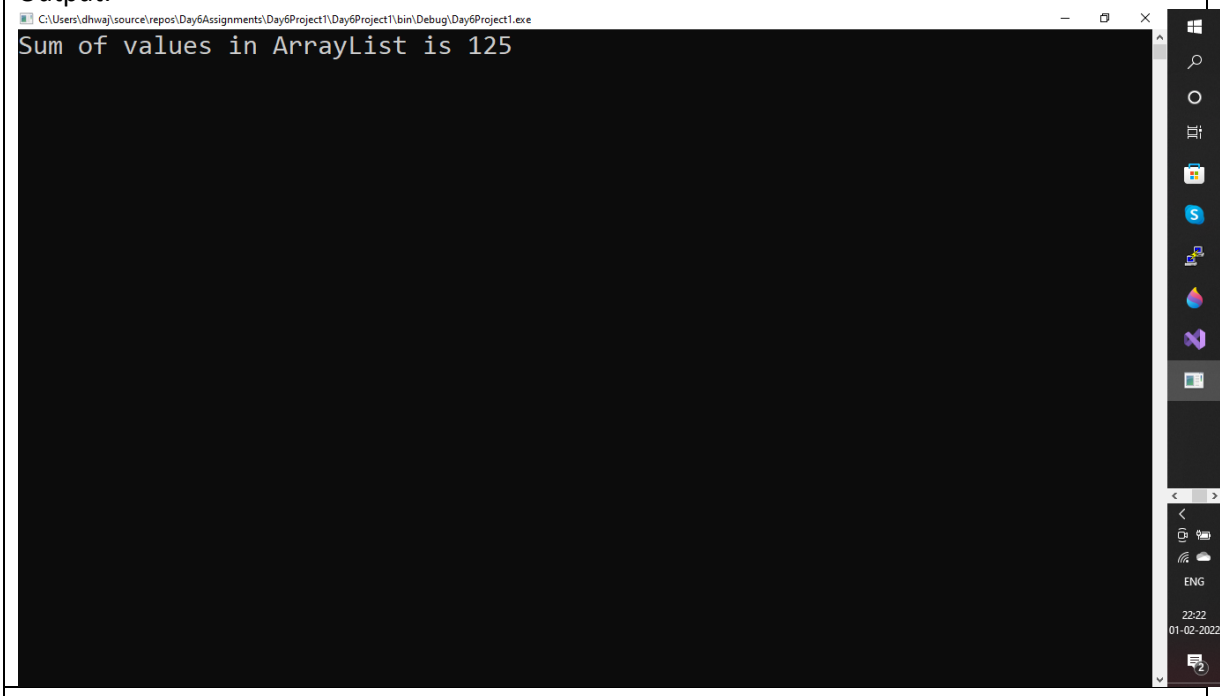
1.Create a simple program to declare ArrayList and assign some values
And find sum.

Code:

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

namespace Day6Project1
{
    //Author:Rc
    /*Purpose:Create ArrayList and assign some values and find sum
     * */
    internal class Program
    {
        static void Main(string[] args)
        {
            //ArrayList Declaration
            ArrayList data = new ArrayList();
            int sum = 0;
            //Assigning values
            data.Add(5);
            data.Add(10);
            data.Add(20);
            data.Add(30);
            data.Add(60);
            //Finding sum using foreach loop
            foreach(var d in data)
            {
                sum = sum + (int)d;
            }
            Console.WriteLine(sum);
            Console.ReadLine();
        }
    }
}
```

Output:



2.Research and find how the values of ArrayList are stored in the memory.

- In ArrayList the values that are stored in it, are boxed into the object type. Then, we have to unbox them.
- Because of boxing the objects, we can store the objects of different types.
- If you store only objects of reference types in ArrayList , then boxing is not used.
- In the ArrayList , boxed values are stored in the heap memory and unboxed values are stored in the stack memory.

3.What are the disadvantages of ArrayList.

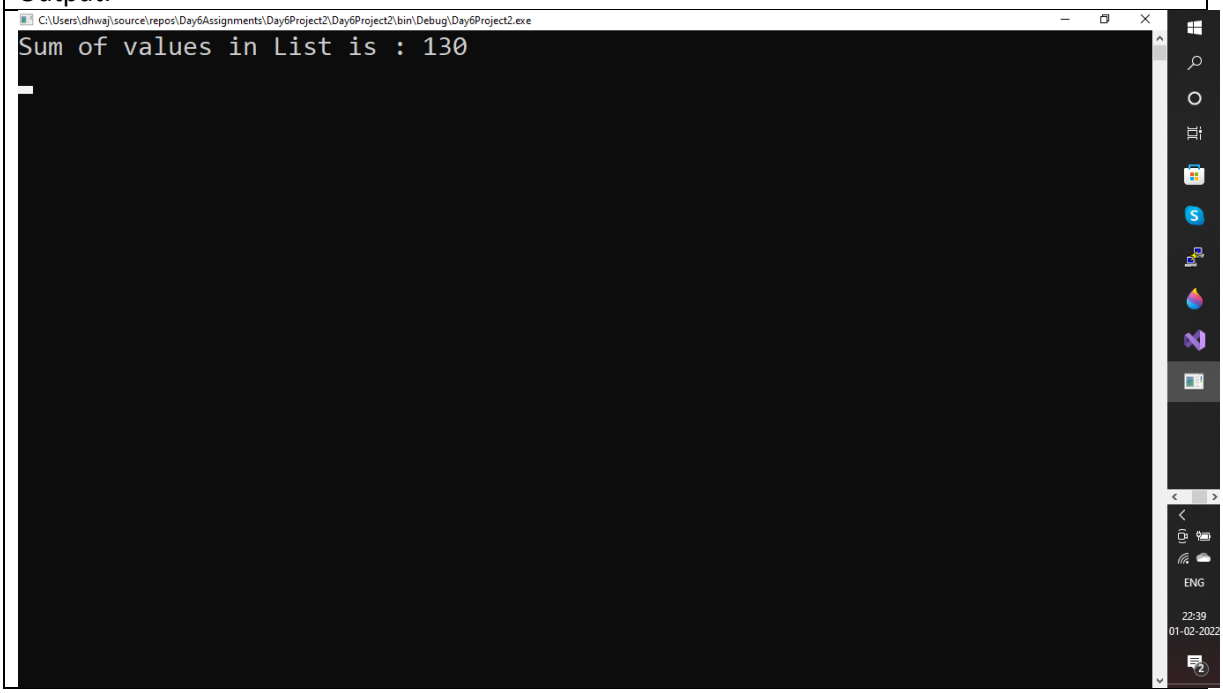
- Every time we have to unbox, to get the values.
- If there is a chance of assigning a wrong datatype then we may get runtime errors. Runtime errors are dangerous than Compile time errors.

4.Create a simple Program to declare List<int> and assign some values and find sum

Code:

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

namespace Day6Project2
{
    //Author:Rc
    /* Create List<T> and assign some values and find sum
     * **/
    internal class Program
    {
        static void Main(string[] args)
        {
            //List<T> Declaration ,Here T is integer
            List<int> data = new List<int>();
            int sum = 0;
            //assigning values
            data.Add(2);
            data.Add(35);
            data.Add(55);
            data.Add(18);
            data.Add(20);
            //Find sum using foreach loop
            foreach (var d in data)
            {
                sum = sum + d;
            }
          Console.WriteLine(sum);
            Console.ReadLine();
        }
    }
}
```

Output:



5.In tabular Format, Write differences between Collections and Generics.

	Collections	Generics
namespcae	System.Collections;	System.Collections.Generics;
Each element of what type	object	Based on input type
Type Casting	Yes, Because of boxing and unboxing.	No.
Example:		
	ArrayList data=new ArrayList();	List<int> data=new List<int>();

6.Research and find how values of List<T> are stored in the memory.

- Generally, List<T> stores objects of same type.
- It implements IList generic interface using array whose size is dynamically increased as required.
- The values are stored in the managed Heap.
- In List<T> makes a single array of <Type>, and can store the values directly. If List<int> means it makes a single array of integers and store the values directly.

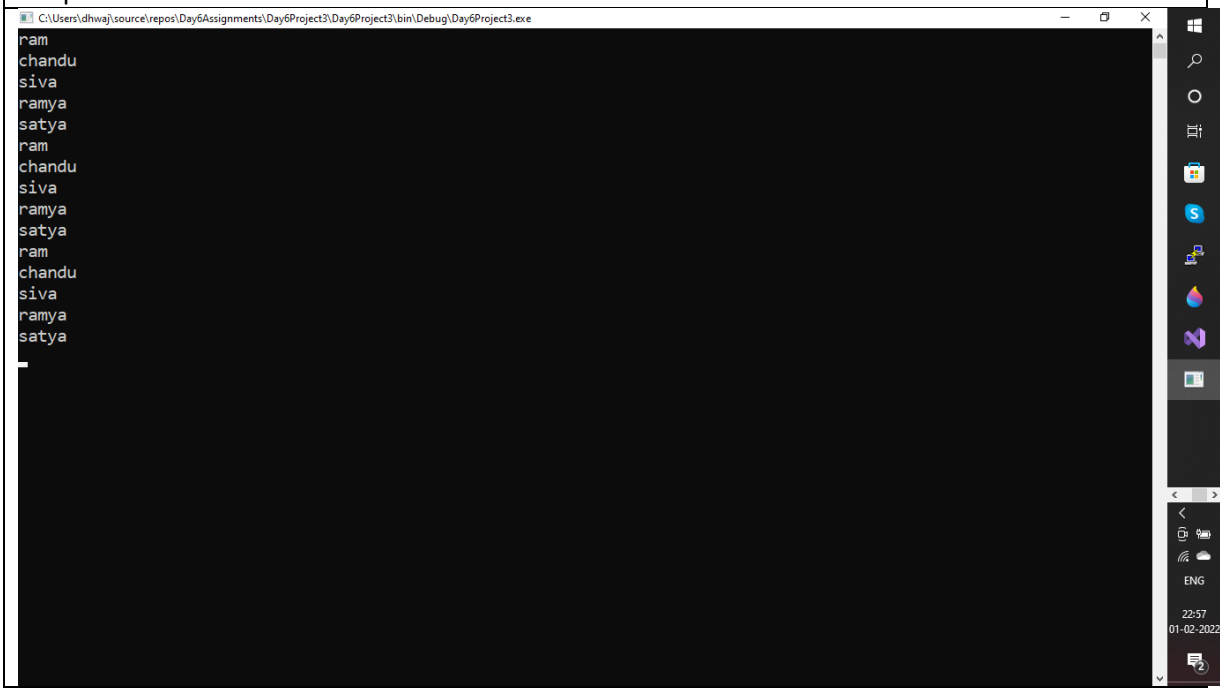
7.Write a C# Program to declare List<string> and add 5 values and print the values using
a.for loop
b.foreach loop
c.Lambda expression

Code:

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

namespace Day6Project3
{
    //Author: Rc
    /* Declare List<string> and add 5 values and print values using
     * for loop
     * foreach loop
     * lambda expression
     * **/
    internal class Program
```

```
    {
        static void Main(string[] args)
        {
            //List<string> Declaration
            List<string> data = new List<string>();
            //Adding values to List<string>
            data.Add("ram");
            data.Add("chandu");
            data.Add("siva");
            data.Add("ramya");
            data.Add("satya");

            //using for loop
            for(int i=0;i<data.Count;i++)
            {
                Console.WriteLine(data[i]);
            }

            //using foreach loop
            foreach(var d in data)
            {
                Console.WriteLine(d);
            }

            //using lambda expression
            data.ForEach(p=>Console.WriteLine(p));

            Console.ReadLine();
        }
    }
}
```

Output:

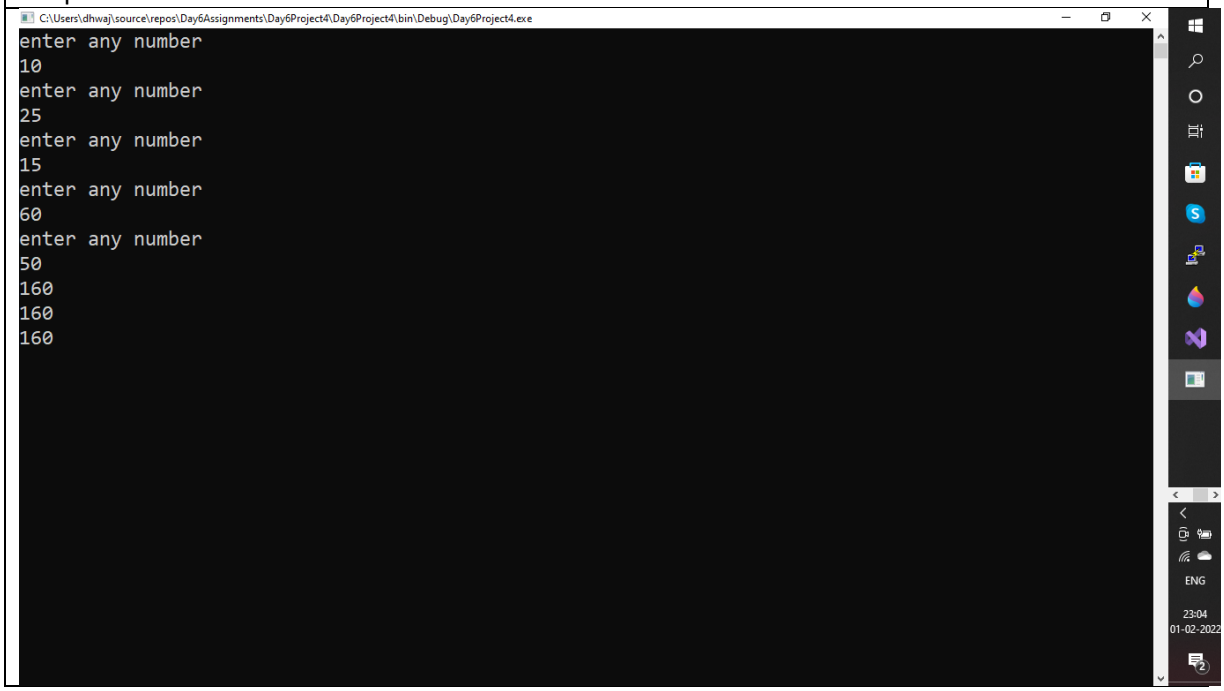


8. .Write a C# Program to declare List<int> and read 5 values from user and find sum using
a.for loop
b.foreach loop
c.Lambda expression

Code:

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

namespace Day6Project4
{
    //Author: Rc
    /* Declare List<int> and read 5 values from user and find sum using
     * for loop
     * foreach loop
     * lambda expression
     * **/
    internal class Program
    {
        static void Main(string[] args)
        {
            //List<int> Declaration
            List<int> data = new List<int>();
            int sum1 = 0,sum2 = 0,sum3 = 0;
            int temp;

            //user input
            for(int i=1;i<=5;i++)
            {
                Console.WriteLine("enter any number");
                temp = Convert.ToInt32(Console.ReadLine());
                data.Add(temp);
            }

            //sum using for loop
            for (int i = 0; i < data.Count; i++)
            {
                sum1+=(data[i]);
            }

            //using foreach loop
            foreach (var d in data)
            {
                sum2+=(d);
            }

            //using lambda expression
            data.ForEach(d=> sum3+=d);

            Console.WriteLine(sum1);
            Console.WriteLine(sum2);
            Console.WriteLine(sum3);


            Console.ReadLine();
        }
    }
}
```

Output:



9.In Tabular format, write all datatypes and their respective alias names.

Data Types and their respective alias names:

DATATYPE	ALIAS NAME
1. byte	Byte
2. ushort	UInt16
3. uint	UInt32
4. ulong	UInt64
5. sbyte	SByte
6. short	Int16
7. int	Int32

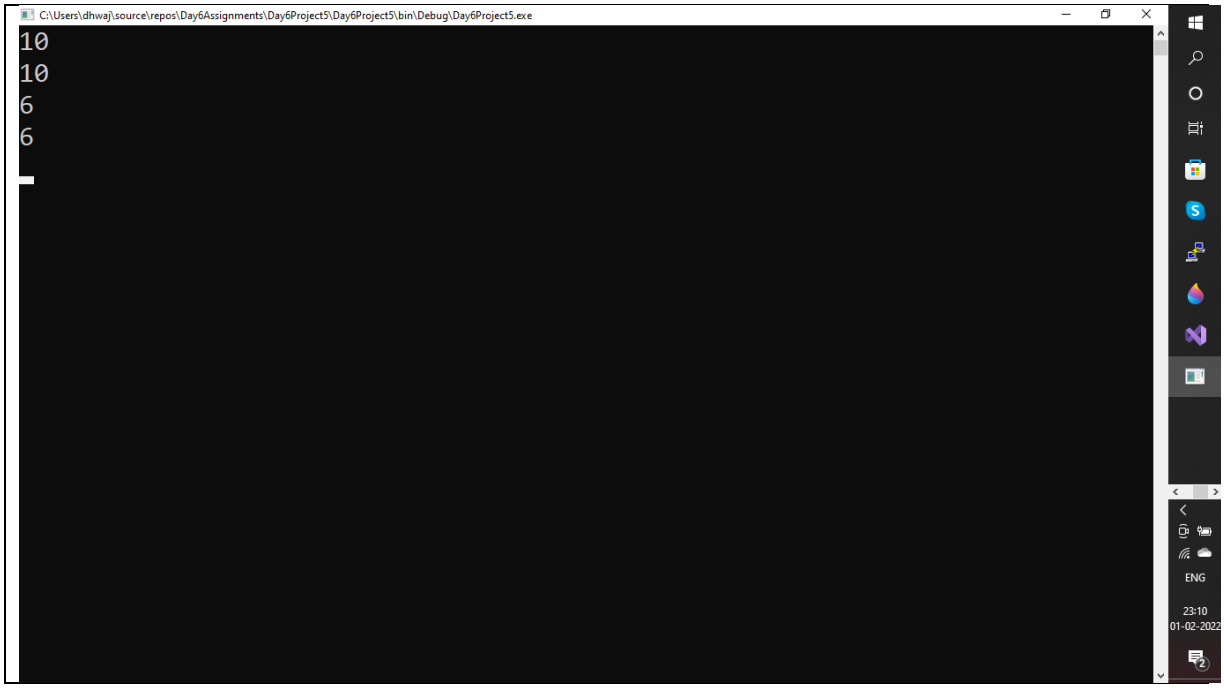
8. long	Int64
9. float	Single
10.double	Double
11.decimal	Decimal
12.char	Char
13.String	String
14.bool	Boolean

10.Write a C# Program for demonstrating Implicit and Explicit Type-casting.

Code:

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

namespace Day6Project5
{
    //Author: Rc
    /*Purpose:Demonstrating Implicit and Explicit type casting
     * */
    internal class Program
    {
        static void Main(string[] args)
        {
            //implicit
            short p=10;
            int q=p;
            Console.WriteLine(p);
            Console.WriteLine(q);

            //explicit
            int a = 6;
            short b=Convert.ToInt16(a);
            Console.WriteLine(a);
            Console.WriteLine(b);

            Console.ReadLine();
        }
    }
}
```

Output:


C:\Users\dhwaj\source\repos\Day6Assignments\Day6Project5\Day6Project5\bin\Debug\Day6Project5.exe
10
10
6
6
ENG
23:10
01-02-2022