

# Swapping 2 numbers – with extra variable

```
namespace swapping_numbers
{
    class Program
    {
        static void Main(string[] args)
        {
            int x = 10;
            int y = 20;
            //swapping numbers with extra variable
            int temp = x;
            x = y;
            y = temp;
            Console.WriteLine(x);
            Console.WriteLine(y);
            Console.ReadLine();
        }
    }
}
```

# Swapping 2 numbers – without extra variable

```
namespace swapping_numbers
{
    class Program
    {
        static void Main(string[] args)
        {
            int x = 10;
            int y = 20;
            //swapping numbers without extra variable
            x = x + y;
            y = x - y;
            x = x - y;
            Console.WriteLine(x);
            Console.WriteLine(y);
            Console.ReadLine();
        }
    }
}
```

# Factorial program

```
namespace factorial_program
{
    class Program
    {
        static void Main(string[] args)
        {
            int x = 5; //lets find factorial of 5
            int result = 1; //for storing result

            for(int i = x; i > 1; i--)
            {
                result = result * i;
            }
            Console.WriteLine("factorial of" + x + "is" + result);
            Console.ReadLine();
        }
    }
}
```

# Fibonacci series

```
namespace Fibonacci_series
{
    class Program
    {
        static void Main(string[] args)
        {
            int fno = 0;
            int sno = 1;
            Console.WriteLine(fno);
            Console.WriteLine(sno);
            int result = fno + sno;
            //keep printing fibonacci series until it reaches 100
            while (result < 100)
            {
                Console.WriteLine(result);
                fno = sno;
                sno = result;
                result = fno + sno;
            }
        }
    }
}
```

# Prime number program

```
namespace prime_number
{
    class Program
    {
        static void Main(string[] args)
        {
            int num = 49; //store the number
            Boolean isprime = true;
            for(int i = 2; i < num; i++)
            {
                if(num % i == 0)
                {
                    isprime = false;
                    break;
                }
            }
            if (isprime)
            {
                Console.WriteLine(num + "is prime");
            }
            else
            {
                Console.WriteLine(num + "is not prime");
            }
        }
    }
}
```

# Palindrome number

```
namespace palindrome__number
{
    class Program
    {
        static void Main(string[] args)
        {
            int num = 159; // store some number
            int temp = num; //take a backup copy of the number
            int rev = 0; // is used to stored reversed number

            while (num > 0)
            {
                int last = num % 10;
                num = num / 10;
                rev = (rev * 10) + last;
            }
            if (rev == temp)
            {
                Console.WriteLine(temp+"is palindrome number");
            }
            else
            {
                Console.WriteLine(temp+"is not palindrome number");
            }
        }
    }
}
```

# Armstrong number

```
namespace armstrong_number
{
    class Program
    {
        static void Main(string[] args)
        {
            int num = 153; //store some number
            int temp = num; // take back up of the number
            int result = 0; //to store result number
            {
                while (num > 0)
                {
                    int last = num % 10;
                    result = result + (last * last * last);
                    num = num / 10;
                }
                if (temp == result)
                {
                    Console.WriteLine(temp+" is armstrong num");
                }
                else
                {
                    Console.WriteLine(temp+"is not armstrong number");
                }
            }
        }
    }
}
```

# Biggest element in an array

```
namespace BiggestElementinArray
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] arr = { 10, 9, 11, 23, 8 };
            int big = arr[0];

            for(int i = 1; i < arr.Length; i++)
            {
                if (arr[i] > big)
                {
                    big = arr[i];
                }
            }
            Console.WriteLine("Biggest element in array is .."+big);
        }
    }
}
```



# Searching an element in an array

```
namespace search_element_in_array
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] arr = { 10, 9, 8, 11, 12 };
            int element = 11; //element to find its position
            int pos = -1;
            for (int i = 0; i < arr.Length; i++)
            {
                if (arr[i] == element)
                {
                    pos = i; // store position in pos
                    break; //gp out for loop
                }
            }
            if (pos == -1)
            {
                Console.WriteLine(element + "not found");
            }
            else
            {
                Console.WriteLine(element + " found at position"+pos);
            }
        }
    }
}
```

# Sorting an Array – bubble sort

```
namespace bubble_sort
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] arr = { 10, 8, -9, 11, 12 };
            int len = arr.Length;
            //bubble sort logic
            for(int i = 0; i < len - 1; i++)
            {
                for(int j = 0; j < len - i - 1; j++)
                {
                    if (arr[j] > arr[j + 1])
                    {
                        int temp = arr[j];
                        arr[j] = arr[j + 1];
                        arr[j + 1] = temp;
                    }
                }
            }
            //lets print array
            for(int i = 0; i < len; i++)
            {
                if(arr[i] != -1)
                {
                    Console.WriteLine(arr[i]);
                }
            }
        }
    }
}
```

# Removing duplicates from array

```
namespace RemoveDuplicates
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] arr = { 10, 8, 10, 8, 12 };
            int len = arr.Length;
            //logic
            for(int i = 0; i < len - 1; i++)
            {
                for(int j = i + 1; j < len - 1; j++)
                {
                    if (arr[i] == arr[j] && arr[i] != -1)
                    {
                        arr[j] = -1;
                    }
                }
            }
            //lets print array
            for(int i=0; i < len; i++)
            {
                if(arr[i] != -1)
                    Console.WriteLine(arr[i]);
            }
        }
    }
}
```

# Reverse a string

```
namespace reverse_a_string
{
    class Program
    {
        static void Main(string[] args)
        {
            string name = "palle";
            for(int i = name.Length-1; i > 0; i--)
            {
                Console.WriteLine(name[i]);
            }
            Console.ReadLine();
        }
    }
}
```

# Palindrome string program

```
namespace palindrome_string
{
    class Program
    {
        static void Main(string[] args)
        {
            string name = "madam";
            string reverse = "";
            for(int i = name.Length - 1; i >= 0; i--)
            {
                reverse = reverse + name[i];
            }
            if (reverse.Equals(name))
            {
                Console.WriteLine(name + " is a palindrome string");
            }
            else
            {
                Console.WriteLine(name + "is not a palindrome string");
            }
        }
    }
}
```

# Counting words in a String

```
namespace CountingWordsinaString
{
    class Program
    {
        static void Main(string[] args)
        {
            string name="palle technologies .net training";
            int count = 1;

            for(int i = 0; i < name.Length; i++)
            {
                if (name[i] == ' ')
                {
                    count++;
                }
            }
            Console.WriteLine("total words =" +count);
        }
    }
}
```

# Reversing words in a String

```
namespace Reverse_words
{
    class Program
    {
        static void Main(string[] args)
        {
            string name = "palle technologies .net training";
            string[] arr = name.Split();

            for(int i = 0; i < name.Length; i++)
            {
                string element = arr[i];
                for(int j = element.Length - 1; j >= 0; j--)
                {
                    Console.WriteLine(element[j]);
                }
            }
        }
    }
}
```

# Count vowels in a String

```
namespace count_vowels
{
    class Program
    {
        static void Main(string[] args)
        {
            string name = "palle technologies";
            int count = 0;
            for (int i = 0; i < name.Length; i++)
            {
                char ch = name[i];
                object u = null;
                if(ch=='a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
                {
                    count++;
                }
            }
            Console.WriteLine("total vowels" + count);
        }
    }
}
```



# Right angle triangle pattern

```
namespace RightAngleTriangle
{
    class Program
    {
        static void Main(string[] args)
        {
            for(int i = 1; i <= 4; i++)
            {
                for(int j = 1; j <= i; j++)
                {
                    Console.Write("*");
                }
            }
        }
    }
}
```

```
*
**
***
****
```

# Reversed right angle triangle pattern

```
namespace ReversedRightAngleTringle
{
    class Program
    {
        static void Main(string[] args)
        {
            for(int i = 4; i >= 1; i--)
            {
                for(int j = 1; j <= i; j++)
                {
                    Console.Write("*");
                }
            }
        }
    }
}
```

\*\*\*\*

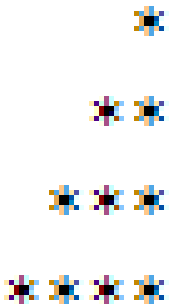
\*\*\*

\*\*

\*

# Mirrored right angle triangle pattern

```
namespace mirrorRightAngle
{
    class Program
    {
        static void Main(string[] args)
        {
            for(int i = 1; i <= 4; i++)
            {
                for(int j = 1; j <= 4; j++)
                {
                    if (j <= 4 - i)
                    {
                        Console.Write(" ");
                    }
                    else
                    {
                        Console.Write("*");
                    }
                }
                Console.WriteLine();
            }
        }
    }
}
```



The pattern consists of four rows of stars. The first row has one star, the second has two, the third has three, and the fourth has four. The stars are mirrored across a vertical axis, creating a symmetrical shape.

# Pyramid star patter

```
namespace pyramid_star_pattern
```

```
{
```

```
    class Program
```

```
    {
```

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

```
        {
```

```
            for(int i = 1; i <= 4; i++)
```

```
            {
```

```
                for(int j = 1; j <= 4 + i - 1; j++)
```

```
                {
```

```
                    if (j <= 4 - i)
```

```
                    {
```

```
                        Console.Write(" ");
```

```
                    }
```

```
                    else
```

```
                    {
```

```
                        Console.Write("*");
```

```
                    }
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
    *
```

```
  ***
```

```
 *****
```

```
 *******
```