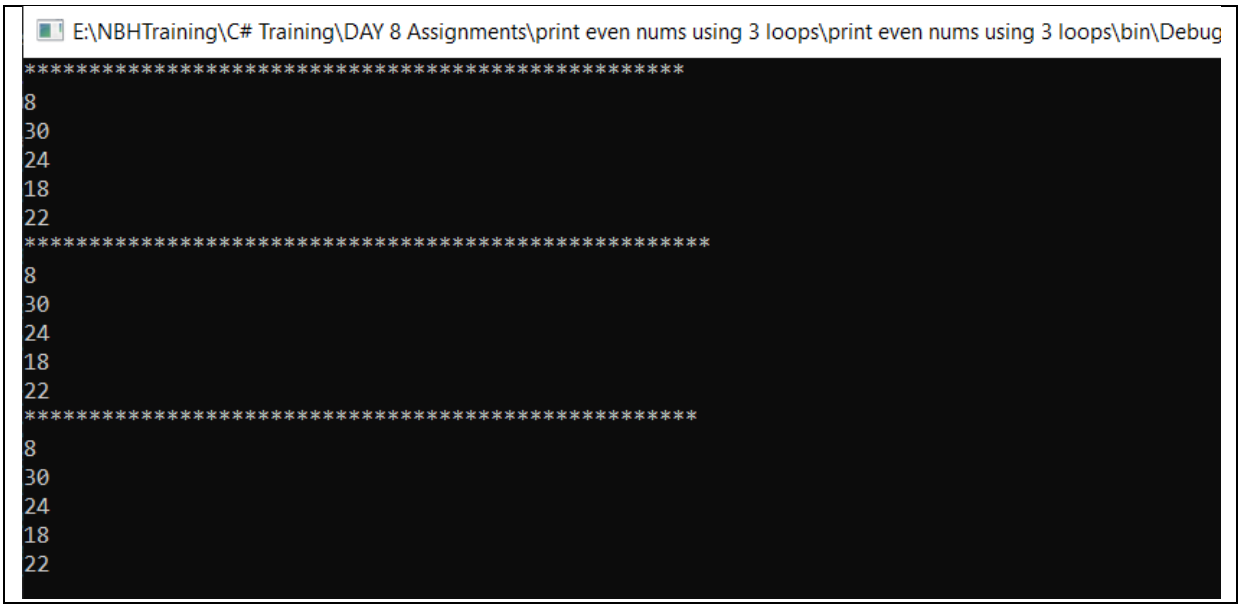
DAY 8 MNG ASSIGNMENT SUBMITTED
BY
POTUKANUMA JEEVITHA
02-02-2022

1. Declare and initialize a list with 8 values.write for loop, foreach loop, lambda to print even numbers.

Code:

```
using System;
using System.Collections.Generic;
using  System.Linq;
using  System.Text;
using System.Threading.Tasks;

namespace print_even_nums_using_3_loops
{
    internal class Program
    {
        static void Main(string[] args)
        {
            List<int> data = new List<int>() { 7, 8, 21, 30, 24, 18, 27, 22 };
            Console.WriteLine("**************************************************");
            //Even nums using FOR LOOP
            for (int i=0;i<data.Count;i++)
            {
                if(data[i]%2==0)
                    Console.WriteLine(data[i]);
            }

Console.WriteLine("**************************************************");
            // even nums using FOREACH LOOP
            foreach(var d in data)
{
                if (d%2==0)
                    Console.WriteLine(d);
            }

Console.WriteLine("**************************************************");
            // even nums using LAMBDA EXPRESSION
            data.Where(d=>d%2==0).ToList().ForEach(d => Console.WriteLine(d));
            Console.ReadLine();
        }
    }
}
```

Output:

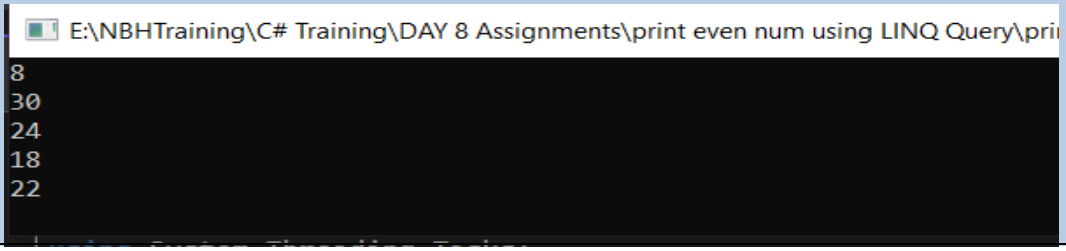


Declare and initialize a list with 8 values. Write LINQ query to print even numbers.

Code:

```
using System;
using System.Collections.Generic;
using  System.Linq;
using  System.Text;
using System.Threading.Tasks;

namespace print_even_num_using_LINQ_Query
{
    internal class Program
    {
        static void Main(string[] args)
        {
            List<int> data = new List<int>() { 7, 8, 21, 30, 24, 18, 27, 22 };
            // even nums using LINQ query
            var result = from d in data
                         where d%2==0
                         select  d;
            result.ToList().ForEach(d => Console.WriteLine(d));
            Console.ReadLine();
        }
    }
}
```

Output:



2.Create a class Employee with three variables as discussed in the class and create a list of Employees

public int id;
public string name;
public int salary;

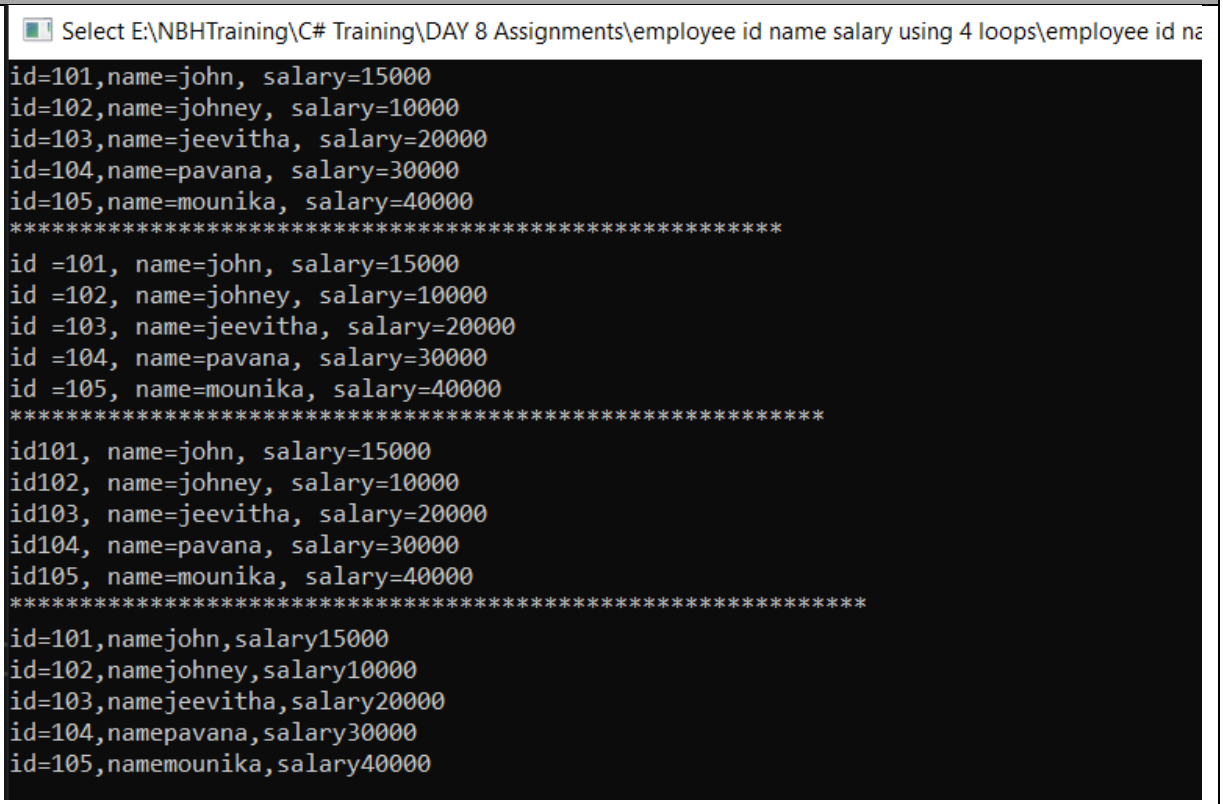
write for loop,foreach loop,lambda expression,linq query.

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace employee_id_name_salary_using_4_loops
{
    class Employee
    {
        public int Id;
        public string Name;
        public int salary;
    }


        internal class Program
    {


            static void Main(string[] args)
        {
            List<Employee> employees = new List<Employee>()
            {
                new Employee() { Id = 101, Name = "john", salary = 15000 },
                new Employee() { Id = 102, Name = "johney", salary = 10000 },
                new Employee() { Id = 103, Name = "jeevitha", salary = 20000 },
                new Employee() { Id = 104, Name = "pavana", salary = 30000 },
new Employee() { Id = 105, Name = "mounika", salary = 40000 },
            };
            // create employees using FOR LOOP
            for (int i = 0; i < employees.Count ; i++)
            {

Console.WriteLine($"id={employees[i].Id},name={employees[i].Name},
salary={employees[i].salary}");

            }
```

```
Console.WriteLine("*********************************************************");
            // create employeees using FOREACH LOOP
            foreach (var e in employees)
            {
                Console.WriteLine($"id ={e.Id}, name={e.Name},
salary={e.salary}");
            }

Console.WriteLine("*********************************************************");
            // create employees using LAMBDA EXPRESSION
            employees.ToList().ForEach(e => Console.WriteLine($"id{e.Id},
name={e.Name}, salary={e.salary}"));

Console.WriteLine("*********************************************************"
);
            // create employees using LINQ QUERY
                  var result = from e in employees

                               select e;
            result.ToList().ForEach(e => Console.WriteLine
($"id={e.Id},name{e.Name},salary{e.salary}"));
            Console.ReadLine();

        }
    }
}
```

Output:



3.Create a class Product and add variables id, name, price, brand.
print product (name and brand) whose price is more than 500 using for, foreach loop, lambda, linq query.

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace product_id_name_salary_using_4_loops
{

    class Product
    {
        public int Id;
        public string Name;
        public int Price;
        public string Brand;
    }
    internal class Program
    {


        static void Main(string[] args)
        {
          list<Product>  product = new list<Product>()
          {
                        new Product() { Id = 101, Name ="book", Price =30,
Brand="Pragathi"},
                        new Product() { Id = 102, Name ="fan",Price=1000,
Brand="USHA"},
                        new Product() { Id = 103, Name ="chocolate",Price=100,
Brand="Dairymilk"},
                        new Product() { Id = 104, Name ="pen",Price=10,
Brand="Cello"},
                        new Product() { Id = 105, Name ="watch",Price=2000,
Brand="Fastrack"}
                        };

Console.WriteLine("**************************************************************
*****");
            //price is >500 using FORLOOP

            for (int i = 0; i < product.Count; i++)
            {
                if (product[i].Price >= 500)

Console.WriteLine($"name={product[i].Name},Brand={product[i].Brand}");

            }

Console.WriteLine("**************************************************************
******");
            //price is >500 using FOREACH LOOP
            foreach (var e in product)
            {
                if (e.Price >= 500)
                    Console.WriteLine($"Name={e.Name},Brand{e.Brand}");
```

```
            }

Console.WriteLine("*************************************************************");
            //price is >500 using LAMBDA EXPRESSION
            product.ToList().Where(e => e.Price >= 500).ToList().ForEach(e =>
Console.WriteLine($"Name={e.Name},Brand={e.Brand}"));


Console.WriteLine("*************************************************************");
            //price is >500 using LINQ QUERY
            var result=from e in product
                       where e.Price >= 500
                       select e;
            result.ToList().ForEach(e => Console.WriteLine($"Name={e.Name},
Brand={e.Brand}"));
            Console.ReadLine();
        }
    }
    }
```

Output:

```
E:\NBHTraining\C# Training\DAY 8 Assignments\product id name salary using 4 loops\product id nar
*********************************************************************
name=fan,Brand=USHA
name=watch,Brand=Fastrack
*********************************************************************
Name=fan,BrandUSHA
Name=watch,BrandFastrack
*************************************************************
Name=fan,Brand=USHA
Name=watch,Brand=Fastrack
*************************************************************
Name=fan, Brand=USHA
Name=watch, Brand=Fastrack
```

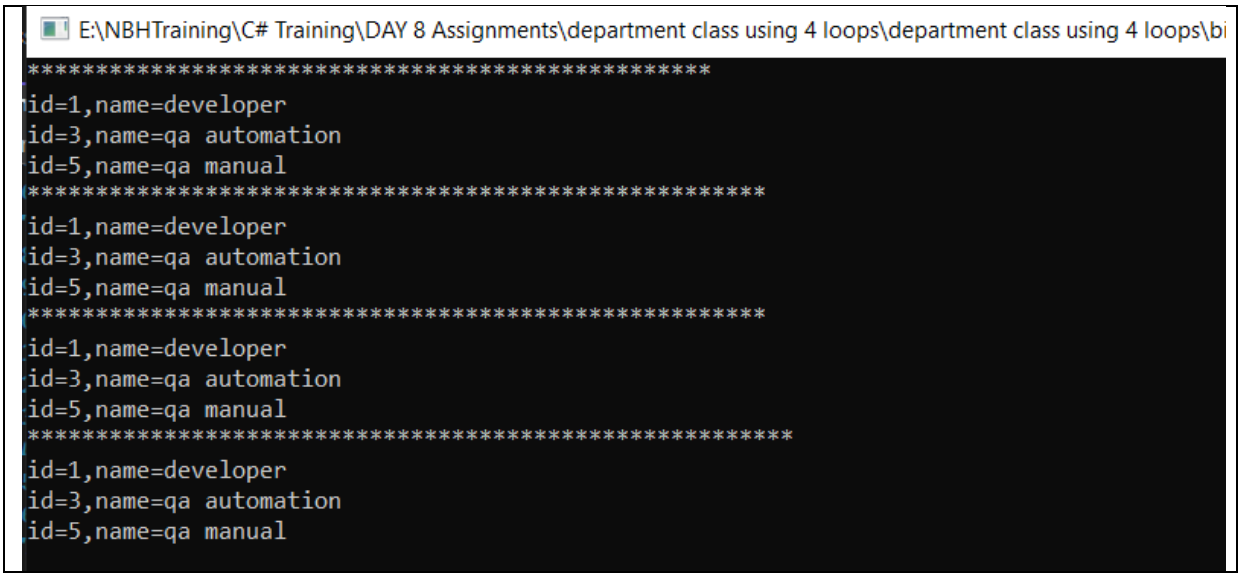
4.Create a Department class and add variables id,name,empcount. write code to print id,name of departments whose empcount is greater than 50 using for ,foreach, lambda ,linq query.

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace department_class_using_4_loops
{
    class Department
    {
        public int id;
        public string name;
        public int empcount;
```

```
    }
        internal class Program
    {
        static void Main(string[] args)
        {
            list<Department> department = new list<Department>()
            {
                new Department(){ id = 1,  name  ="developer",empcount=51},
                new Department(){ id = 2, name ="scrum master",empcount =30},
                new Department(){ id = 3, name ="qa automation", empcount =100},
                new Department(){ id = 4, name ="sql developer", empcount=29},
                new Department(){ id = 5, name ="qa manual",empcount=(63)}
            };
            Console.WriteLine("*************************************************");
            //empcount >50 using FORLOOP

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

            {
                if (department[i].empcount > 50)

Console.WriteLine($"id={department[i].id},name={department[i].name}");
            }

Console.WriteLine("*************************************************");
            // empcount >50 using FOREACH LOOP
            foreach (var e in department)
            {
                if (e.empcount > 50)
                    Console.WriteLine($"id={e.id},name={e.name}");
            }

Console.WriteLine("*************************************************");
            // empcount >50 using LAMBDA EXPRESSION
            department.ToList().Where(e => e.empcount > 50).ToList().ForEach(e =>
Console.WriteLine($"id={e.id},name={e.name}"));

Console.WriteLine("*************************************************");
            // empcount >50 using LINQ query
            var result = from e in department
                         where e.empcount > 50
                         select e;
            result.ToList().ForEach(e =>
Console.WriteLine($"id={e.id},name={e.name}"));




            Console.ReadLine();
        }
    }
}
```

Output:



5.Create your own class and variables and initialize with some values for, foreach, lambda, linq query

Code:

```
using System;
using System.Collections.Generic;
using  System.Linq;
using  System.Text;
using System.Threading.Tasks;

namespace own_program_using_4_loops
{
    class Student
    {
        public string Name;
        public int Rollnum;
        public string Section;
        public int Marks;
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            List<Student> student = new list<Student>
            {
              new  Student  {  Name  =  "john",Rollnum  =100,  Section  ="A",Marks=496},
              new Student { Name = "jeevitha",Rollnum =101, Section ="A",Marks=400},
              new Student { Name = "pavana",Rollnum =102, Section ="A",Marks=395},
              new Student { Name = "johney",Rollnum =103, Section ="A",Marks=456},
              new Student { Name = "mohith",Rollnum =104, Section ="A",Marks=302},
              new Student { Name = "mounika",Rollnum =105, Section ="A",Marks=399},
              new Student { Name = "raju",Rollnum =106, Section ="A",Marks=300},
            };

Console.WriteLine("**********************************************************");
            //print marks >=400 using FORLOOP
            for (int i = 0; i < student.Count; i++)
```

```
            {
                if (student[i].Marks >=  400)

Console.WriteLine($"name={student[i].Name},rollnum={student[i].Rollnum}");
            }

Console.WriteLine("*****************************************************************")
;
            //print marks >=400 using FOREACH LOOP
            foreach (var e in student)
            {
                if (e.Marks >=400)
                    Console.WriteLine($"name={e.Name},rollnum={e.Rollnum}");
            }

Console.WriteLine("*****************************************************************")
;
            //print marks >=400 using LAMBDA EXPRESSION
            student.ToList().Where(e => e.Marks >=400).ToList().ForEach(e =>
Console.WriteLine($"name={e.Name},rollnum={e.Rollnum}"));

Console.WriteLine("*****************************************************************")
;
            //print marks >=400 using LINQ QUARY
            var result = from e in student
                         where e.Marks >=400
                         select e;
            result.ToList().ForEach(e =>
Console.WriteLine($"name={e.Name},rollnum={e.Rollnum}"));
             Console.ReadLine();
        }
    }
}
```

Output:

```
E:\NBHTraining\C# Training\DAY 8 Assignments\own program using 4 loops\own program using 4 loops\bir
************************************************************
name=john,rollnum=100
name=jeevitha,rollnum=101
name=johney,rollnum=103
*****************************************************************
name=john,rollnum=100
name=jeevitha,rollnum=101
name=johney,rollnum=103
*****************************************************************
name=john,rollnum=100
name=jeevitha,rollnum=101
name=johney,rollnum=103
*****************************************************************
name=john,rollnum=100
name=jeevitha,rollnum=101
name=johney,rollnum=103
```