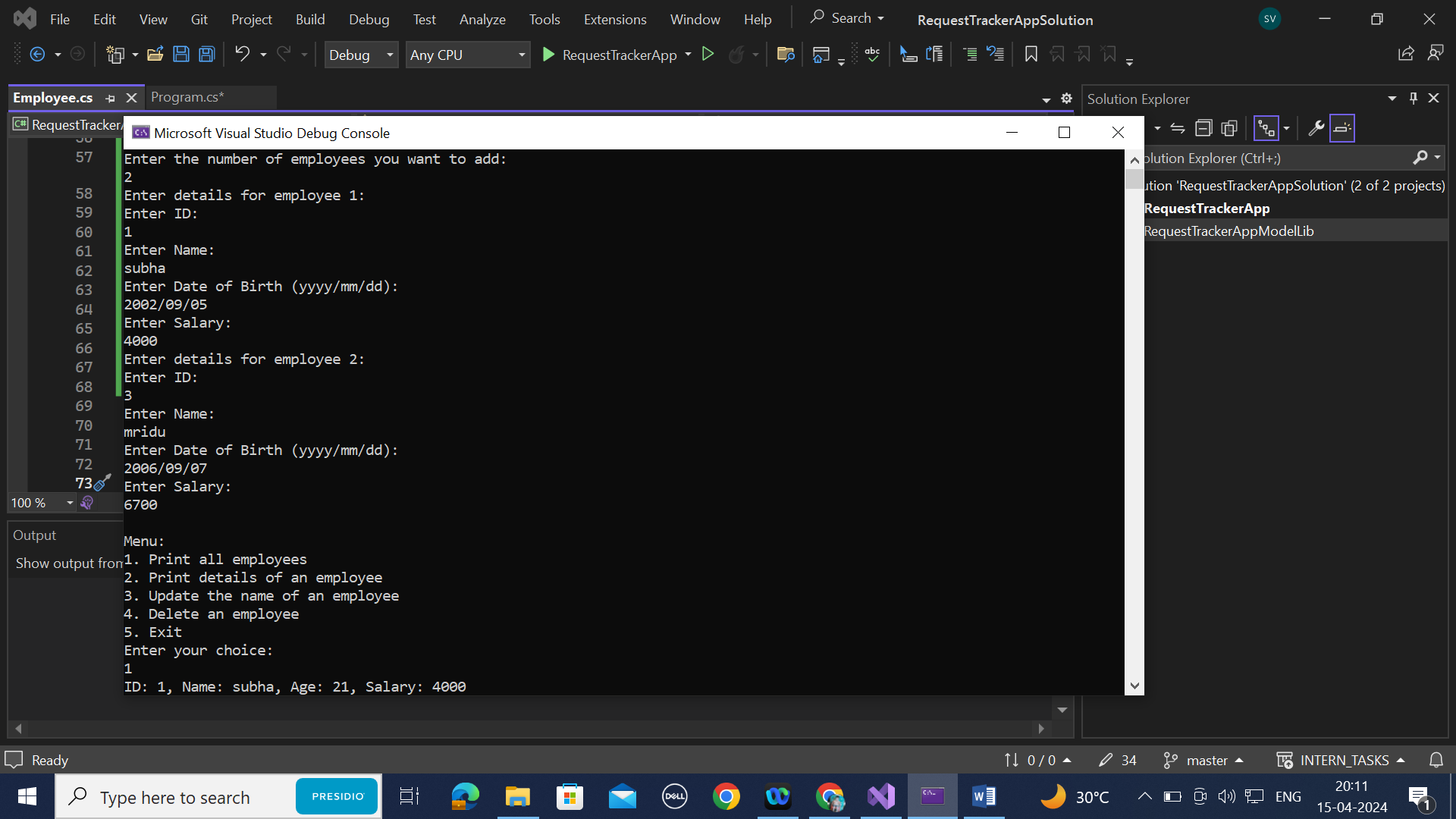
}

OUTPUT:



REQUEST TRACKER APPLICATION:

OUTPUT:



CODING:

Employee.cs

namespace RequestTrackerModelLib

{

public class Employee

{

int age;

DateTime dob;

public int Id { get; set; }

public string Name { get; set; }

public int Age

{

get

{

return age;

}

}

public DateTime DateOfBirth

{

get => dob;

set

{

dob = value;

age = ((DateTime.Today - dob).Days) / 365;

}

}

public double Salary

{

get; set;

}

public Employee()

{

Id = 0;

Name = string.Empty;

Salary = 0.0;

DateOfBirth = new DateTime();

}

public Employee(int id, string name, DateTime dateOfBirth, double salary)

{

Id = id;

Name = name;

DateOfBirth = dateOfBirth;

Salary = salary;

}

public void BuildEmployeeFromConsole()

{

Console.WriteLine("Please enter the Name");

Name = Console.ReadLine() ?? String.Empty;

Console.WriteLine("Please enter Date of Birth");

DateOfBirth = Convert.ToDateTime(Console.ReadLine());

Console.WriteLine("Pleaseenter the basic salary");

Salary = Convert.ToDouble(Console.ReadLine());

}

public void PrintEmployeeDetails()

{

Console.WriteLine("Employee Id:" + Id);

Console.WriteLine("Employee Name:" + Name);

Console.WriteLine("Date of birth : " + DateOfBirth);

Console.WriteLine("Employee Age : " + Age);

Console.WriteLine("Employee Salary : Rs." + Salary);

}

}

}

Program.cs

using RequestTrackerModelLib;

using System.Globalization;

namespace RequestTrackerApp

{

internal class Program

{

private List<Employee> employees = new List<Employee>();

public Employee BuildEmployee(int id, string name, DateTime dob, double salary)

{

return new Employee(id, name, dob, salary);

}

public void AddEmployee(Employee employee)

{

employees.Add(employee);

}

public void PrintEmployee(Employee employee)

{

Console.WriteLine($"ID: {employee.Id}, Name: {employee.Name}, Age: {employee.Age}, Salary: {employee.Salary}");

}

public void PrintAllEmployees()

{

foreach (var employee in employees)

{

PrintEmployee(employee);

}

}

public int GetEmployeeId(Employee employee)

{

return employee.Id;

}

public Employee SearchEmployee(int id)

{

for (int i = 0; i < employees.Count; i++)

{

if (employees[i].Id == id)

{

return employees[i];

}

}

return null; // Return null if no employee with the given ID is found

}

public void PrintEmployeeDetails(int id)

{

var employee = SearchEmployee(id);

if (employee != null)

{

PrintEmployee(employee);

}

else

{

Console.WriteLine("Employee not found");

}

}

public void UpdateName(int id, string newName)

{

var employee = SearchEmployee(id);

if (employee != null)

{

employee.Name = newName;

}

else

{

Console.WriteLine("Employee not found");

}

}

public void DeleteEmployee(int id)

{

if (employees == null)

{

Console.WriteLine("Employees list is not initialized");

return;

}

var employee = SearchEmployee(id);

if (employee != null)

{

int index = employees.IndexOf(employee);

if (index != -1)

{

employees.RemoveAt(index);

}

}

else

{

Console.WriteLine("Employee not found");

}

}

static void Main(string[] args)

{

Program program = new Program();

Console.WriteLine("Enter the number of employees you want to add:");

int numEmployees = Convert.ToInt32(Console.ReadLine());

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

{

Console.WriteLine($"Enter details for employee {i + 1}:");

Console.WriteLine("Enter ID:");

int id = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Name:");

string name = Console.ReadLine();

Console.WriteLine("Enter Date of Birth (yyyy/mm/dd):");

DateTime dob = DateTime.ParseExact(Console.ReadLine(), "yyyy/MM/dd", CultureInfo.InvariantCulture);

Console.WriteLine("Enter Salary:");

double salary = Convert.ToDouble(Console.ReadLine());

Employee emp = program.BuildEmployee(id, name, dob, salary);

program.AddEmployee(emp);

}

while (true)

{

Console.WriteLine("\nMenu:");

Console.WriteLine("1. Print all employees");

Console.WriteLine("2. Print details of an employee");

Console.WriteLine("3. Update the name of an employee");

Console.WriteLine("4. Delete an employee");

Console.WriteLine("5. Exit");

Console.WriteLine("Enter your choice:");

int choice = Convert.ToInt32(Console.ReadLine());

switch (choice)

{

case 1:

program.PrintAllEmployees();

break;

case 2:

Console.WriteLine("Enter the ID of the employee:");

int id = Convert.ToInt32(Console.ReadLine());

program.PrintEmployeeDetails(id);

break;

case 3:

Console.WriteLine("Enter the ID of the employee:");

id = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the new name of the employee:");

string newName = Console.ReadLine();

program.UpdateName(id, newName);

break;

case 4:

Console.WriteLine("Enter the ID of the employee:");

id = Convert.ToInt32(Console.ReadLine());

program.DeleteEmployee(id);

break;

case 5:

return;

default:

Console.WriteLine("Invalid choice. Please enter a number between 1 and 5.");

break;

}

}

}

}

}