

Write C# code for the following problems.

Use OOP concepts wherever applicable. EXCEPTIONS to be provided for CONSTRAINTS given  
(Userdefined / System defined).

All Pogram Main Functions:

`using System;`

`namespace Assessments_3 // Note: actual namespace depends on the project name.`  
`{`

`public class Program`  
 `{`

`public static void Main(string[] args)`  
 `{`

`//1.`  
 `/*try`  
 `{`  
 `FileConcept concept = new FileConcept();`  
 `// concept.createfile();`  
 `concept.Writefile();`  
 `concept.Readfile();`  
 `concept.Written();`  
 `}`  
 `catch`  
 `{`  
 `Console.WriteLine("the file doesnot implement ");`  
 `}*/`

`//2.`  
 `/*string str = Console.ReadLine();`  
 `StngDupRmve stngDupRmve = new StngDupRmve(str);`  
 `stngDupRmve.rmvdup();*/`

`//3.`  
 `/*string str = Console.ReadLine();`  
 `CntStr cntStr = new CntStr(str);`  
 `cntStr.cnt();*/`

`//4.`  
 `/*string str1 = Console.ReadLine();`  
 `string str2 = Console.ReadLine();`  
 `SubStr subStr = new SubStr(str1, str2);`  
 `subStr.subcount();*/`

`//5.`  
 `/*int n = int.Parse(Console.ReadLine());`  
 `BallColor ballColor = new BallColor(n);`  
 `ballColor.res();*/`

`//6.`  
 `/*string[] inputs = Console.ReadLine().Split(' ');`  
 `long n = long.Parse(inputs[0]);`  
 `int k = int.Parse(inputs[1]);`  
 `Prison prison = new Prison(n, k);`

```

        prison.reslt();*/

        //7
        int n = int.Parse(Console.ReadLine());
        int m = int.Parse(Console.ReadLine());
        Buses buses = new Buses(n,m);
        buses.fetchs();
    }
}
}

```

1. Write a program to

a. create and write some line of text into a file which does not contain a given set of phrase in a line.

Constraints:

Check for file existence

Check if the file is operable

Input the string to ignore the line : fox

Input number of lines to write in the file : 2

Input 2 strings below :

Input line 1 : the quick brown fox jumps

Input line 2 : over the lazy dog.

Expected Output :

The line have ignored which contain the string 'fox'.

The content of the file is :

-----

over the lazy dog.

b. Read a specific line from a file

PROGRAM:

FileConcept.cs:

```

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

```

```

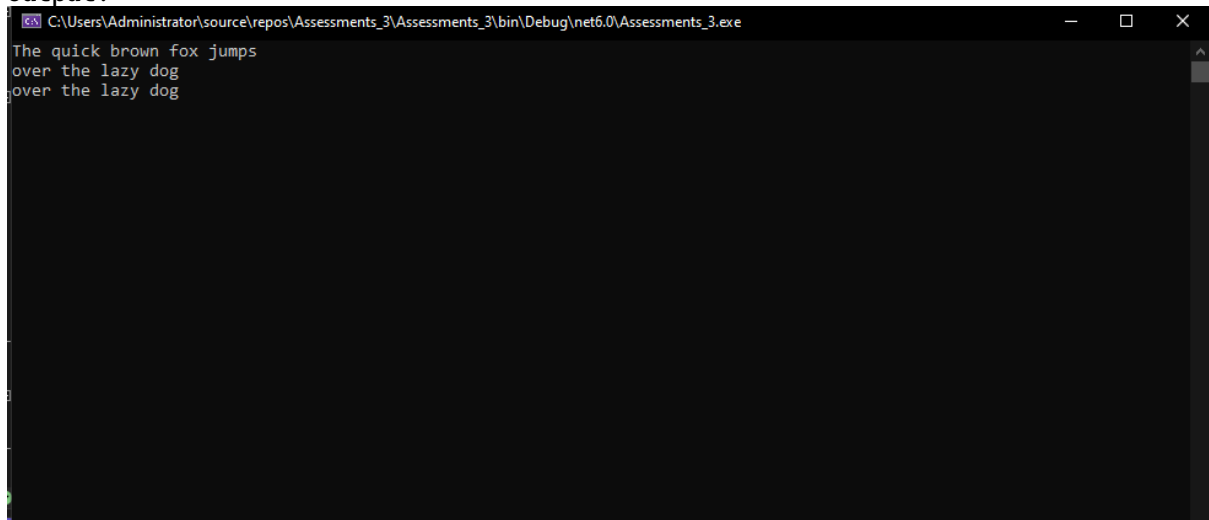
namespace Assessments_3
{
    internal class FileConcept
    {
        public void createfile()
        {
            string path = "D:\\bala\\kanini\\c#\\FileQuestion\\file1.txt";
            FileStream fileStream = File.Create(path);
            Console.WriteLine("this file is now created");
        }

        public void Writefile()
        {
            string path = "D:\\bala\\kanini\\c#\\FileQuestion\\file1.txt";
            FileStream fileStream = new FileStream
(path, FileMode.Create, FileAccess.Write);
            StreamWriter fileWriter = new StreamWriter(fileStream);
            string str1 = Console.ReadLine();
            string str2 = Console.ReadLine();
            fileWriter.WriteLine(str1);
            fileWriter.WriteLine(str2);
            fileWriter.Flush();
            fileWriter.Close();
            fileStream.Close();
        }
        public void Readfile()
        {
            string path = "D:\\bala\\kanini\\c#\\FileQuestion\\file1.txt";
            FileStream fileStream = new FileStream(path, FileMode.Open,
FileAccess.Read);
            StreamReader reader = new StreamReader(fileStream);
            reader.BaseStream.Seek(0, SeekOrigin.Begin);
            string str = reader.ReadLine();
            string str1 = "fox";
            while (str != null)
            {
                if (!str.Contains(str1))
                {
                    Console.WriteLine(str);
                }
                str = reader.ReadLine();
                FileConcept concept = new FileConcept();
                //concept.Written(str);
            }
            reader.Close();
            fileStream.Close();
        }
        public void Written()
        {
            string path = "D:\\bala\\kanini\\c#\\FileQuestion\\file1.txt";
            FileStream fileStream = new FileStream(path, FileMode.Create,
FileAccess.Write);
            StreamWriter fileWriter = new StreamWriter(fileStream);
            string str = Console.ReadLine();
            fileWriter.WriteLine(str);
            fileWriter.Flush();
            fileWriter.Close();
            fileStream.Close();
        }
    }
}

```

```
}
```

Output:



```
C:\Users\Administrator\source\repos\Assessments_3\Assessments_3\bin\Debug\net6.0\Assessments_3.exe
The quick brown fox jumps
over the lazy dog
over the lazy dog
```

2. Given a string,  $str = s_1s_2\dots s_n$ , consisting of lowercase English characters (a-z), remove all of the characters that occurred previously in the string.

Input Format

A single line of input containing a string  $str$  of length  $n$ .

Constraints

Output Format

Print the string after removing all the characters that occurred previously.

Sample Input

Accabb

Sample Output

acb

StngdupRmve.cs:

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

namespace Assessments_3
{
    internal class StngDupRmve
    {
        private string str;
        private string str2 = string.Empty;
    }
}
```

```

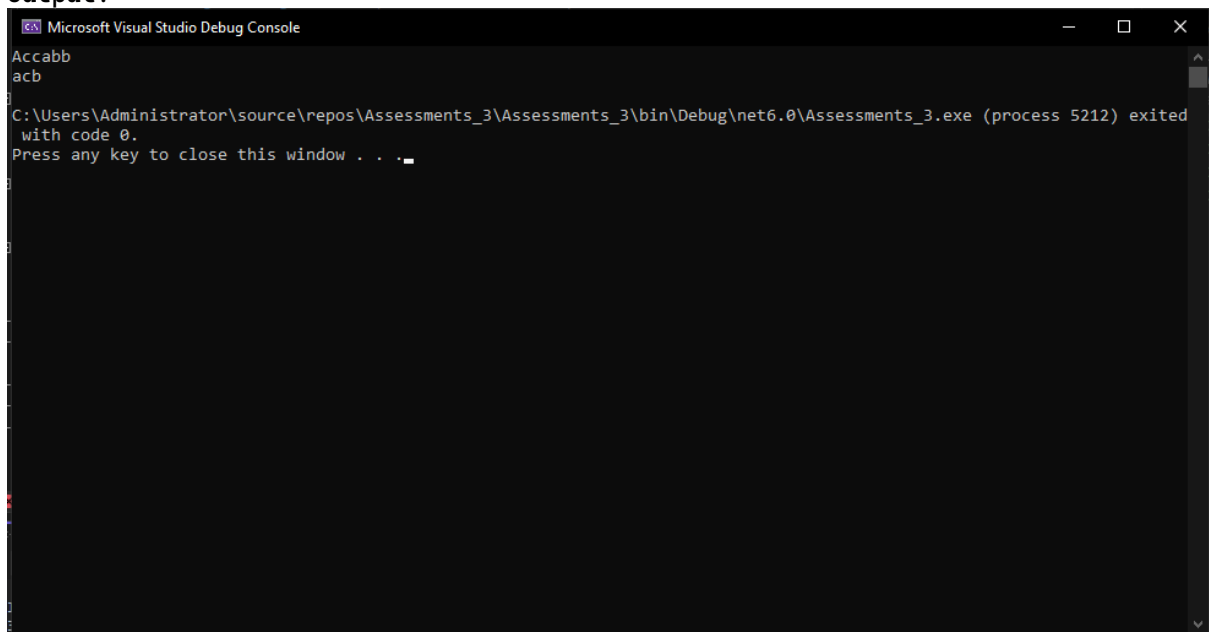
public StngDupRmve(string str)
{
    this.Str = str;
}

public string Str { get => str; set => str = value; }
public string Str2 { get => str2; set => str2 = value; }

public void rmvdup()
{
    //Console.WriteLine(this.Str);
    for(int i=0;i<this.Str.Length; i++)
    {
        if (!this.Str2.Contains(this.Str[i]))
        {
            this.Str2 += this.Str[i];
        }
    }
    Console.WriteLine(this.Str2);
}
}
}

```

Output:



```

Microsoft Visual Studio Debug Console
Accabb
acb
C:\Users\Administrator\source\repos\Assessments_3\Assessments_3\bin\Debug\net6.0\Assessments_3.exe (process 5212) exited
with code 0.
Press any key to close this window . . .

```

3. Joseph and Jane are making a contest for apes. During the process, they have to communicate frequently with

each other. Since they are not completely human, they cannot speak properly. They have to transfer messages

using postcards of small sizes. To save space on the small postcards, they devise a string compression algorithm

## Input

The only line of input contains a string, msg.

## Output

Print the string msg as a compressed message.

## Constraints

- msg consists of lowercase English characters () only.

## Sample Input

Abcaaabbb

## Sample Output

abca3b3

CntStr.cs:

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

namespace Assessments_3
{
    internal class CntStr
    {
        private string str;
        public CntStr(string str)
        {
            this.Str = str;
        }

        public string Str { get => str; set => str = value; }

        public void cnt()
        {
            string str2 = "";
            int count = 1;
            for (int i = 1; i < this.Str.Length; i++)
            {
                if (this.Str[i] == this.Str[i - 1])
                {
                    count++;
                }
                else
                {
                    str2 += this.Str[i - 1] ;
                    count = 1;
                }
            }
            str2 += this.Str[this.Str.Length - 1] + count.ToString();

            Console.WriteLine(str2.ToLower());
        }
    }
}
```

```

    }
}

```

Output:

```

Microsoft Visual Studio Debug Console
abcbbb
abcb3
C:\Users\Administrator\source\repos\Assessments_3\Assessments_3\bin\Debug\net6.0\Assessments_3.exe (process 20732) exited with code 0.
Press any key to close this window . . .

```

4. You are in charge of data transfer between two Data Centers. Each set of data is represented by a pair of

strings. Over a period of time you have observed a trend: most of the times both strings share some prefix.

You want to utilize this observation to design a data compression algorithm which will be used to reduce

amount of data to be transferred.

SubStr.cs:

```

int length;
int index;
public void process(string s1, string s2)
{
    int prefixLen = 0;
    for (int i = 0; i < Math.Min(s1.Length, s2.Length); i++)
    {
        if (s1[i] == s2[i])
        {
            prefixLen++;
        }
        else
        {
            break;
        }
    }

    string compressedS1 = s1.Substring(prefixLen);
    string compressedS2 = s2.Substring(prefixLen);

    Console.WriteLine($"{prefixLen} {s1.Substring(0, prefixLen)}");
    Console.WriteLine($"{compressedS1.Length} {compressedS1}");
    Console.WriteLine($"{compressedS2.Length} {compressedS2}");
}

```

Sample Input

abcdefpr

abcpqr

Sample Output

3 abc

5 defpr

3 pqr

Sample Input

kitkat

kit

Sample Output

3 kit

3 kat

0

Sample Input

puppy

puppy

Sample Output

5 puppy

0

0

Output:



```
Microsoft Visual Studio Debug Console
kitkat
kit
3 kit
3 kat
0

C:\Users\Administrator\source\repos\Assessments_3\Assessments_3\bin\Debug\net6.0\Assessments_3.exe (process 10384) exited with code 0.
Press any key to close this window . . .
```

5. You are given a sequence of balls in 4 colors: red, green, yellow and blue. The sequence is full of colors if

and only if all of the following conditions are true:

- There are as many red balls as green balls.
- There are as many yellow balls as blue balls.
- Difference between the number of red balls and green balls in every prefix of the sequence is at most 1.
- Difference between the number of yellow balls and blue balls in every prefix of the sequence is at most 1.

Your task is to write a program, which for a given sequence prints True if it is full of colors, otherwise it

prints False.

Sample Input

4

RGGR

RYBG

RYRB

YGYGRBRB

Sample Output

True

True

False

False

BallColor.cs:

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

namespace Assessments_3
{
    internal class BallColor
    {
        int n;
        public BallColor(int n)
        {
            this.N = n;
        }

        public int N { get => n; set => n = value; }

        static bool IsFullOfColors(string sequence)
        {
            int redCount = 0;
            int greenCount = 0;
            int yellowCount = 0;
            int blueCount = 0;

            for (int i = 0; i < sequence.Length; i++)
            {
                switch (sequence[i])
                {
                    case 'R':
                        redCount++;
                        break;
                    case 'G':
                        greenCount++;
                        break;
                    case 'Y':
                        yellowCount++;
                        break;
                    case 'B':
                        blueCount++;
                        break;
                }

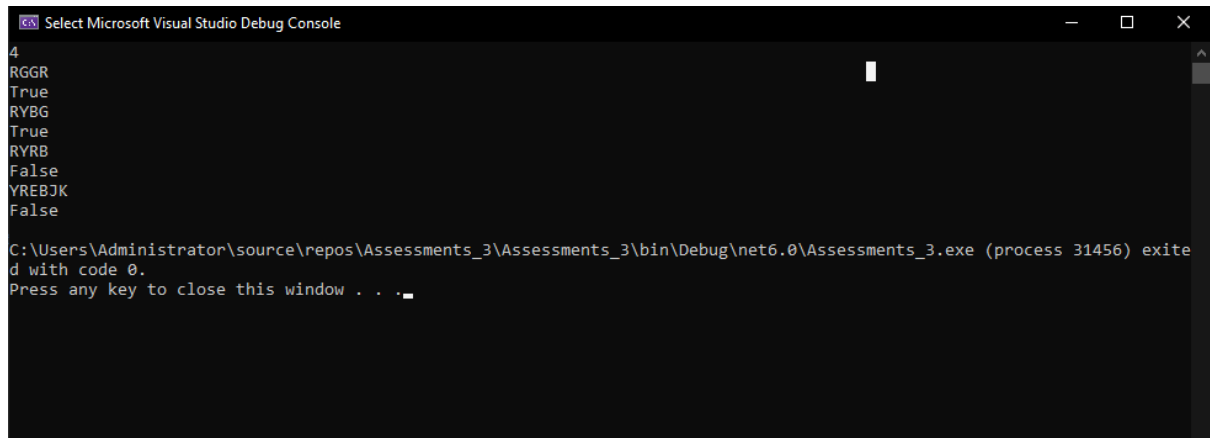
                if (Math.Abs(redCount - greenCount) > 1 || Math.Abs(yellowCount -
blueCount) > 1)
                {
                    return false;
                }
            }

            return redCount == greenCount && yellowCount == blueCount;
        }

        public void res()
        {
            for (int i = 0; i < N; i++)
            {
                string sequence = Console.ReadLine();
                Console.WriteLine(IsFullOfColors(sequence) ? "True" : "False");
            }
        }
    }
}
```

```
}  
}
```

Output:



```
Select Microsoft Visual Studio Debug Console  
4  
RGGR  
True  
RYBG  
True  
RYRB  
False  
YREBJK  
False  
  
C:\Users\Administrator\source\repos\Assessments_3\Assessments_3\bin\Debug\net6.0\Assessments_3.exe (process 31456) exited with code 0.  
Press any key to close this window . . .
```

6. We define super digit of an integer using the following rules:

- If has only digit, then its super digit is .
- Otherwise, the super digit of is equal to the super digit of the digit-sum of . Here, digit-sum of a number

is defined as the sum of its digits.

For example, super digit of will be calculated as:

$\text{super\_digit}(9875) = \text{super\_digit}(9+8+7+5)$

$= \text{super\_digit}(29)$

$= \text{super\_digit}(2+9)$

$= \text{super\_digit}(11)$

$= \text{super\_digit}(1+1)$

$= \text{super\_digit}(2)$

$= 2.$

Sample Input

148 3

Sample Output

3

Prison.cs:

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

namespace Assessments_3
{
    internal class Prison
    {
        private long n;
        private int k;

        public Prison(long n, int k)
        {
            this.N = n;
            this.K = k;
        }

        public long N { get => n; set => n = value; }
        public int K { get => k; set => k = value; }

        static int SuperDigit(long n)
        {
            if (n < 10)
            {
                return (int)n;
            }
            else
            {
                long digitSum = 0;
                while (n > 0)
                {
                    digitSum += n % 10;
                    n /= 10;
                }
                return SuperDigit(digitSum);
            }
        }

        public void reslt()
        {
            long digitSum = 0;
            while (N > 0)
            {
                digitSum += N % 10;
                N /= 10;
            }
            digitSum *= K;
            Console.WriteLine(SuperDigit(digitSum));
        }
    }
}
```

Output:

```
Microsoft Visual Studio Debug Console
148 3
3
C:\Users\Administrator\source\repos\Assessments_3\Assessments_3\bin\Debug\net6.0\Assessments_3.exe (process 31536) exited with code 0.
Press any key to close this window . . .
```

7. There are  $N$  inmates numbered between  $[1, N]$  in a prison. These inmates have superhuman strength because

they have drunk a special concoction made by Dr. Evil. They have to be transported by some buses to a new

facility. But they are bound by special chains which are made from strong carbon fibres. Each inmate is either

chained alone or is chained in a group along with one or more inmates. A group of inmates are those who are

directly or indirectly connected to each other. Only one group can be transported per bus.

There are buses which will charge fixed amount bucks for transferring inmates. Charges are directly proportional

to the capacity of bus. If a bus charge  $K$  bucks then it can carry upto  $K$

2

inmates at one time. Buses are available

for all positive integral cost ranging from  $[1, 2, 3, \dots]$ . A bus can be used multiple times, and each time it will

charge. Note that a bus can also transfer less number of inmates than it's capacity.

Find the minimal cost to transport all the inmates.

Input

The first line contains  $N$  representing the number of inmates. Second line contains another integer,  $M$ , number of

pairs of inmates who are handcuffed together. Then follows  $M$  lines. Each of these lines contains two integers,  $P$

$Q$ , which means inmate numbered  $P$  is handcuffed to inmate numbered  $Q$ .

Output

For the given arrangement, print the minimal cost which can be incurred while transferring inmates.

Constraints

$2 \leq N \leq 100000$

$1 \leq M \leq \min(N*(N-1)/2, 100000)$

$1 \leq P, Q \leq N$

$P \neq Q$

Sample Input

4

2

1 2

1 4

Sample Output

3

Explanation

Inmates #1, #2, #4 are connected to each other (1--2--4) so they lies in a single group. So a bus of cost 2 (with

capacity  $2*2 = 4$ ) is required to carry them. Inmate #3 is not handcuffed with anyother. So he can be transported in

a bus of cost 1 (with capacity  $1*2 = 1$ ).

Buses.cs:

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

namespace Assessments_3
{
    internal class Buses
    {
        static int[] parent;
        static int[] size;
        private int n;
        private int m;

        public Buses(int n, int m)
        {
            this.N = n;
        }
    }
}
```

```

        this.M = m;
    }

    public static int[] Parent { get => parent; set => parent = value; }
    public static int[] Size { get => size; set => size = value; }
    public int N { get => n; set => n = value; }
    public int M { get => m; set => m = value; }

    static int Find(int x)
    {
        if (Parent[x] == x)
        {
            return x;
        }
        return Parent[x] = Find(Parent[x]);
    }

    static void Union(int x, int y)
    {
        int rootX = Find(x);
        int rootY = Find(y);
        if (rootX != rootY)
        {
            if (Size[rootX] < Size[rootY])
            {
                int temp = rootX;
                rootX = rootY;
                rootY = temp;
            }
            Parent[rootY] = rootX;
            Size[rootX] += Size[rootY];
        }
    }

    public void fetchs()
    {
        parent = new int[n + 1];
        size = new int[n + 1];
        for (int i = 1; i <= n; i++)
        {
            parent[i] = i;
            size[i] = 1;
        }
        for (int i = 0; i < m; i++)
        {
            string[] line = Console.ReadLine().Split();
            int x = int.Parse(line[0]);
            int y = int.Parse(line[1]);
            Union(x, y);
        }
        int[] groupSize = new int[n + 1];
        for (int i = 1; i <= n; i++)
        {
            groupSize[Find(i)]++;
        }
        int cost = 0;
        int remaining = n;
        for (int i = 1; i <= n; i++)
        {
            if (groupSize[i] > 0)
            {
                int k = (int)Math.Ceiling(Math.Sqrt(groupSize[i]));
                cost += k;
                remaining -= k * k;
            }
        }
    }

```

```

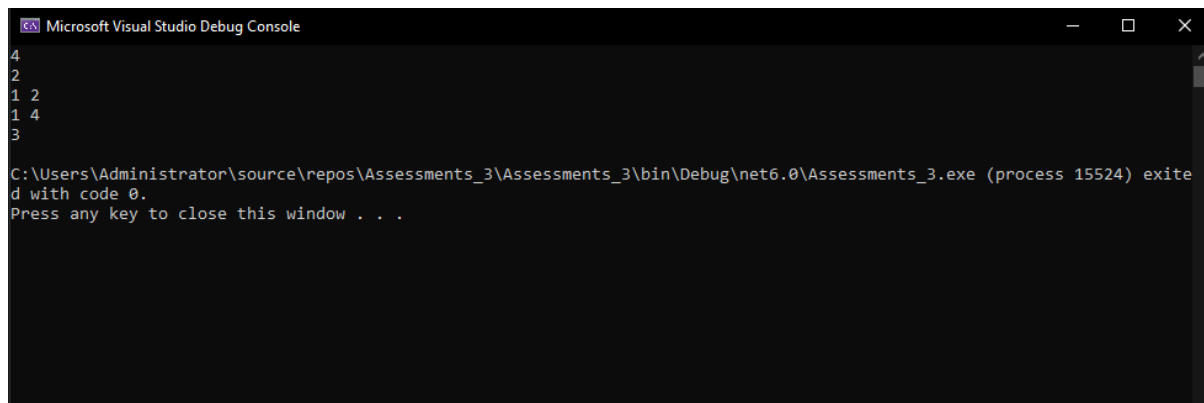
        }
    }
    if (remaining > 0)
    {
        cost += (int)Math.Ceiling((double)remaining /
Math.Sqrt(remaining));
    }
    Console.WriteLine(cost);
}

}

}

```

Output:



The screenshot shows the Microsoft Visual Studio Debug Console window. The title bar reads "Microsoft Visual Studio Debug Console". The console output displays the following lines:

```

4
2
1 2
1 4
3

```

Below the output, the console shows the process path and exit status:

```

C:\Users\Administrator\source\repos\Assessments_3\Assessments_3\bin\Debug\net6.0\Assessments_3.exe (process 15524) exited with code 0.
Press any key to close this window . . .

```