C# ASSESSMENT **VASANTHABALAN M**

Main Program To Excute All Questions

```
using ExecutionAll;

using System;

using System.Diagnostics;


namespace ExecutionAll // Note: actual namespace depends on the project name.

{

    internal class Program

    {

        static void Main(string[] args)

        {

            /*

            Console.WriteLine("Enter the chicken count:");

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

            Console.WriteLine("Enter the cows count:");

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

            Console.WriteLine("Enter the Pigs count:");

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

            BaseFarmer baseFarmer =new BaseFarmer(chicken,cows,pigs);

            Console.WriteLine("The total number of legs in their farmlands is:"+ baseFarmer.LegsCount());*/



            /*2.

            Console.WriteLine("Enter the Matches wins:");

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

            Console.WriteLine("Enter the Draws wins:");

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

            Console.WriteLine("Enter the Losses wins:");
```

```
int losses = Convert.ToInt32(Console.ReadLine());
if(wins >=0 && drwas>=0 && losses>=0)
{
   Football football = new Football(wins, drwas, losses);
   Console.WriteLine("The number of points scored by football team is:"+ football.scores());
}
else
{
   Console.WriteLine("Enter numbers must be greater than or equal to zero");
}*/

//3.
//Jadded2D jadded2D = new Jadded2D();
//jadded2D.display();
//jadded2D.disp();



//LambdaLE lambdaLE = new LambdaLE();
//lambdaLE.evennums();



/*Console.WriteLine("Enter the prob:");
double probs = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("Enter the Prize:");
int przе = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter the Pay:");
int pays = Convert.ToInt32(Console.ReadLine());

ProfitableGaming profitable = new ProfitableGaming(probs,prze,pays);
```

```
bool res = profitable.calc();
Console.WriteLine(res);*/

//4
/*Console.WriteLine("Enter the Stackboxes:");
int count = Convert.ToInt32(Console.ReadLine());
Cubes cubes = new Cubes(count);
Console.WriteLine("The answer is:" +cubes.squares());*/

//5
/*Console.WriteLine("Enter the age");
int age = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the breaktime");
bool breaktime = Convert.ToBoolean(Console.ReadLine());

BarTender bartender = new BarTender();
Console.WriteLine(bartender.ShouldServeDrinks(age, breaktime));*/

//6
/*Console.WriteLine("Enter the cup count");
int cup = Convert.ToInt32(Console.ReadLine());
Cofee cofee = new Cofee();
Console.WriteLine("Total Number of cups : " + cofee.TotalCups(cup));*/

//7

//8
Console.WriteLine("Enter the number of switches");
```

```
            int num = Convert.ToInt32(Console.ReadLine());
            Switchs switchs = new Switchs();
            Console.WriteLine(switchs.PosCom(num));

        }

    }
}
}
```

1)

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

namespace ExecutionAll
{
    internal class BaseFarmer
    {
        private int chicken;
        private int cows;
        private int pigs;

        public BaseFarmer(int chicken, int cows, int pigs)
        {
            this.Chicken = chicken;
            this.Cows = cows;
            this.Pigs = pigs;
        }

        public int Chicken { get => chicken; set => chicken = value; }
        public int Cows { get => cows; set => cows = value; }
        public int Pigs { get => pigs; set => pigs = value; }

        public int LegsCount()
        {
            int result = (this.Chicken * 2 + this.Cows * 4 + this.Pigs * 4);
            return result;
        }
    }
}
```

2)

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

namespace ExecutionAll
{
    internal class Football
    {
        private int wins;
        private int draws;
        private int losses;
        public Football(int wins, int draws, int losses)
        {
            this.Wins = wins;
            this.Draws = draws;
            this.Losses = losses;
        }

        public int Wins { get => wins; set => wins = value; }
        public int Draws { get => draws; set => draws = value; }
        public int Losses { get => losses; set => losses = value; }

        public int scores()
        {
            int points = this.Wins * 3 +this.Draws * 1 +this.Losses * 0;
            return points;
        }
    }
}
```

3)

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

namespace ExecutionAll
{
    internal class ProfitableGaming
    {
        private double prob;
        private int prize;
        private int pay;

        public ProfitableGaming(double prob, int prize, int pay)
        {
            this.Prob = prob;
            this.Prize = prize;
            this.Pay = pay;
        }

        public double Prob { get => prob; set => prob = value; }
        public int Prize { get => prize; set => prize = value; }
        public int Pay { get => pay; set => pay = value; }

        public bool calc()
        {
            if (this.Prob * this.Prize > this.Pay)
            {
                return true;
            }
            else
            {
                return false;
            }

        }
    }
}
```

4)

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

namespace ExecutionAll
{
    internal class Cubes
    {
        private int cube;

        public Cubes(int cube)
        {
            this.Cube = cube;
        }

        public int Cube { get => cube; set => cube = value; }

        public int squares()
        {
            int res = cube * cube;
            return res ;
        }
    }
}
```

5)

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

namespace ExecutionAll
{
    internal class BarTender
    {
        private int age;
        private bool breaktime;

        public int Age { get => age; set => age = value; }
        public bool Breaktime { get => breaktime; set => breaktime = value; }

        public bool ShouldServeDrinks(int age, bool breaktime)
        {
            if ((age >= 18) && (breaktime == false))
            {
                return true;
            }
            else
            {
                return false;
            }
        }
    }
}
```

6)

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

namespace ExecutionAll
{
    internal class Cofee
    {


        private int cups;

        public int Cups { get => cups; set => cups = value; }

        public int TotalCups(int cups)
        {
            int total_cups = (cups / 6) + cups;
            return total_cups;
        }


    }
}
```

7)

8)

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

namespace ExecutionAll
{
    internal class Switchs
    {
        public int PosCom(int num)
        {
            return (int)Math.Pow(2, num);
        }

    }
}
```