Problem 1 : Print triangle - and allow user to set height of it in. Like in the following case it’s 4

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

Solution :

using System;

class Program

{

static void Main()

{

Console.WriteLine("Enter the height of the triangle:");

if (int.TryParse(Console.ReadLine(), out int height))

{

PrintTriangle(height);

}

else

{

Console.WriteLine("Invalid input. Please enter a valid integer.");

}

}

static void PrintTriangle(int height)

{

for (int i = 1; i <= height; i++)

{

for (int j = 0; j < height - i; j++)

{

Console.Write(" ");

}

for (int k = 0; k < 2 \* i - 1; k++)

{

Console.Write("\*");

}

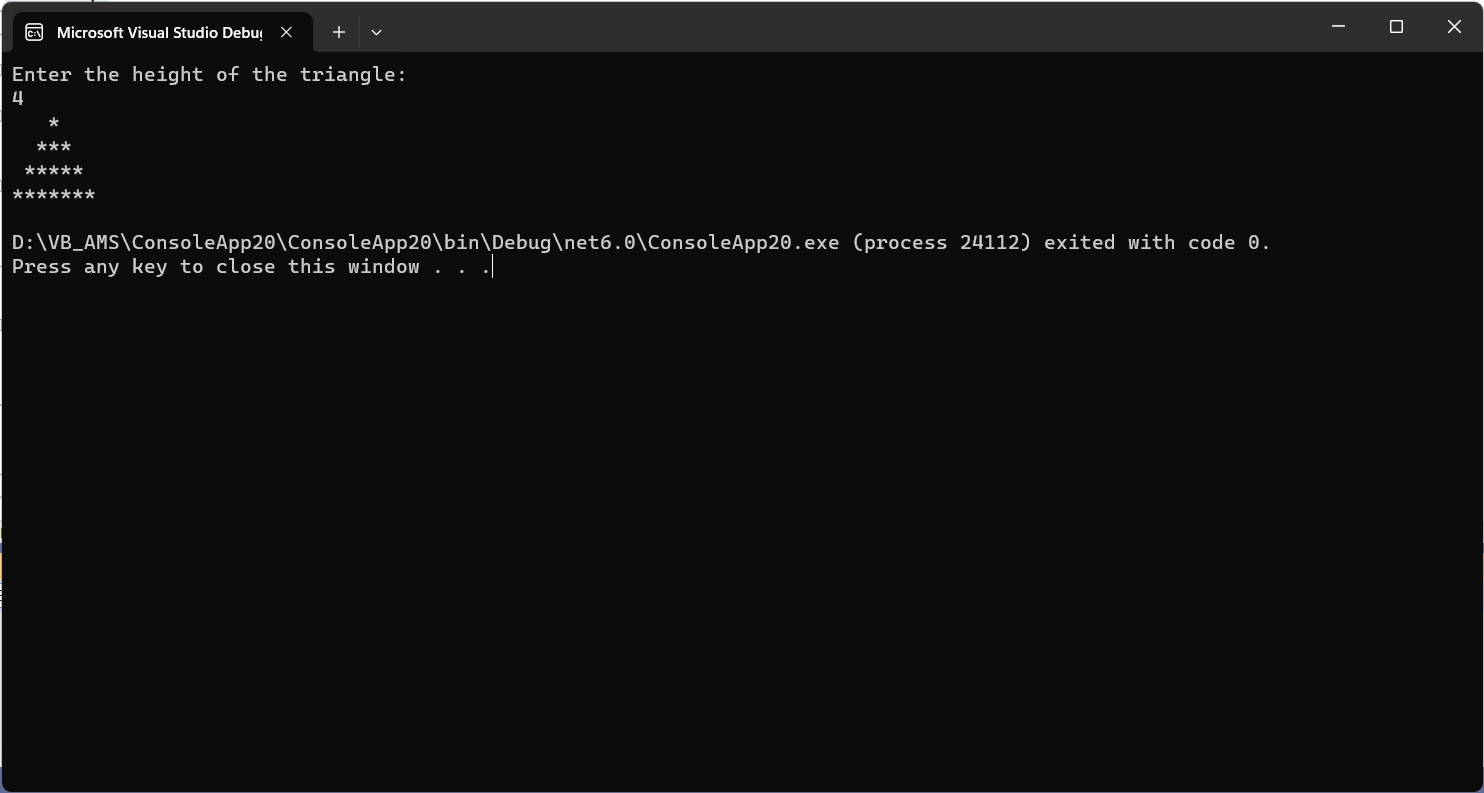
Console.WriteLine();

}

}

}

Output :



Problem 2 : Find valid date (MMDDYYYY) from String.

For Example : Hdjsh as2323234jghsd hjsdg sdhk 12212021 idf32432 32423 d34234jh dfh

Solution 1 : Without user input

using System;

using System.Text.RegularExpressions;

class ValidDateFinder

{

static void Main()

{

string input = "Hdjsh as2323234jghsd hjsdg sdhk 12212021 idf32432 32423 d34234jh dfh";

string validDate = FindValidDate(input);

if (validDate != null)

{

Console.WriteLine("Valid date found: " + validDate);

}

else

{

Console.WriteLine("No valid date found in the input.");

}

}

private static string FindValidDate(string input)

{

// Define a regular expression for a date pattern (MMDDYYYY)

Regex regex = new Regex(@"\b(0[1-9]|1[0-2])(0[1-9]|[12][0-9]|3[01])(19|20)\d\d\b");

Match match = regex.Match(input);

if (match.Success)

{

return match.Value;

}

else

{

return null;

}

}

}

Output : Valid date found: 12212021

Solution 2 : With user input

using System;

using System.Text.RegularExpressions;

class ValidDateFinder

{

static void Main()

{

Console.Write("Enter a string: ");

string input = Console.ReadLine();

string validDate = FindValidDate(input);

if (validDate != null)

{

Console.WriteLine("Valid date found: " + validDate);

}

else

{

Console.WriteLine("No valid date found in the input.");

}

}

private static string FindValidDate(string input)

{

Regex regex = new Regex(@"\b(0[1-9]|1[0-2])(0[1-9]|[12][0-9]|3[01])(19|20)\d\d\b");

Match match = regex.Match(input);

if (match.Success)

{

return match.Value;

}

else

{

return null;

}

}

}

Output : Valid date found: 12212021