

1. A farmer is asking you to tell him how many legs can be counted among all his animals. The farmer breeds three species:

chickens = 2 legs  
cows = 4 legs  
pigs = 4 legs

The farmer has counted his animals and he gives you a subtotal for each species. You have to implement a function that returns the total number of legs of all the animals.

```
using Basic_Program;  
using System.Net.Http.Headers;
```

```
class Demo1  
{  
    public static void Main(string[] args)  
    {  
        Console.WriteLine("Enter the chicken count");  
        int chicken = Convert.ToInt32(Console.ReadLine());  
        Console.WriteLine("Enter the cow count");  
        int cow = Convert.ToInt32(Console.ReadLine());  
        Console.WriteLine("Enter the pig count");  
        int pig = Convert.ToInt32(Console.ReadLine());  
        Animal1 animal1 = new Animal1(chicken, cow, pig);  
        Console.WriteLine(animal1.count_legs());  
    }  
}
```

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

```
namespace Basic_Program  
{  
    internal class Animal1  
    {  
        private int chickens;  
        private int cows;  
        private int pigs;
```

```

public Animal1(int chickens, int cows, int pigs)
{
    this.Chickens = chickens;
    this.Cows = cows;
    this.Pigs = pigs;
}

public int Chickens { get => chickens; set => chickens = value; }
public int Cows { get => cows; set => cows = value; }
public int Pigs { get => pigs; set => pigs = value; }

public int count_legs()
{
    int result = (this.Chickens * 2) + (this.Cows * 4) + (this.Pigs * 4);
    return result;
}
}

```

```

Enter the chicken count
2
Enter the cow count
3
Enter the pig count
5
36

```

2. Create a function that takes the number of wins, draws and losses and calculates the number of

points a football team has obtained so far.

wins get 3 points draws get 1 point losses get 0 points

```

using Basic_Program;
using System.Net.Http.Headers;

```

```

class Demo1
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Enter the number of wins");
        int wins = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter the number of draws");
    }
}

```

```

        int draws = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter the losses");
        int losses = Convert.ToInt32(Console.ReadLine());
        Football football = new Football();
        Console.WriteLine(football.FootballPoints(wins, draws, losses));
    }
}

```

```

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

```

```

namespace Basic_Program
{
    internal class Football
    {
        private int wins;
        private int draws;
        private int 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 FootballPoints(int wins, int draws, int losses)
        {
            int result = (wins * 3) + (draws * 1) + (losses * 0);
            return result;
        }
    }
}

```

```

Enter the number of wins
0
Enter the number of draws
0
Enter the losses
1
0

```

3. Create a function that takes three arguments prob, prize, pay and returns true if prob \* prize > pay otherwise return false.

```
using Basic_Program;
using System.Net.Http.Headers;

class Demo1
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Enter the prob");
        double prob = Convert.ToDouble(Console.ReadLine());
        Console.WriteLine("Enter the prize");
        int prize = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter the pay");
        int pay = Convert.ToInt32(Console.ReadLine());

        Gamble gamble = new Gamble();
        Console.WriteLine(gamble.ProfitableGamble(prob, prize, pay));
    }
}
```

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

namespace Basic_Program
{
    internal class Gamble
    {
        private double prob;
        private int prize;
        private int 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 ProfitableGamble(double prob, int prize, int pay)
        {
            if (prob * prize > pay)
            {

```

```

        return true;
    }
    else
    {
        return false;
    }
}
}
}

```

```

Enter the prob
0.2
Enter the prize
50
Enter the pay
9
True

```

4. Write a function that takes a number n and returns the number of stacked boxes in a model n levels high, visible and invisible.

```

using Basic_Program;
using System.Net.Http.Headers;

class Demo1
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Enter the number");
        int number = Convert.ToInt32(Console.ReadLine());

        Box box = new Box();
        Console.WriteLine("Number os stacked boxes : " + box.StackBoxes(number));
    }
}

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

```

```

namespace Basic_Program
{
    internal class Box
    {
        private int number;

        public int Number { get => number; set => number = value; }

        public int StackBoxes(int number)
        {
            return number * number;
        }
    }
}

```

```

Enter the number
2
Number os stacked boxes : 4

```

5. A bartender is writing a simple program to determine whether he should serve drinks to someone. He only serves drinks to people 18 and older and when he's not on break. Given the person's age, and whether break time is in session, create a function which returns whether he should serve drinks.

```

using Basic_Program;
using System.Net.Http.Headers;

class Demo1
{
    public static void Main(string[] args)
    {
        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));
    }
}

```

```

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

namespace Basic_Program
{
    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;
            }
        }
    }
}

```

```

Enter the age
30
Enter the breaktime
true
False

```

6. For each of the 6 coffee cups I buy, I get a 7th cup free. In total, I get 7 cups. Create a function that takes n cups bought and return the total number of cups I would get.

```

using Basic_Program;
using System.Net.Http.Headers;

```

```

class Demo1
{

```

```

public static void Main(string[] args)
{
    Console.WriteLine("Enter the cup count");
    int cup = Convert.ToInt32(Console.ReadLine());
    Coffee coffee = new Coffee();
    Console.WriteLine("Total Number of cups : " + coffee.TotalCups(cup));
}
}

```

```

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

```

```

namespace Basic_Program
{
    internal class Coffee
    {
        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;
        }
    }
}

```

```

Enter the cup count
213
Total Number of cups : 248

```

7. Create a function that adds a string ending to each member in an array.

```

using Basic_Program;
using System.Net.Http.Headers;

```

```

class Demo1
{

```



```

public static void Main(string[] args)
{
    Ending ending = new Ending();
    string[] temp = { "clever", "meek", "hurried", "nice"};
    ending.AddEnding(temp, "ly");
}

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

namespace Basic_Program
{
    internal class Ending
    {
        public void AddEnding(string[] words, string addon)
        {
            for(int i=0;i<words.Length;i++)
            {
                words[i] = words[i] + addon;
            }

            for (int i = 0; i < words.Length; i++)
            {
                Console.Write(words[i] + " ");
            }
        }
    }
}

```

```
cleverly meekly hurriedly nicely
```

8. Create a function that returns how many possible arrangements can come from a certain number

of switches (on / off). In other words, for a given number of switches, how many different patterns of on and off can we have?

```
using Basic_Program;
```

```
using System.Net.Http.Headers;
```

```
class Demo1
```

```
{  
    public static void Main(string[] args)  
    {  
        Console.WriteLine("Enter the number of switches");  
        int num = Convert.ToInt32(Console.ReadLine());  
        Switch switches = new Switch();  
        Console.WriteLine(switches.PosCom(num));  
    }  
}
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Basic_Program
```

```
{  
    internal class Switch  
    {  
        public int PosCom(int num)  
        {  
            return (int)Math.Pow(2, num);  
        }  
    }  
}
```

```
Enter the number of switches  
10  
1024
```