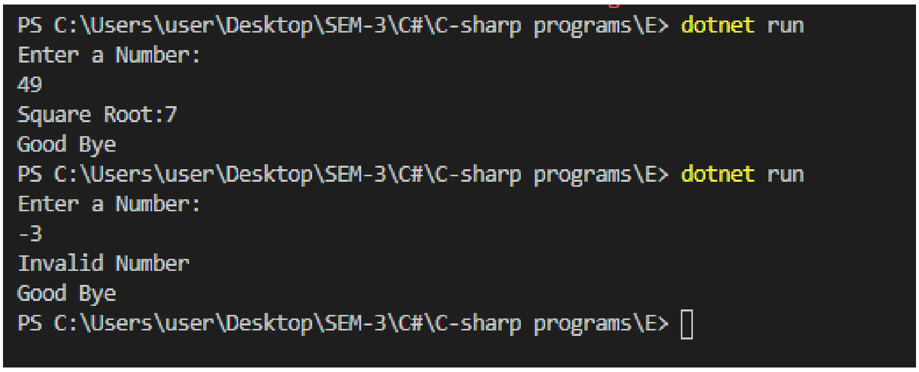
Lab Assignment #6

EP1: Implement a program that takes a positive integer from the console and prints the square root of this integer. If the input is negative or invalid print "Invalid Number" in the console. In all cases print "Good Bye".

```
using System;

namespace EP1
{
    class Program
    {
        class NegativeNumberNotAllowed: Exception
        {
        public NegativeNumberNotAllowed(string message): base(message){}
        }
        static void Main(string[] args)
        {
            Console.WriteLine("Enter a Number:");
            int num =Convert.ToInt32(Console.ReadLine());
            try
            {
                if(num >0)
                    Console.WriteLine("Square Root:"+Math.Sqrt(num));
                else
                    throw new NegativeNumberNotAllowed("Invalid Number");

            }
            catch (NegativeNumberNotAllowed e)
            {
                Console.WriteLine(e.Message);
            }
            finally
            {
                Console.WriteLine("Good Bye");
            }
        }
    }
}
```

OUTPUT



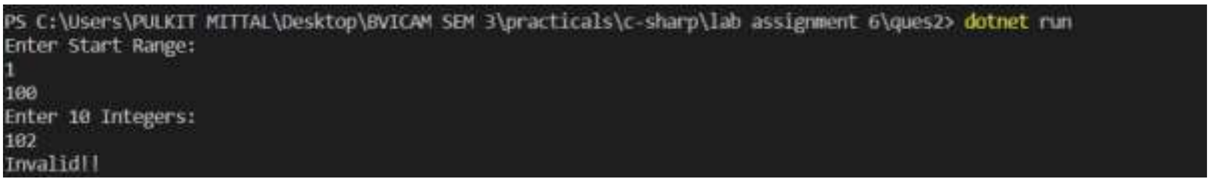
```
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\E> dotnet run
Enter a Number:
49
Square Root:7
Good Bye
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\E> dotnet run
Enter a Number:
-3
Invalid Number
Good Bye
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\E> 
```

EP2 Write a method ReadNumber(int start, int end) that reads an integer from the console in the range [start…end]. In case the input integer is not valid or it is not in the required range throw appropriate exception. Using this method, write a program that takes 10 integers a1, a2, …, a10 such that 1 < a1 < … < a10 < 100.

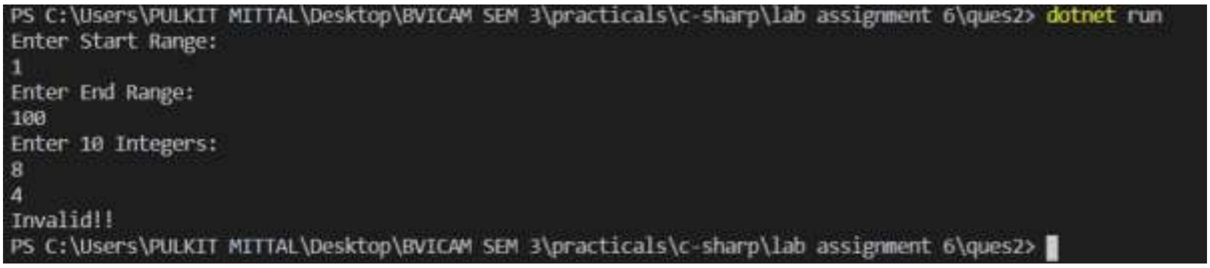
```
using System;

namespace EP2
{
   class Program
   {
      class NotInRange: Exception
      {
      public NotInRange(string message): base(message){}
      }

      public static void ReadNumber(int start, int end)
      {
         int[] arr=new int[10];
         Console.WriteLine("Enter 10 Integers:");
         for(int i=0;i<10;i++)
         {
            arr[i]=Convert.ToInt32(Console.ReadLine());
            if(arr[i]>=start && arr[i]<=end)
            {
               if((i!=0) && (arr[i] <arr[i-1]))
                  throw new NotInRange("Invalid!!");
            }
            else
               throw new NotInRange("Invalid!!");
         }
      }
      static void Main(string[] args)
      {
         Console.WriteLine("Enter Start Range:");
         int start=Convert.ToInt32(Console.ReadLine());
         Console.WriteLine("Enter End Range:");
         int end=Convert.ToInt32(Console.ReadLine());
         try
```

```
            {
                Program.ReadNumber(start,end);

            }catch(NotInRange e)
            {
                Console.WriteLine(e.Message);
            }
            catch(FormatException e)

            {
                Console.WriteLine("Exceptiom: "+e.Message);
            }
        }
    }
}
```

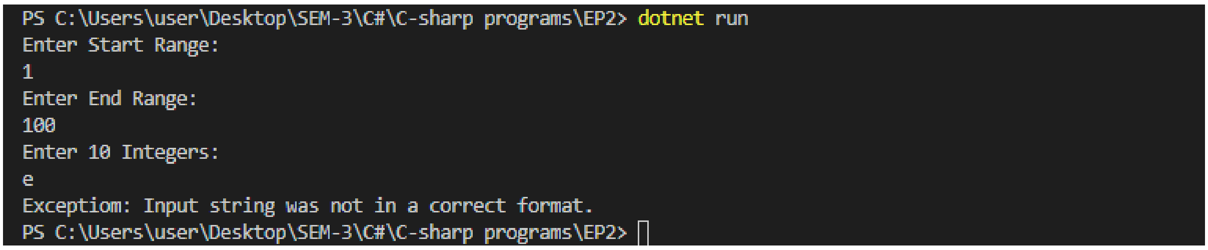
OUTPUT



```
PS C:\Users\PULKIT MITTAL\Desktop\BVICAM SEM 3\practicals\c-sharp\lab assignment 6\ques2> dotnet run
Enter Start Range:
1
100
Enter 10 Integers:
102
Invalid!!
```



```
PS C:\Users\PULKIT MITTAL\Desktop\BVICAM SEM 3\practicals\c-sharp\lab assignment 6\ques2> dotnet run
Enter Start Range:
1
Enter End Range:
100
Enter 10 Integers:
8
4
Invalid!!
PS C:\Users\PULKIT MITTAL\Desktop\BVICAM SEM 3\practicals\c-sharp\lab assignment 6\ques2>
```



```
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\EP2> dotnet run
Enter Start Range:
1
Enter End Range:
100
Enter 10 Integers:
e
Exceptiom: Input string was not in a correct format.
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\EP2>
```

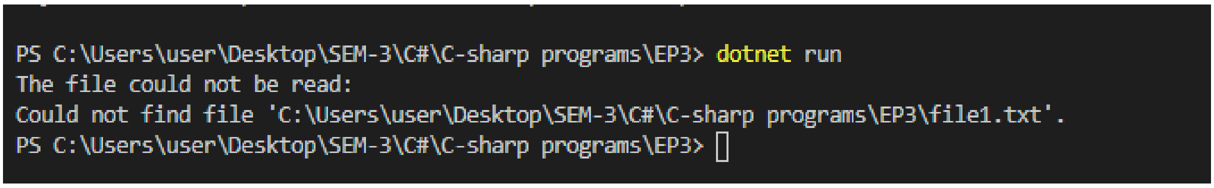
EP3 Implement a program that takes as a parameter the name of a text file, reads the file and returns its content as string. What should the method do if and exception is thrown?

```
using System;
using System.IO;

namespace EP3
{
    class Program
```

```
   {
      public static void ReadFile(string file){
          try {
          using (StreamReader sr = new StreamReader(file)) {
            string line;

            // Read and display lines from the file until
            // the end of the file is reached.
            while ((line = sr.ReadLine()) != null) {
              Console.WriteLine(line);
            }
          }
        } catch (FileNotFoundException e) {
          Console.WriteLine("The file could not be read:");
          Console.WriteLine(e.Message);
        }
      }
      static void Main(string[] args) {
       ReadFile("file1.txt");
     }
   }
}
```

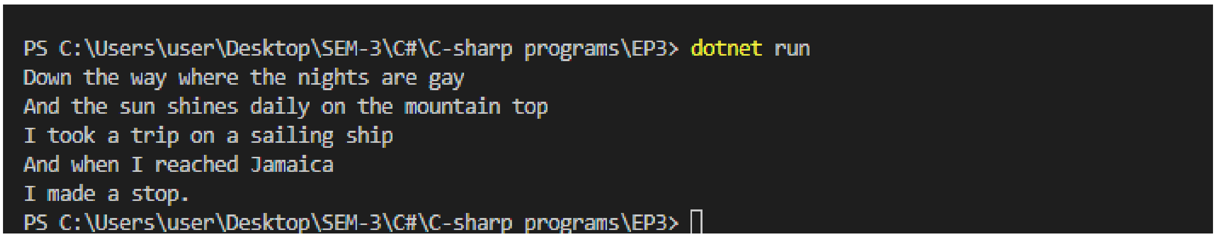
OUTPUT

If File Not Exists:



```
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\EP3> dotnet run
The file could not be read:
Could not find file 'C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\EP3\file1.txt'.
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\EP3> 
```

If File Exists:



```
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\EP3> dotnet run
Down the way where the nights are gay
And the sun shines daily on the mountain top
I took a trip on a sailing ship
And when I reached Jamaica
I made a stop.
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\EP3> 
```

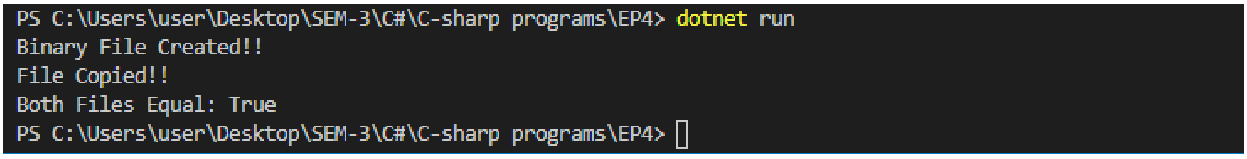
EP4 Implement a program that takes as a parameter the name of a binary file, reads the content of the file and returns it as an array of bytes. Write a method that writes the file content to another file. Compare both files.

```
using System;
using System.IO;

namespace EP4
{
   class Program
```

```
{
    static bool FileEquals(string path1, string path2)
    {
        try
        {
            byte[] file1 = File.ReadAllBytes(path1);
            byte[] file2 = File.ReadAllBytes(path2);
            if (file1.Length == file2.Length)
            {
                for (int i = 0; i < file1.Length; i++)
                {
                    if (file1[i] != file2[i])
                    {
                        return false;
                    }
                }
                return true;
            }
        }catch (FileNotFoundException e) {
            Console.WriteLine("The file could not be read:");
            Console.WriteLine("File Not Found Exception: "+e.Message);
        }
        catch (IOException e){
            Console.WriteLine("IO Exception: "+e.Message);
        }
        catch (Exception e) {
            Console.WriteLine("Exception: "+e.Message);
        }
        return false;
    }
    static void copyBytes(string sourcePath,string destPath){
        try
        {
            using (FileStream fsSource = new FileStream(sourcePath,FileMode.Open, FileAcc
ess.Read))
            {
                byte[] bytes = new byte[fsSource.Length];
                int numBytesToRead = (int)fsSource.Length;
                int numBytesRead = 0;
                while (numBytesToRead > 0)
                {
                    int n = fsSource.Read(bytes, numBytesRead, numBytesToRead);
                    if (n == 0)
                        break;
                    numBytesRead += n;
                    numBytesToRead -= n;
                }
                numBytesToRead = bytes.Length;
```

```
                    // Write the byte array to the other FileStream.
                    using (FileStream fsNew = new FileStream(destPath,FileMode.Create, FileAccess.Write))
                    {
                        fsNew.Write(bytes, 0, numBytesToRead);
                    }
                    Console.WriteLine("File Copied!!");
                }
            }
            catch (IOException e){
                Console.WriteLine("IO Exception: "+e.Message);
            }
            catch (Exception e) {
                Console.WriteLine("Exception: "+e.Message);
            }
        }
        static void Main(string[] args)
        {
            var sourcePath = "source.txt";
            var destPath = "destination.txt";

            try {
                if (!File.Exists(sourcePath)){
                    using (BinaryWriter binWriter =  new BinaryWriter(File.Open(sourcePath, FileMode.Create)))
                    {
                        binWriter.Write("This is EP4 of C# Assignment.\nThis is my content.");
                    }
                    Console.WriteLine("Binary File Created!!");
                }
            } catch (IOException e){
                Console.WriteLine("IO Exception: "+e.Message);
            }
            copyBytes(sourcePath,destPath);
            bool a = FileEquals(sourcePath, destPath);
            Console.WriteLine("Both Files Equal: "+a);
        }
    }
}
```

OUTPUT



```
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\EP4> dotnet run
Binary File Created!!
File Copied!!
Both Files Equal: True
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\EP4> 
```