Question 2

```
using System;
using System.Collections.Generic;
using System.Text.RegularExpressions;

0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        Console.WriteLine("Enter your code (press Enter twice to finish):");
        string input = ReadMultilineInput();

        var variables = ExtractVariables(input);

        DisplayResults(variables);
    }

    1 reference
    static string ReadMultilineInput()
    {
        string input = "";
        string line;
        int emptyLineCount = 0;

        while ((line = Console.ReadLine()) != null)
        {
            if (string.IsNullOrWhiteSpace(line))
            {
                emptyLineCount++;
                if (emptyLineCount >= 1) break;
            }
            else
            {
                input += line + Environment.NewLine;
                emptyLineCount = 0;
            }
```

```
35          }
36
37          return input;
38      }
39
        1 reference
40      static List<VariableInfo> ExtractVariables(string input)
41      {
42          var variables = new List<VariableInfo>();
43
44
45          string pattern = @"\b([abc][a-zA-Z0-9_]*\d+)\s*=\s*([^;]+?[@#$%^&*\-+=].*?
46
47          var matches = Regex.Matches(input, pattern);
48
49          foreach (Match match in matches)
50          {
51              if (match.Groups.Count >= 3)
52              {
53                  string varName = match.Groups[1].Value;
54                  string value = match.Groups[2].Value;
55
56                  char specialSymbol = '\0';
57                  foreach (char c in value)
58                  {
59                      if (!char.IsLetterOrDigit(c) && !char.IsWhiteSpace(c))
60                      {
61                          specialSymbol = c;
62                          break;
63                      }
64                  }
65
66                  string tokenType = "Unknown";
67                  if (value.Contains("@")) tokenType = "Float";
68                  else if (value.Contains("#")) tokenType = "Integer";
69                  else if (value.Contains("$")) tokenType = "String";
70                  else if (value.Contains("%")) tokenType = "Percentage";
```

```
            else if (value.Contains("%")) tokenType = "Percentage";

            variables.Add(new VariableInfo
            {
                VarName = varName,
                SpecialSymbol = specialSymbol.ToString(),
                TokenType = tokenType
            });
        }
    }

    return variables;
}

1 reference
static void DisplayResults(List<VariableInfo> variables)
{
    if (variables.Count == 0)
    {
        Console.WriteLine("No matching variables found.");
        return;
    }

    int nameWidth = Math.Max("VarName".Length, GetMaxLength(variables, v => v.VarName));
    int symbolWidth = Math.Max("SpecialSymbol".Length, GetMaxLength(variables, v => v.SpecialSymbol));
    int typeWidth = Math.Max("TokenType".Length, GetMaxLength(variables, v => v.TokenType));

    Console.WriteLine();
    Console.WriteLine($"| {"VarName".PadRight(nameWidth)} | {"SpecialSymbol".PadRight(symbolWidth)} | {"TokenType".PadRight(typeWidth)} |");
    Console.WriteLine($"|{new string('-', nameWidth + 2)}|{new string('-', symbolWidth + 2)}|{new string('-', typeWidth + 2)}|");

    foreach (var variable in variables)
    {
        Console.WriteLine($"| {variable.VarName.PadRight(nameWidth)} | {variable.SpecialSymbol.PadRight(symbolWidth)} | {variable.TokenType.PadRight(typeWidth)} |");
    }
}
```

```
3 references
static int GetMaxLength(List<VariableInfo> variables, Func<VariableInfo, string> selector)
{
    int max = 0;
    foreach (var variable in variables)
    {
        int length = selector(variable).Length;
        if (length > max) max = length;
    }
    return max;
}


eferences
ass VariableInfo

    3 references
    public string VarName { get; set; }
    3 references
    public string SpecialSymbol { get; set; }
    3 references
    public string TokenType { get; set; }
```

Output

```
C:\WINDOWS\system32\cmd.exe
Enter your code (press Enter twice to finish):
a1 = test@email.com;
b2_value = 3.14#pi_constant;
c3 = "security$key";


| VarName | SpecialSymbol | TokenType |
|---------|---------------|-----------|
| a1      | @             | Float     |
| c3      | "             | String    |
Press any key to continue . . .
```