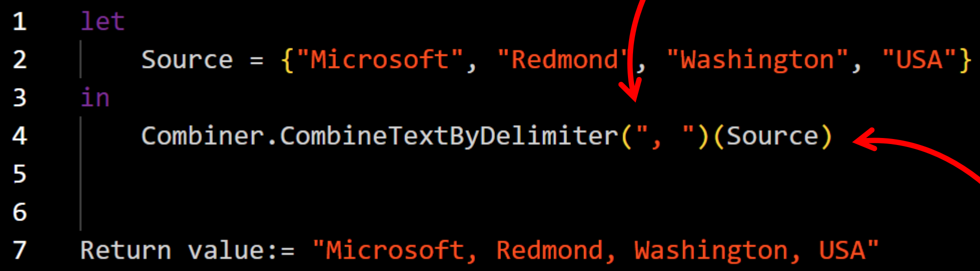
Power Query (M):

Exploring the Combiner functions

CombineTextByDelimiter

Probably the simplest of the combiner functions. We pass the delimiter into the function itself.

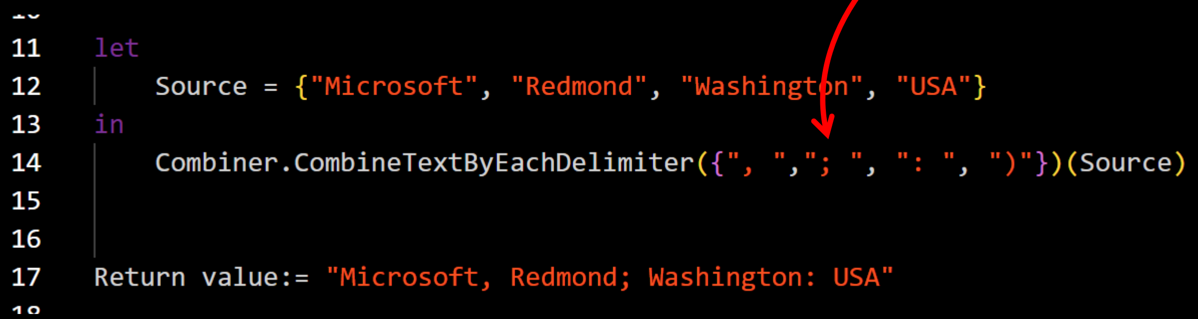


```
let
    Source = {"Microsoft", "Redmond", "Washington", "USA"}
in
    Combiner.CombineTextByDelimiter(", ")(Source)


Return value:= "Microsoft, Redmond, Washington, USA"
```

It returns a function of one parameter – a list of text.

CombineTextByEachDelimiter

We pass a list of delimiters into the Combiner function.



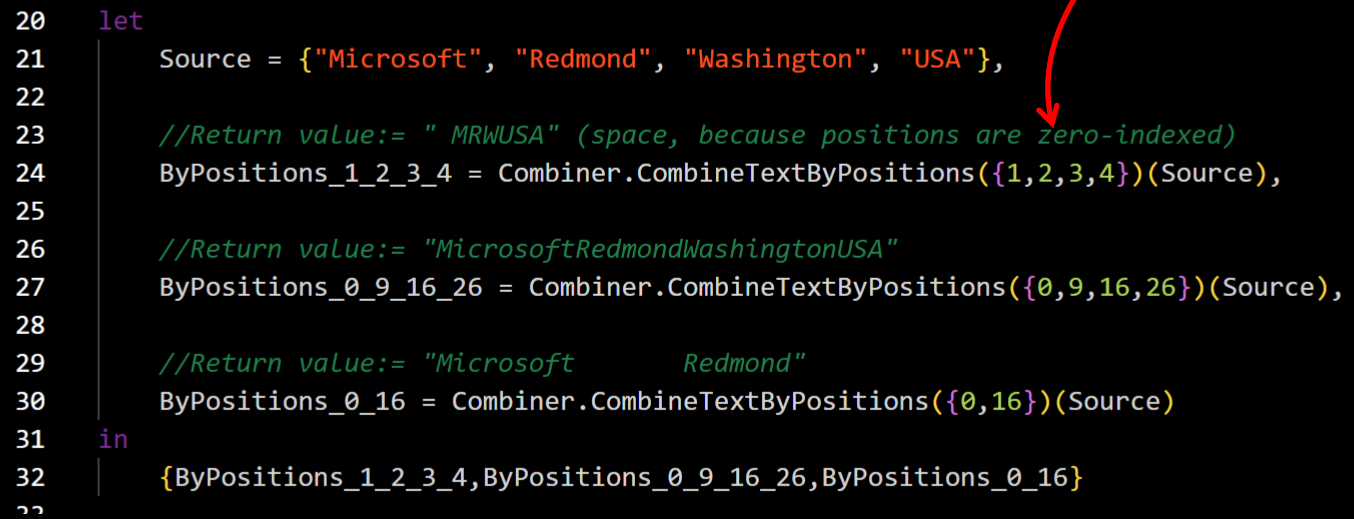
```
let
    Source = {"Microsoft", "Redmond", "Washington", "USA"}
in
    Combiner.CombineTextByEachDelimiter({", ","; ", ": ", ")"})(Source)


Return value:= "Microsoft, Redmond; Washington: USA"
```

Each delimiter in the list is used one by one in-turn to combine the elements of the text list.

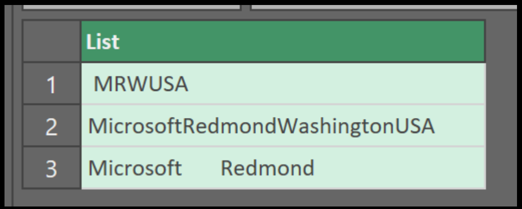
Surplus delimiters are ignored, like the close parenthesis above.

CombineTextByPositions

We pass a list of integer positions into the Combiner function.



```
let
    Source = {"Microsoft", "Redmond", "Washington", "USA"},

    //Return value:= " MRWUSA" (space, because positions are zero-indexed)
    ByPositions_1_2_3_4 = Combiner.CombineTextByPositions({1,2,3,4})(Source),

    //Return value:= "MicrosoftRedmondWashingtonUSA"
    ByPositions_0_9_16_26 = Combiner.CombineTextByPositions({0,9,16,26})(Source),

    //Return value:= "Microsoft       Redmond"
    ByPositions_0_16 = Combiner.CombineTextByPositions({0,16})(Source)
in
    {ByPositions_1_2_3_4,ByPositions_0_9_16_26,ByPositions_0_16}
```

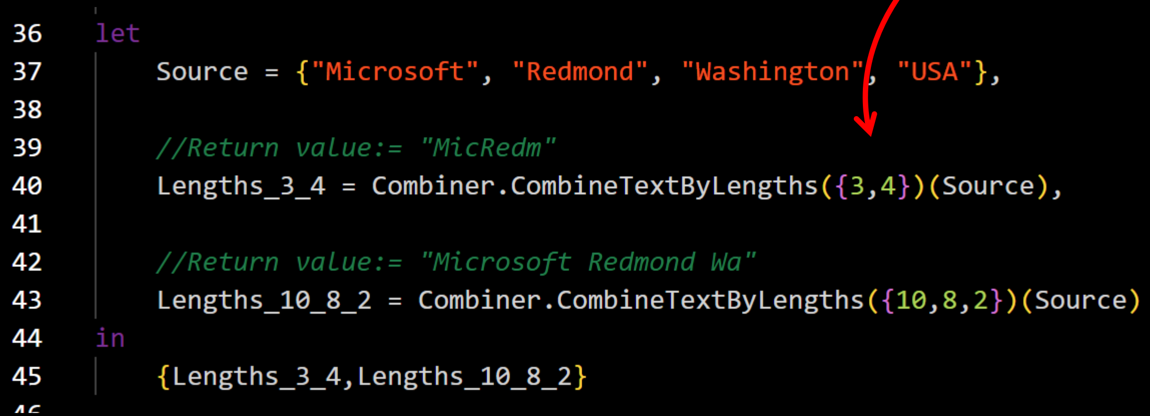


	List
1	MRWUSA
2	MicrosoftRedmondWashingtonUSA
3	Microsoft Redmond

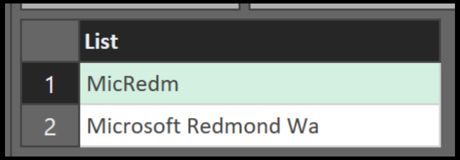
Each position in the positions list corresponds to the equivalent position in the text list. The integer in that position-list element represents the starting character position in the output text.

CombineTextByLengths

We pass a list of integer lengths into the Combiner function.



```
let
    Source = {"Microsoft", "Redmond", "Washington", "USA"},

    //Return value:= "MicRedm"
    Lengths_3_4 = Combiner.CombineTextByLengths({3,4})(Source),

    //Return value:= "Microsoft Redmond Wa"
    Lengths_10_8_2 = Combiner.CombineTextByLengths({10,8,2})(Source)
in
    {Lengths_3_4,Lengths_10_8_2}
```

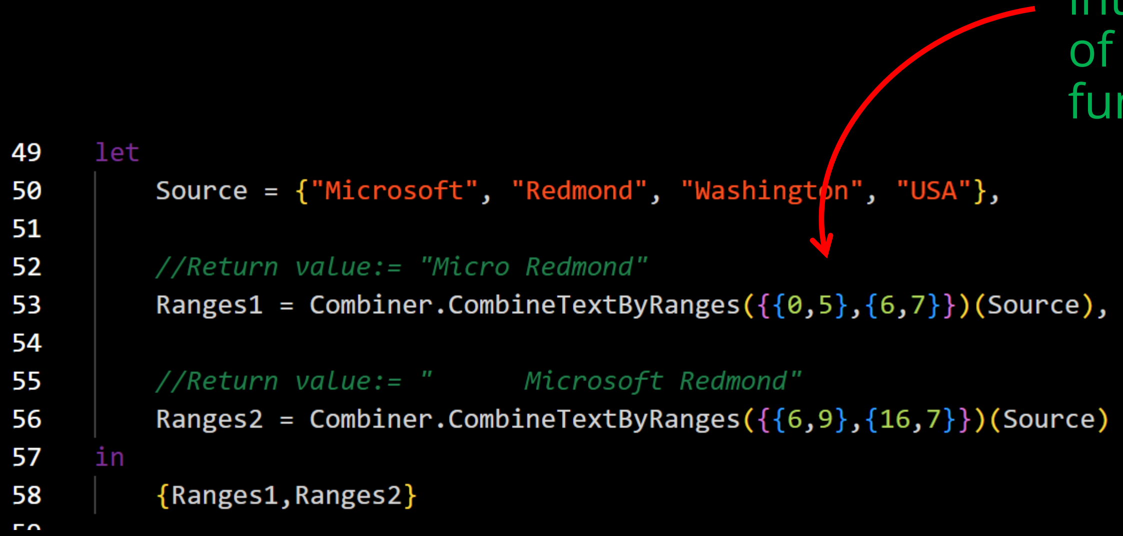


	List
1	MicRedm
2	Microsoft Redmond Wa

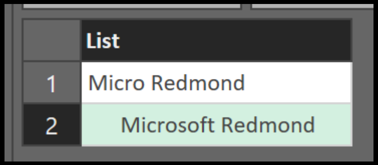
Each element in the lengths list corresponds to the equivalent element in the text list. The integer at that element represents the number of characters to retrieve from equivalent element in the text list. Lengths longer than their corresponding text are right-padded with spaces.

CombineTextByRanges

We pass a list of lists (of integers) into the parameter of the Combiner function



```
let
    Source = {"Microsoft", "Redmond", "Washington", "USA"},

    //Return value:= "Micro Redmond"
    Ranges1 = Combiner.CombineTextByRanges({{0,5},{6,7}})(Source),

    //Return value:= "      Microsoft Redmond"
    Ranges2 = Combiner.CombineTextByRanges({{6,9},{16,7}})(Source)
in
    {Ranges1,Ranges2}
```

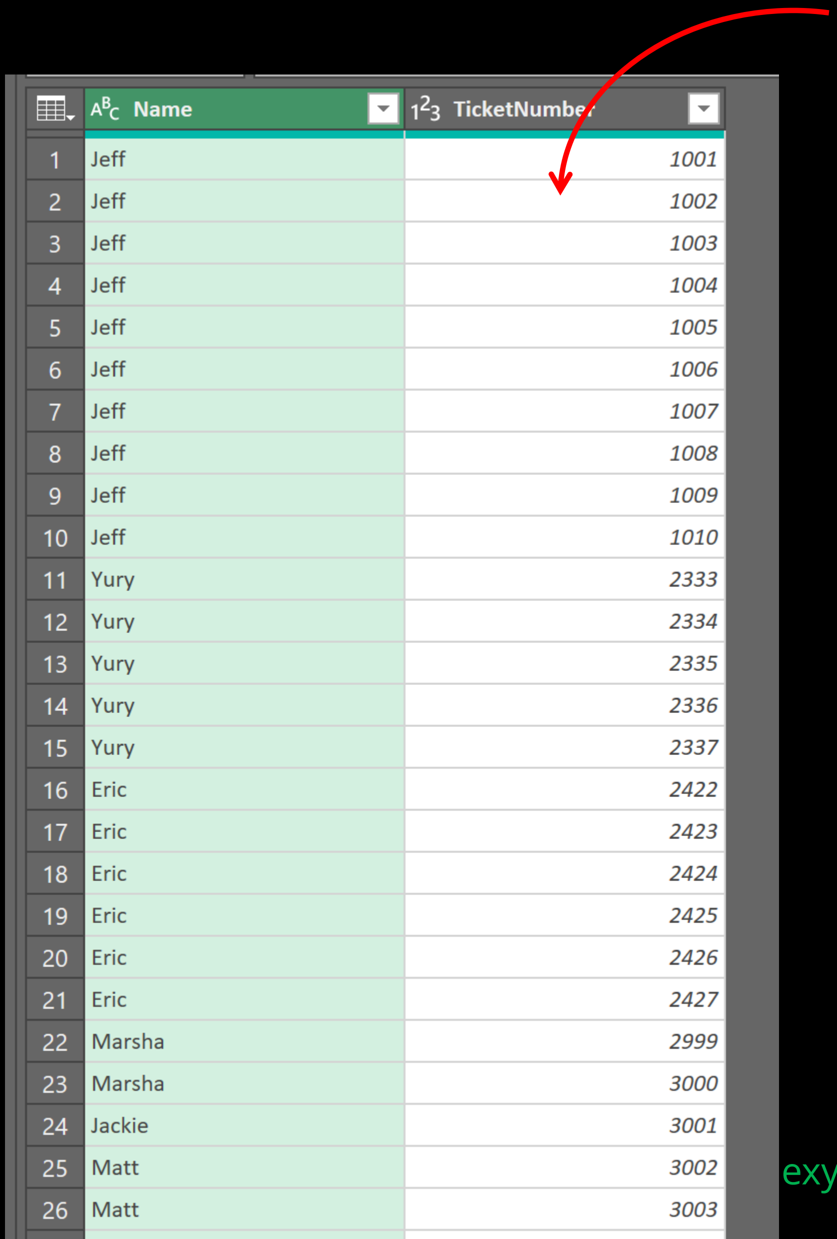


	List
1	Micro Redmond
2	Microsoft Redmond

Each sub-list is two integers. The first integer is the position to place the text in the output string. The second integer is the number of characters to retrieve from the text list element in the same position as the current sub-list. So, the first sub-list gets text from “Microsoft”, the second sub-list gets text from “Redmond”.

Example using CombineTextByEachDelimiter

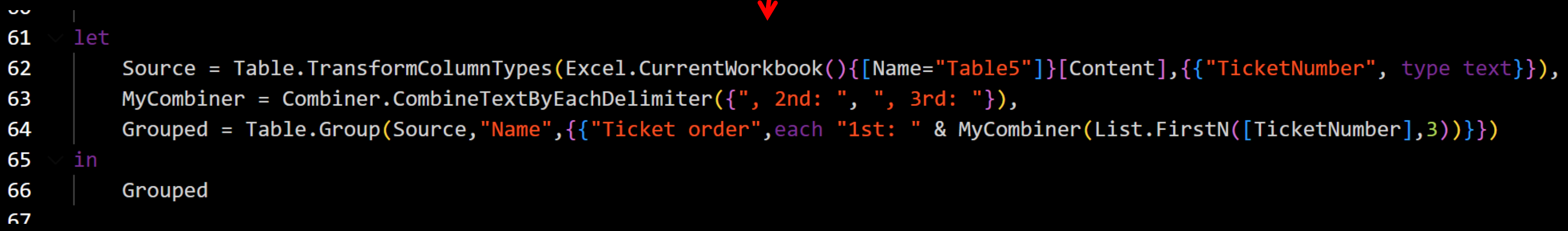
Suppose we have this table of names and ticket numbers



	Name	TicketNumber
1	Jeff	1001
2	Jeff	1002
3	Jeff	1003
4	Jeff	1004
5	Jeff	1005
6	Jeff	1006
7	Jeff	1007
8	Jeff	1008
9	Jeff	1009
10	Jeff	1010
11	Yury	2333
12	Yury	2334
13	Yury	2335
14	Yury	2336
15	Yury	2337
16	Eric	2422
17	Eric	2423
18	Eric	2424
19	Eric	2425
20	Eric	2426
21	Eric	2427
22	Marsha	2999
23	Marsha	3000
24	Jackie	3001
25	Matt	3002
26	Matt	3003

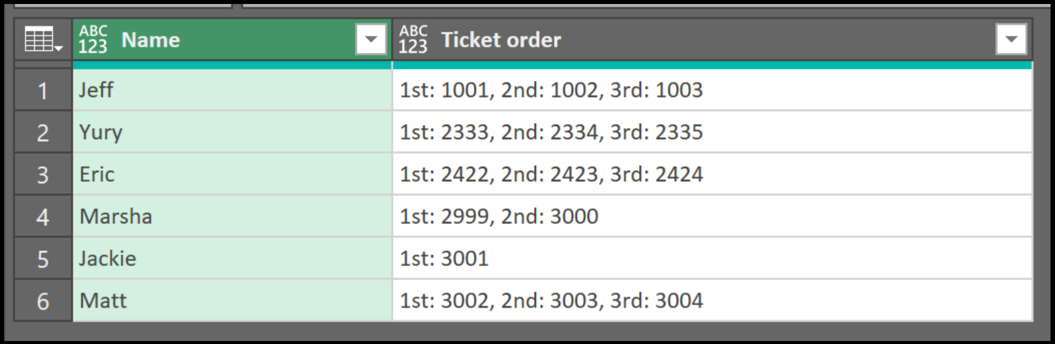
Example using CombineTextByEachDelimiter

We can create a custom Combiner



```
let
    Source = Table.TransformColumnTypes(Excel.CurrentWorkbook(){[Name="Table5"]}[Content],{{"TicketNumber", type text}}),
    MyCombiner = Combiner.CombineTextByEachDelimiter({", 2nd: ", ", 3rd: "}),
    Grouped = Table.Group(Source,"Name",{{"Ticket order",each "1st: " & MyCombiner(List.FirstN([TicketNumber],3))}})
in
    Grouped
```

And use it in Table.Group to create an ordinal list



	Name	Ticket order
1	Jeff	1st: 1001, 2nd: 1002, 3rd: 1003
2	Yury	1st: 2333, 2nd: 2334, 3rd: 2335
3	Eric	1st: 2422, 2nd: 2423, 3rd: 2424
4	Marsha	1st: 2999, 2nd: 3000
5	Jackie	1st: 3001
6	Matt	1st: 3002, 2nd: 3003, 3rd: 3004