***Section 1***

# Question: 1

Write a C# method which accepts two integers as parameters (m, n) both of which must be less than 20 and print the grade-school multiplication table **m x n.**

**Expected Output for 2 x 12:**

1 2 3 4 5 6 7 8 9 10 11 12  
2 4 6 8 10 12 14 16 18 20 22 24

public static void MultiplicationTable(int m, int n)

{

if (m >= 20 || n >= 20)

{

Console.WriteLine("Both m and n values should be less than 20.");

}

else

{

string result = string.Empty;

for(int i = 1; i <= n; i++)

{

Console.Write(i + " ");

result += (m \* i) + " ";

}

Console.WriteLine();

Console.WriteLine(result);

}

}

# Question: 2

Write a C# method which accepts a number as parameter and reverse it without converting it to string and don’t use array.

public static int ReverseInt(int num)

{

int rNum = 0;

while (num > 0)

{

rNum = (rNum \* 10) + (num % 10);

num = num / 10;

}

return rNum;

}

# Question: 3

Write a C# method which accepts a string as parameter and prints the duplicate characters along with the number of times it had been duplicated.

public static void PrintDuplicates(string input)

{

var result = input.GroupBy(c => c).Where(c => c.Count() > 1).Select(c => new { charc = c.Key, countc = c.Count() });

foreach(var c in result)

{

Console.WriteLine(c.charc +" "+ c.countc);

}

}

# Question: 4

Write a C# method which accepts a string as parameter check whether it contains only digits without using regular expression.

public static bool HasOnlyDigits(string input)

{

int inputAsNumber = 0;

return int.TryParse(input, out inputAsNumber);

}

***Section 2***

# Question: 1

Let us assume for an upcoming project, one of the requirements is to mimic “Microsoft Windows File System”. And it is your task to create the data models needed. Come up with a POCO class design to represent the file system in C#.

**Ans:** I have created below POCO classes for sample and We have System.IO library in c#

which is a wrapper of Windows file system, this library has classes like File, FileInfo, Directory, DirectoryInfo and Path and other classes with all properties and methods to read, write, Update and Delete files, directories in windows.

public class DirectoryInformation

{

public DateTime CreationTime { get; set; }

public bool Exists { get; set; }

public string Extension { get; set; }

public string FullName { get; set; }

public DateTime LastAccessTime { get; set; }

public DateTime LastWriteTime { get; set; }

public string Name { get; set; }

}

public class FileInformation

{

public DateTime CreationTime { get; set; }

public DirectoryInformation Directory { get; set; }

public string DirectoryName { get; set; }

public bool Exists { get; set; }

public string Extension { get; set; }

public string FullName { get; set; }

public DateTime LastAccessTime { get; set; }

public DateTime LastWriteTime { get; set; }

public long Length { get; set; }

public string Name { get; set; }

}

# Question: 2

In C# create the POCO classes needed to Model the Animal Kingdom, for use in a Virtual Zoo Program. Make sure you have 10 different Animals. Also ensure the Models created should support further enhancements like, New Animals & More Food Options.

public interface IBase

{

int Weight { get; set; }

int Height { get; set; }

int Length { get; set; }

string Sound { get; set; }

int LifeSpan { get; set; }

}

public interface IWalkable

{

int Legs { get; set; }

int RunningSpeed { get; set; }

int MaxDistance { get; set; }

}

public interface ISwimable

{

int SwimSpeed { get; set; }

}

public interface IFlyable

{

int Wings { get; set; }

int MaxHeight { get; set; }

int MaxDistance { get; set; }

}

public interface IFood

{

string FoodType { get; set; }

string ConsumptionType { get; set; }

int RequiredFood { get; set; }

}

public class Animal : IBase, IWalkable, ISwimable, IFood

{

//code for other properties and metohds

int IBase.Weight { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

int IBase.Height { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

int IBase.Length { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

string IBase.Sound { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

int IBase.LifeSpan { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

int IWalkable.Legs { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

int IWalkable.RunningSpeed { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

int IWalkable.MaxDistance { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

int ISwimable.SwimSpeed { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

string IFood.FoodType { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

string IFood.ConsumptionType { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

int IFood.RequiredFood { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

}

public class Bird : IBase, IFlyable, ISwimable, IFood

{

//Code for other properties and methods

//Code for interface implimentation

}

public class Mamal : IBase, IWalkable, IFlyable, ISwimable, IFood

{

//Code for other properties and methods

//Code for interface implimentation

}

# Question: 3

In C# create the POCO classes needed to support Restaurant Reservation System.

public class Table

{

public int TableNumber { get; set; }

public int Seats { get; set; }

}

public class Slots

{

public DateTime FromTime { get; set; }

public DateTime ToTime { get; set; }

}

public class Customer

{

public string Name { get; set; }

public string Phone { get; set; }

public string Email { get; set; }

}

public class Reservation

{

public Customer Customer { get; set; }

public Slots Slot { get; set; }

public Table Table { get; set; }

public string ReservationType { get; set; }

}