

```

% Define the folder path where your files are located
folderPath = 'C:\Users\hanib\Desktop\validationSetup\Validation Version
5\ValidFolder\data';

% Get a list of all CSV files in the folder
fileList = dir(fullfile(folderPath, 'scenario_*.csv'));

% Extract the scenario number from the file names and sort the list by this number
scenarioNumbers = zeros(length(fileList), 1); % Preallocate for scenario numbers

for i = 1:length(fileList)
    % Extract the number after "scenario_" and before the next "_"
    fileName = fileList(i).name;
    tokens = regexp(fileName, 'scenario_(\d+)', 'tokens'); % Use regular expression
    % to extract the number
    scenarioNumbers(i) = str2double(tokens{1}); % Convert the number to a double
end

% Sort the fileList based on the scenarioNumbers
[~, sortedIdx] = sort(scenarioNumbers);
fileList = fileList(sortedIdx); % Sort fileList using the indices

% Preallocate a matrix to store the final values for each scenario
finalValues = zeros(length(fileList), 2); % Columns for Gob_Temperature and
Mold_Temperature

% Loop through each file and extract the final non-NaN values
for i = 1:length(fileList)
    % Construct the full file path
    fileName = fileList(i).name;
    filePath = fullfile(folderPath, fileName);

    % Read the data from the CSV file
    data = readtable(filePath);

    % Check for the correct columns and find last non-NaN value manually
    if ismember('Gob_Temperature', data.Properties.VariableNames) && ...
        ismember('Mold_Temperature', data.Properties.VariableNames)

        % Loop from the end to find the last non-NaN Gob_Temperature and
        Mold_Temperature
        for j = height(data):-1:1
            if ~isnan(data.Gob_Temperature(j)) && ~isnan(data.Mold_Temperature(j))
                % Store the final non-NaN values in the matrix
                finalValues(i, 1) = data.Gob_Temperature(j);
                finalValues(i, 2) = data.Mold_Temperature(j);
                break; % Break loop after finding the last valid row
            end
        end
    else

```

```

        warning('Columns "Gob_Temperature" or "Mold_Temperature" not found in %s',
fileName);
    end
end

% Create a table for the results with scenario number
resultTable = array2table([(1:length(fileList))', finalValues], 'VariableNames',
{'Scenario', 'Gob_Temperature', 'Mold_Temperature'});

% Display the result table live
disp(resultTable);

```

Scenario	Gob_Temperature	Mold_Temperature
1	991.88	395.57
2	990.97	398.38
3	989.61	400.13
4	988.27	402.67
5	987.09	404.81
6	996.9	394.78
7	989.98	398.14
8	990.59	400.35
9	990.66	402.27
10	990.97	404.17
11	994.3	395.89
12	994.58	397.4
13	994.88	399.26
14	995.25	401.52
15	995.54	403.42
16	999.37	396.86
17	999.47	396.96
18	999.73	398.48
19	1000.4	401.08
20	1000.4	402.63
21	1004.7	397.82
22	1004.8	397.91
23	1004.9	398
24	1005.2	399.9
25	1005.8	402.14
26	1026.4	398.04
27	1027	400.61
28	1027.1	402.15
29	1027.5	404.43
30	1027.8	406.33
31	1031.1	397.28
32	1031.5	399.51
33	1032.1	401.72
34	1032.2	403.65
35	1032.5	405.56
36	1036.2	397.2
37	1036.4	398.71
38	1036.7	400.58
39	1037.1	402.85
40	1037.4	404.75
41	1041.6	398.11
42	1041.7	398.21
43	1042	399.73
44	1042.6	402.33
45	1042.6	403.89
46	1047.4	399

47	1047.5	399.1
48	1047.6	399.19
49	1047.9	401.09
50	1048.4	403.33
51	1067.5	399.47
52	1068.2	402.05
53	1068.2	403.59
54	1068.6	405.87
55	1068.9	407.78
56	1072.6	398.65
57	1073	400.89
58	1073.6	403.11
59	1073.7	405.04
60	1074	406.95
61	1078	398.51
62	1078.3	400.03
63	1078.6	401.9
64	1079	404.17
65	1079.3	406.08
66	1083.9	399.36
67	1084	399.47
68	1084.2	400.99
69	1084.9	403.6
70	1084.9	405.16
71	1090.1	400.19
72	1090.1	400.29
73	1090.2	400.38
74	1090.5	402.29
75	1091.1	404.53

```
% Write the results to a new Excel file called 'final_value.xlsx'
writetable(resultTable, fullfile(folderPath, 'final_value.xlsx'));
```

```
% Notify the user that the process is complete
disp('Final values have been extracted and saved to final_value.xlsx.');
```

Final values have been extracted and saved to final_value.xlsx.