1)Write a console program that obtains three int values from the user and displays the total.

Introduction:

The simplest method to get input from the user is by using the Readline( ) method of the console class. However, Read( ) and Readkery( ) are also available for getting input from the user. They are also included in Console class.

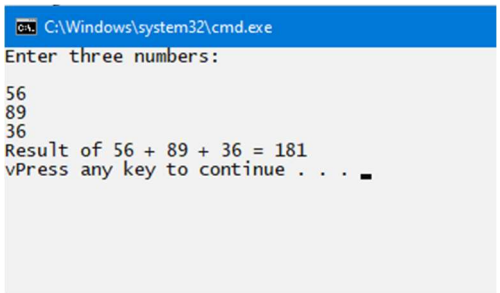
Syntax:

datatype identifier – Console.ReadLine ( )

Code:

```
using System;
namespace ConsoleApp1{
      class Lab1{
            static void Main(string[] args){
                  int n1, n2, n3, result;
                  Console.WriteLine("Enter three numbers: \n");
                  n1 = Convert.ToInt32(Console.ReadLine());
                  n2 = Convert.ToInt32(Console.ReadLine());
                  n3 = Convert.ToInt32(Console.ReadLine());
                  result = n1 + n2 + n3;
                  Console.WriteLine("Result of {0} + {1} + {2} = {3}",
                  n1, n2, n3, result);
                  Console.ReadKey();
            }
         }
      }
```

Output:



2) Write a program using conditional statements to check whether the character is rowel or consonant.

Introduction:

A statement that can be executed based on a condition is known as a "conditional statement. The statement is often a block of code. There are two types; conditional branching and looping.

Syntax:

if (condition)

<statements>;

else if (condition)

<statements>;

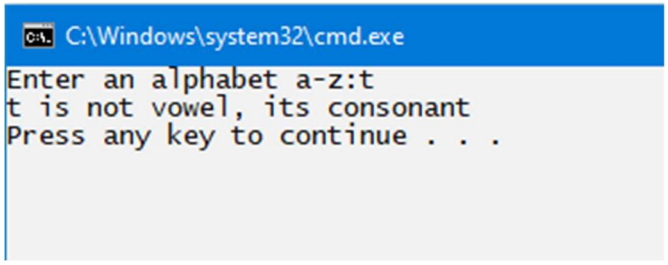
Else

Code:

```
using System;
namespace ConsoleApp1{
    class Lab1{
        static void Main(string[] args){
            char ch;
            Console.Write("Enter an alphabet a-z:");
            ch = Convert.ToChar(Console.ReadLine());
            int i = ch;
            if (i >= 48 && i <= 57){
                Console.Write("You have entered a number, please enter an alphabet.");
            }
            else{
                switch (ch){
                    case 'a':
                        Console.WriteLine($"{ch} is vowel.");
                        break;
```

```
                    case 'e':
                        Console.WriteLine($"{ch} is vowel.");
                        break;
                    case 'i':
                        Console.WriteLine($"{ch} is vowel.");
                        break;
                    case 'o':
                        Console.WriteLine($"{ch} is vowel.");
                        break;
                    case 'u':
                        Console.WriteLine($"{ch} is vowel.");
                        break;
                    default:
                        Console.WriteLine($"{ch}    is    not    vowel,    its
consonant");
                        break;
                }
            }
            Console.ReadKey();
        }
    }
}
```

Output:



3) Write a console program that print the string with double quotation marks around each word in a string. “Welcome” "6" "the" “Dotnet" "Technology".

Introduction:

A verbatim string is created using a special symbol @ which is known as a verbatim identifier. If a string contains @ as a prefix followed by double quotes, then compiler identifies that string as a verbatim string and compile that string.

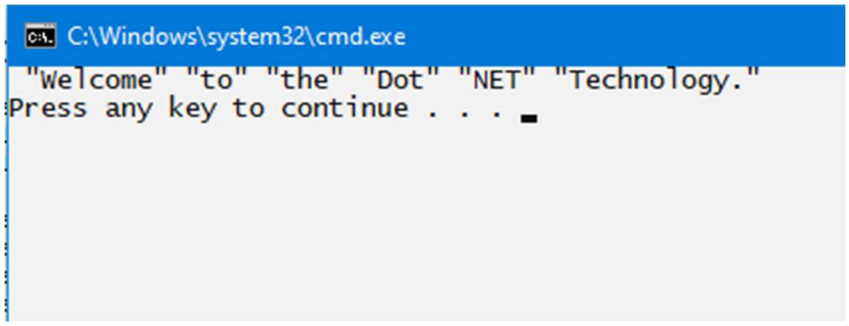
Syntax:

dataType variable-name = @ “ “ “ required_string ” ” ” ;

Code:

```
using System;
namespace ConsoleApp2{
    class Lab1{
        static void Main(string[] args){
            String line = @" ""Welcome"" ""to"" ""the"" ""Dot"" ""NET"" ""Technology."" ";
            Console.WriteLine(line);
        }
    }
}
```

Output:



4) Write a console program to accept 10 numbers from user in an array and find largest number.

Introduction:

An array is a group of like-typed variables that are referred to by a common name. And each data item is called an element of the array. The data types of the elements may be any valid data type like char, int, float, etc. and the elements are stored in a contiguous location. Length of the array specifies the number of elements present in the array. In C# the allocation of memory for the arrays is done dynamically. And arrays are kinds of objects, therefore it is easy to find their size using the predefined functions. The variables in the array are ordered and each has an index beginning from 0. Arrays in C# work differently than they do in C/C++. A jagged array elements are reference types and are initialized to null. Array elements can be of any type, including an array type.

Array types are reference types which are derived from the abstract base type Array. These types implement IEnumerable and for it, they use foreach iteration on all arrays in C#.

Syntax:

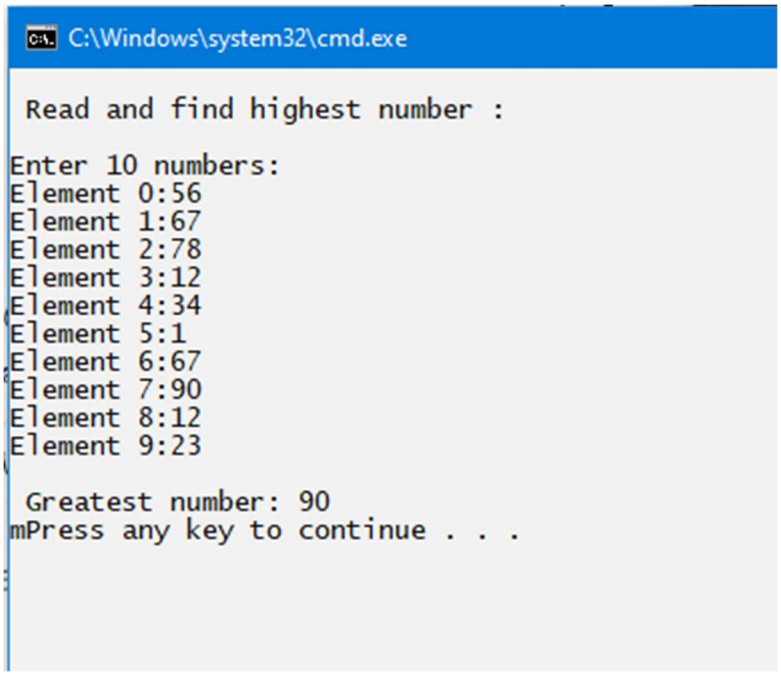
string [ ] var_name = new string[provided_size_here];

Code:

```
using System;
namespace ConsoleApp2
{
    class Lab1
    {
        static void Main(string[ ] args)
        {
            int[ ] arr = new int[10];
            int i;
            Console.Write("\n Read and find highest number : \n");
            Console.Write("\n");
            Console.Write("Enter 10 numbers: \n");
            for (i = 0; i < 10; i++)
            {
                Console.Write("Element {0}:", i);
```

```
                arr[i] = Convert.ToInt32(Console.ReadLine());
            }
            Array.Sort(arr);
            Console.WriteLine("\n Greatest number: " + arr[arr.Length - 1]);
            Console.ReadKey();
        }
    }
}
```

Output:



5) Write a console program to demonstrate following String operation substring, split and replace.

Introduction:

- ➔ substring starts at a specified character position and continues to the end of the string.
- ➔ split splits a string into a maximum number of substrings based on specified delimiting characters and options.
- ➔ Replace returns a new string in which all occurrences of a specified Unicode character are replaced with another one.

Syntax:

- ➔ string substring (int startIndex);
- ➔ string [ ] var2_name = var1_name.Split(string [ ], Int 32);
- ➔ string var_name 2 = var_name1.Replace (string, string);

Code:

```
using System;

namespace ConsoleApp2

{

    class Lab1

    {

        static void Main(string[] args)

        {

            string quote = "Everyone should have their mind blown once a day.";

            Console.WriteLine($" --> Given sentence : \n {quote}");

            Console.WriteLine("in--> using substring :");

            string cut = quote.Substring(0, 15); Console.WriteLine(cut);

            Console.WriteLine("\n --> Using Split :");

            string[] s1 = quote.Split();

            foreach(String s in s1)

            {

                Console.WriteLine(s);

            }
```

```
            Console.WriteLine("\n--> Using Replace :");
            string s2 = quote.Replace("Everyone", "Anyone");
            Console.WriteLine(s2);
            Console.ReadKey();
        }
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe
 --> Given sentence :
 Everyone should have their mind blown once a day.
in--> using substring :
Everyone should

 --> Using Split :
Everyone
should
have
their
mind
blown
once
a
day.

--> Using Replace :
Anyone should have their mind blown once a day.
jPress any key to continue . . .
```

6) Write a console program to obtain a integer values and a operator (+ , - , * , /) from users and display the result of the operations.

Introduction:

Switch is a selection statement It executes Code of one of the conditions based on a pattern match with the specified match expression.

Syntax

```
switch (expression) {
case exp_val1:
        stmt...  break;
case exp_val2:
        stmt …..break;
default:
        stmt .. break;
}
```

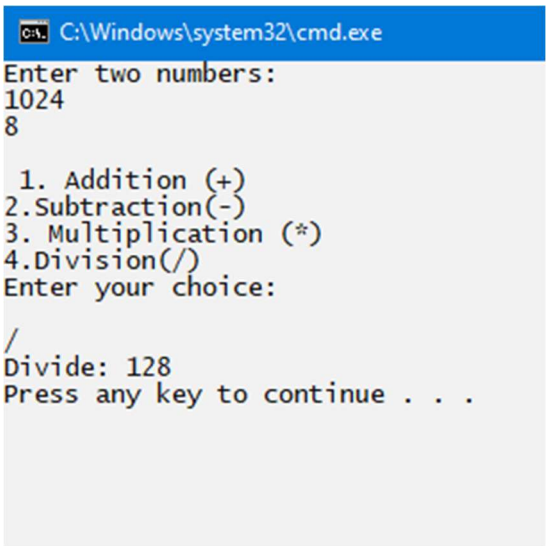
Code:

```
using System;
namespace ConsoleApp2
{
class Lab1
{
    static void Main(string[] args)
    {
        Console.WriteLine("Enter two numbers:");
        float n1 = float.Parse(Console.ReadLine());
        float n2 = float.Parse(Console.ReadLine());

        Console.WriteLine("\n 1. Addition (+)");
        Console.WriteLine("2.Subtraction(-)");
        Console.WriteLine("3. Multiplication (*)");
        Console.WriteLine("4.Division(/)");
```

```
        Console.WriteLine("Enter your choice: \n");
        //Reading choice
        char c = Convert.ToChar(Console.ReadLine());
        switch (c)
        {
            case '+': Console.WriteLine("Add :" + (n1 + n2));
                break;
            case '-': Console.WriteLine("Subtract: " + (n1 - n2));
                break;
            case '*': Console.WriteLine("Multiply: " + (n1 * n2));
                break;
            case '/': Console.WriteLine("Divide: " + (n1 / n2));
                break;
            default: Console.WriteLine("in Choose only 1 to 4 • ");
                break;
        }
        Console.ReadKey();
    }
  }
}
```

Output:



7)Write a console program to copy the elements of one array into array.

Introduction:

Array.CopyTo copies all the elements of the current array to the specified destination array. This method should be called from the source array and it takes two parameters. The first being the array you want to copy to, and the second parameter tells it what index of the destination array it should start copying into. Let's take a look at an example. The values of 1st array are copied using for loop and both results are displayed.

Syntax:

int [] 1st_aray_name = new int [size];

for (i = ---) {

2nd_array = 15t_array_nameame [i];

}

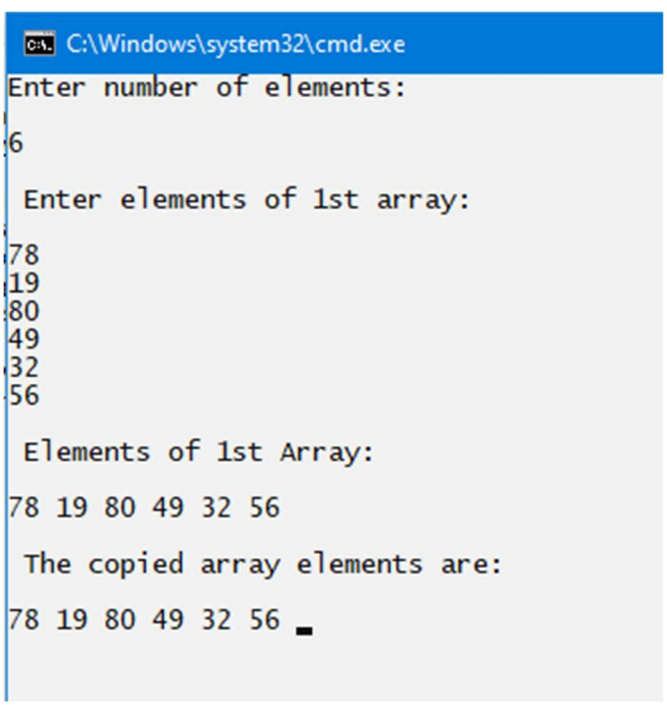
Code:

```
using System;
namespace ConsoleApp2{
    class Lab1{
        static void Main(string[] args){
            int[] first = new int[15];
            Console.WriteLine("Enter number of elements: \n");
            int num = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("\n Enter elements of 1st array: \n");
            for (int i = 0; i < num; i++){
                first[i] = Convert.ToInt32(Console.ReadLine());
            }
            int[] second = new int[num];
            for (int i = 0; i < num; i++){
                second[i] = first[i];
            }
            //Display 1st Array
            Console.WriteLine("\n Elements of 1st Array: \n");
```

```
            for (int i = 0; i < num; i++){
                Console.Write(first[i] + " ");
            }
            Console.WriteLine();

            //Display 2nd Array
            Console.WriteLine("\n The copied array elements are: \n");
            for (int i = 0; i < num; i++){
                Console.Write(second[i] + " ");
            }
            Console.ReadKey();
        }
    }
}
```

Output:



8) Write a console program to implement namespace student in C#.

Introduction:

A namespace is a domain for type names. Types are typically organized into hierarchical namespaces, making them easier to find and avoid conflicts. Namespaces are used to logically arrange classes, structs, interfaces, enums and delegates. The namespaces in C# can be nested. That means one namespace can contain other namespaces also. The .NET framework already contains number of standards namespaces like System, System.Net, System.IO etc. In addition to these standard namespaces the user can define their own namespaces.

Syntax:

namespace Name1 {

class className {

// Body of namespace

namespace Name2 {

//main method

}

Code:

```
using System;
namespace First {
public class Hello
{
    public void sayHello() { Console.WriteLine("Hello First Namespace"); }
}
}
namespace Second
{
    public class Hello
    {
```

```
        public void sayHello() { Console.WriteLine("Hello Second Namespace");
}
    }
}
public class TestNamespace
{
    public static void Main()
    {
        First.Hello h1 = new First.Hello();
        Second.Hello h2 = new Second.Hello();
        h1.sayHello();
        h2.sayHello();

    }
}
```

Output: