Date:15/04/2024  
  
task1:First three digits Repeated Numbers  
  
```

using System;

namespace \_1\_findthreedigit\_repetedno

{

internal class Program

{

void findrepetednums()

{

int count = 0;

int[] nums = { 111, 222, 3325, 4135, 1355 };

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

{

if (nums[i] > 100 && nums[i] < 1000)

{

if (nums[i] % 111 == 0 && nums[i] != 0)

{

count++;

}

}

}

Console.WriteLine($"The repeated three digits numbers are {count}");

}

static void Main(string[] args)

{

Program program = new Program();

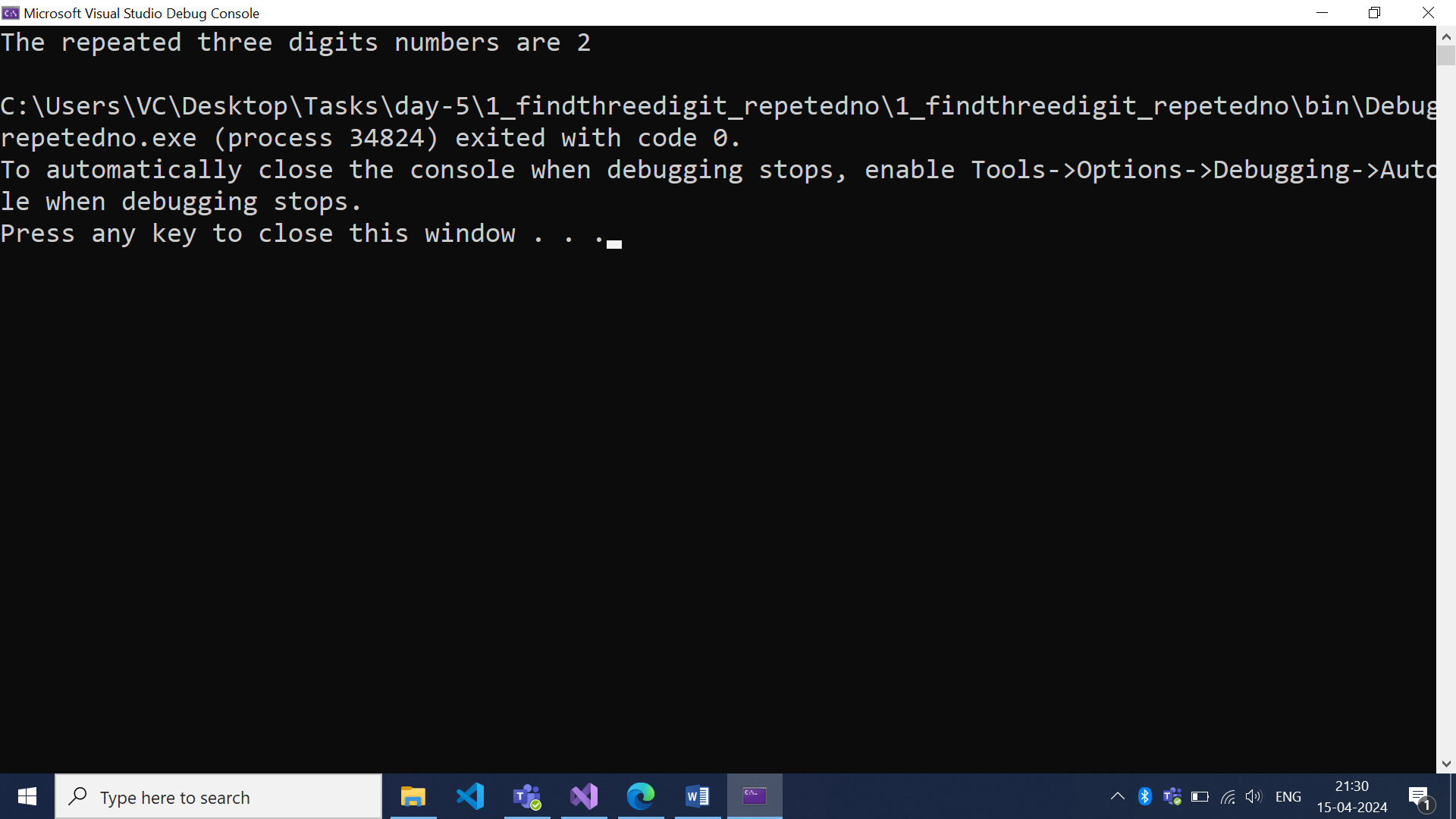
program.findrepetednums();

}

}

}

```



Task:2 Bull and cows

```

namespace \_2\_Bullsand\_cows

{

internal class Program

{

static void Main(string[] args)

{

Cowsandbulls();

}

private static void Cowsandbulls()

{

int cow = 0,bull=0;

Console.WriteLine("Enter a word for the game:");

string word=Console.ReadLine();

while(word.Length!=4)

{

Console.WriteLine("pls Enter 4 character:");

word = Console.ReadLine();

}

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("START GAME");

while (cow != 4)

{

cow = 0; bull = 0;

Console.WriteLine("guess the 4 char word !!");

string guess = Console.ReadLine();

if (guess.Length != 4)

{

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

continue;

}

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

{

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

{

if (word[i] == guess[j] && i == j) cow++;

else if (word[i] == guess[j] ) bull++;

}

}

Console.WriteLine($"cow is {cow}, bull is {bull}");

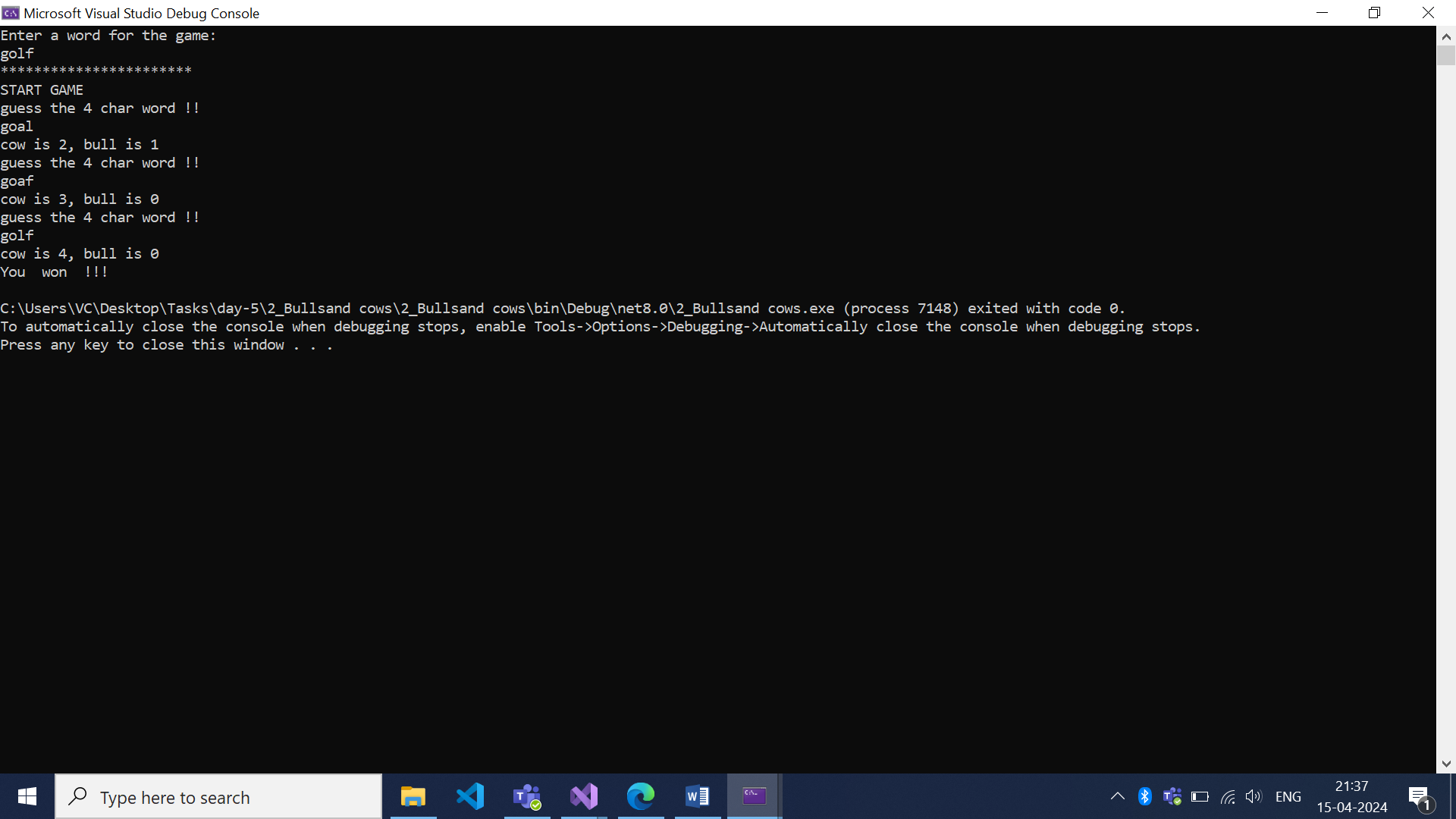
}

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

}

}

}

```  
  


Task 3: CRUD opreations   
  
```

using Request\_tracker\_library;

namespace RequestTrackerApplication

{

internal class Program

{

Employee[] employees;

public Program()

{

employees = new Employee[1];

}

void PrintMenu()

{

Console.WriteLine("1. Add Employee");

Console.WriteLine("2. Print Employees");

Console.WriteLine("3. Search Employee by ID");

Console.WriteLine("4. Update Employee by ID");

Console.WriteLine("5. Delete Employee by ID");

Console.WriteLine("0. Exit");

}

void EmployeeInteraction()

{

int choice = 0;

do

{

PrintMenu();

Console.WriteLine("Please select an option");

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

switch (choice)

{

case 0:

Console.WriteLine("Bye.....");

break;

case 1:

AddEmployee();

break;

case 2:

PrintAllEmployees();

break;

case 3:

SearchAndPrintEmployee();

break;

case 4:

EditUsername();

break;

case 5:

DeleteEmployee();

break;

default:

Console.WriteLine("Invalid choice. Try again");

break;

}

} while (choice != 0);

}

void AddEmployee()

{

if (employees[employees.Length - 1] != null)

{

Console.WriteLine("Sorry we have reached the maximum number of employees");

return;

}

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

{

if (employees[i] == null)

{

employees[i] = CreateEmployee(i);

}

}

}

void PrintAllEmployees()

{

if (employees[0] == null)

{

Console.WriteLine("No Employees available");

return;

}

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

{

if (employees[i] != null)

PrintEmployee(employees[i]);

}

}

Employee CreateEmployee(int id)

{

Employee employee = new Employee();

employee.Id = 101 + id;

employee.BuildEmployeeFromConsole();

return employee;

}

void PrintEmployee(Employee employee)

{

Console.WriteLine("---------------------------");

employee.PrintEmployeeDetails();

Console.WriteLine("---------------------------");

return ;

}

int GetIdFromConsole()

{

int id = 0;

Console.WriteLine("Please enter the employee Id");

while (!int.TryParse(Console.ReadLine(), out id))

{

Console.WriteLine("Invalid entry. Please try again");

}

return id;

}

Employee SearchAndPrintEmployee()

{

int id = GetIdFromConsole();

Employee employee = SearchEmployeeById(id);

if (employee == null)

{

Console.WriteLine("No such Employee is present");

return null;

}

PrintEmployee(employee);

return employee;

}

Employee SearchEmployeeById(int id)

{

Employee employee = null;

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

{

// if ( employees[i].Id == id && employees[i] != null)//Will lead to exception

if (employees[i] != null && employees[i].Id == id)

{

employee = employees[i];

break;

}

}

return employee;

}

void DeleteEmployee()

{

Console.WriteLine("Enter employee id to delete:");

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

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

{

if (employees[i] != null && employees[i].Id == empid)

{

employees[i] = null;

Console.WriteLine("Employee deleted successfully.");

return;

}

}

Console.WriteLine("Employee not found.");

}

void EditUsername()

{

Employee emp = SearchAndPrintEmployee();

Console.WriteLine("Enter new user name:");

String newuser=Console.ReadLine();

emp.Name = newuser;

PrintEmployee(emp);

}

static void Main(string[] args)

{

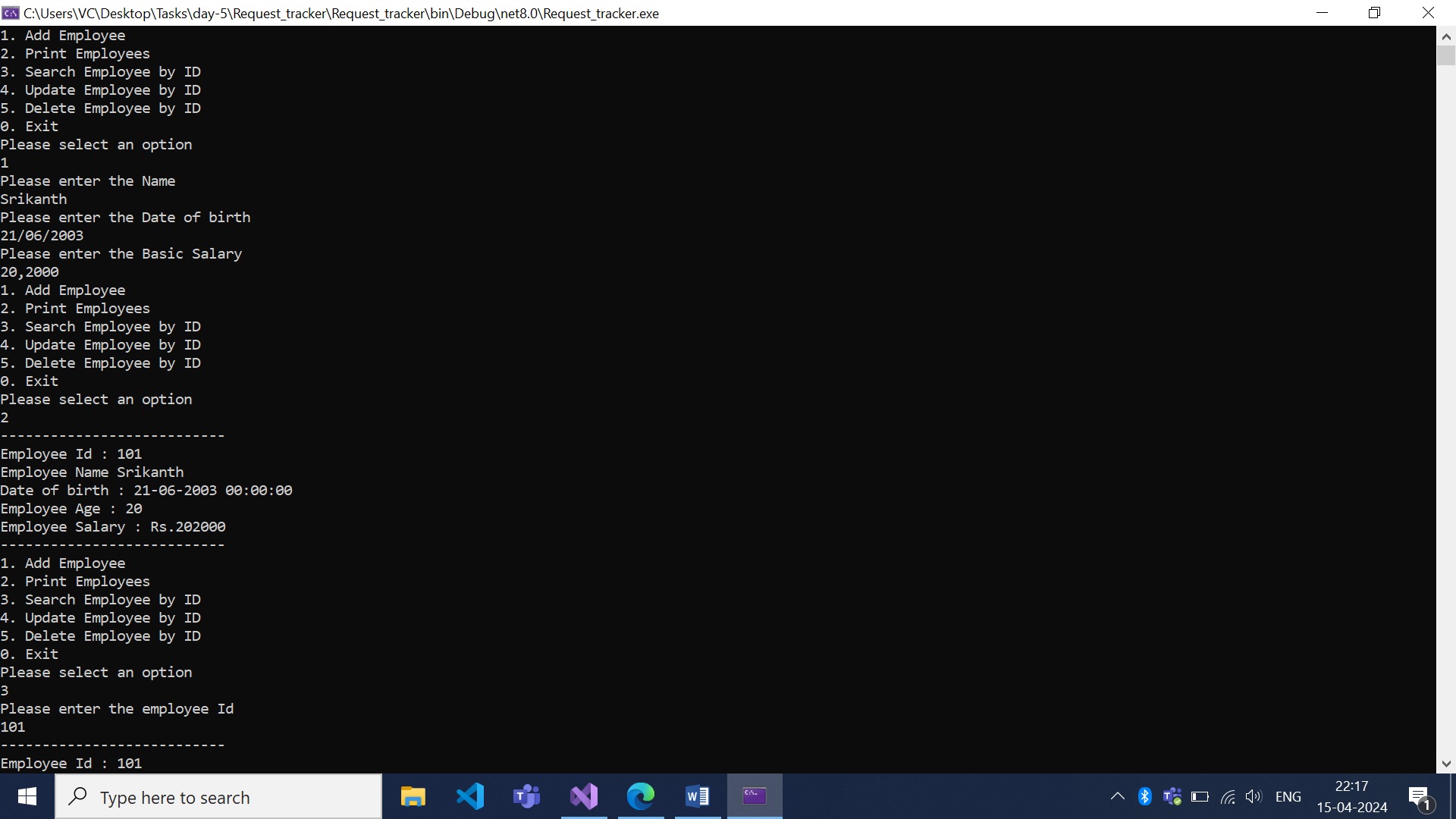
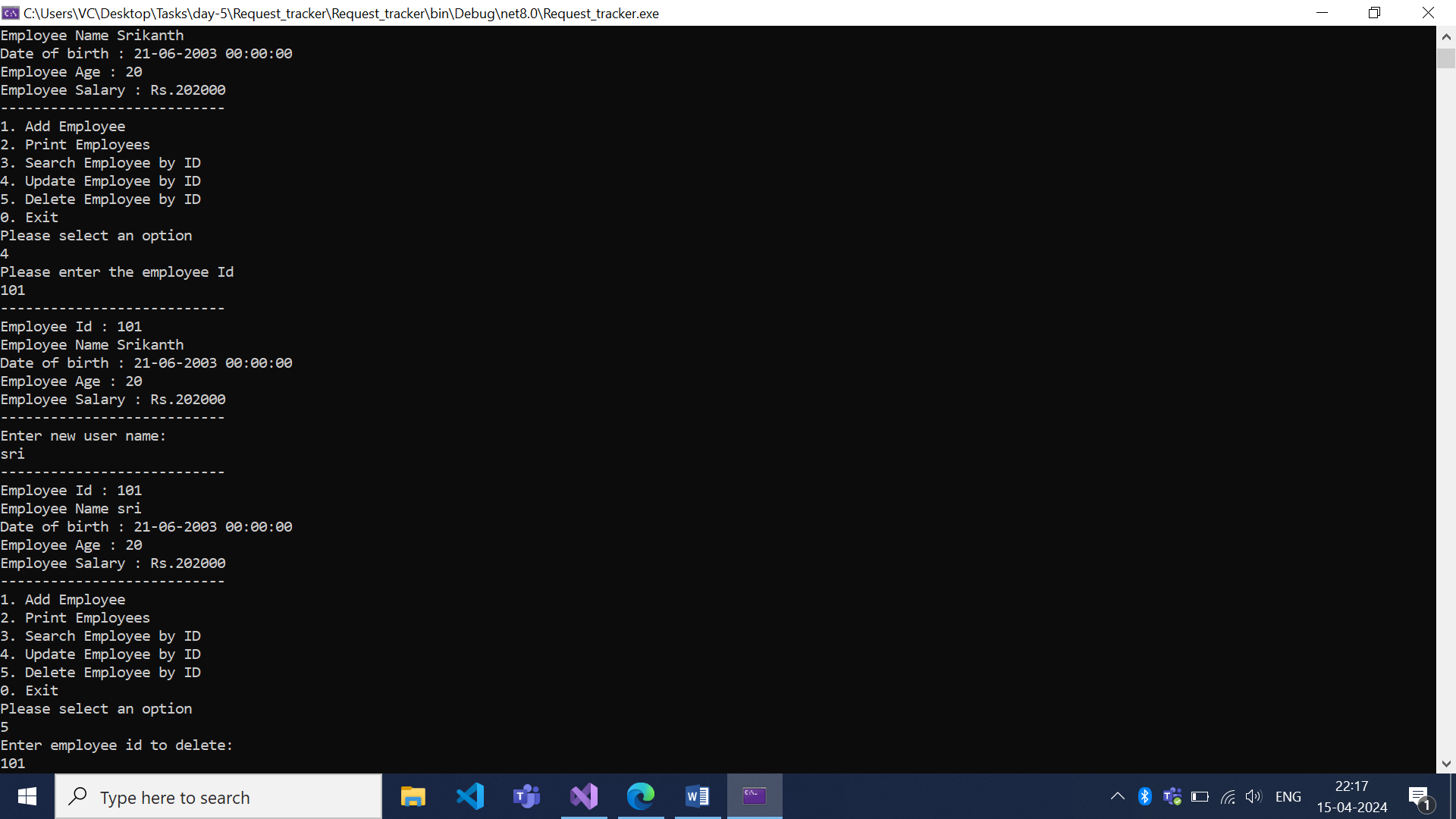
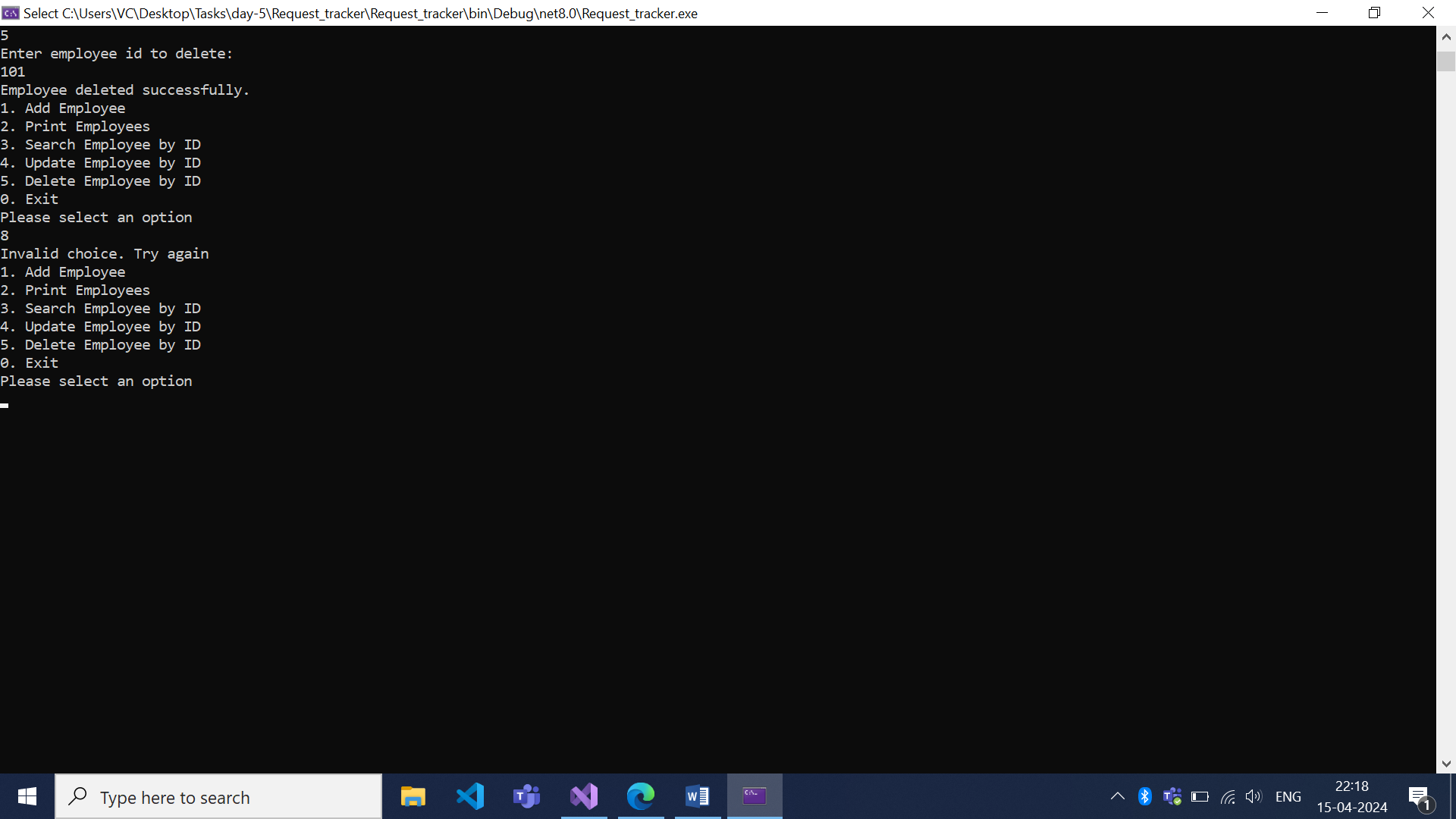
Program program = new Program();

program.EmployeeInteraction();

}

}

}

```  
  
  

Task 4: leetcode (bull and cow):

