using System;

using System.Text.RegularExpressions;

using System.Collections.Generic;

class Program

{

static void Main()

{

Console.WriteLine("Enter your code (e.g., 'var a1 = 12@; float b2 = 3.14$; int c3 = 5#;'):");

string inputCode = Console.ReadLine();

// Regex to match variables starting with a/b/c, ending with digit, with special chars

string pattern = @"(var|int|float|double)\s+([abc][a-zA-Z]\*\d+)\s\*=\s\*([^;]+);";

MatchCollection matches = Regex.Matches(inputCode, pattern);

List<(string VarName, string SpecialSymbol, string TokenType)> extracted = new List<(string, string, string)>();

foreach (Match match in matches)

{

string tokenType = match.Groups[1].Value;

string varName = match.Groups[2].Value;

string value = match.Groups[3].Value;

// Find all special characters

MatchCollection specialChars = Regex.Matches(value, @"[^\w\s.]");

if (specialChars.Count > 0)

{

extracted.Add((varName, string.Join("", specialChars), tokenType));

}

}

// Display results in a box-style table

if (extracted.Count > 0)

{

Console.WriteLine("\n┌──────────┬────────────────┬────────────┐");

Console.WriteLine("│ {0,-8} │ {1,-14} │ {2,-10} │", "Variable", "Special Chars", "Data Type");

Console.WriteLine("├──────────┼────────────────┼────────────┤");

foreach (var item in extracted)

{

Console.WriteLine("│ {0,-8} │ {1,-14} │ {2,-10} │",

item.VarName,

item.SpecialSymbol,

item.TokenType);

}

Console.WriteLine("└──────────┴────────────────┴────────────┘");

}

else

{

Console.WriteLine("\nNo valid variables found with special characters.");

Console.WriteLine("Example input: var a1 = 12@; float b2 = 3.14$; int c3 = 5#;");

}

}

}

