In [30]:

```
import numpy as np
import pandas as pd
```

In [34]:

```
# Loading Data from a CSV File
data = read_csv(r"C:\Users\Admin\Desktop\data.csv")
print(data)
      Time Weather Temperature       Company Humidity           Wind Goes
0  Morning   Sunny         Warm           Yes     Mild         Strong  Yes
1  Evening   Rainy         Cold            No     Mild         Normal   No
2  Morning   Sunny    Moderate           Yes   Normal         Normal  Yes
3  Evening   Sunny         Cold           Yes     High         Strong  Yes
```

In [35]:

```
concepts = np.array(data.iloc[:,0:-1])
print(concepts)
[['Morning' 'Sunny' 'Warm' '         Yes' 'Mild' '       Strong']
 ['Evening' 'Rainy' 'Cold' '         No' 'Mild' '       Normal']
 ['Morning' 'Sunny' 'Moderate' ' Yes' 'Normal' '       Normal']
 ['Evening' 'Sunny' 'Cold' '         Yes' 'High' '       Strong']]
```

In [36]:

```
target = np.array(data.iloc[:,-1])
print(target)
['Yes' 'No' 'Yes' 'Yes']
```

In [37]:

```
def learn(concepts, target):

    '''
    learn() function implements the learning method of the Candidate
elimination algorithm.
    Arguments:
        concepts - a data frame with all the features
        target - a data frame with corresponding output values
    '''

    # Initialise S0 with the first instance from concepts
    # .copy() makes sure a new list is created instead of just pointing to
the same memory location
    specific_h = concepts[0].copy()
    print("\nInitialization of specific_h and general_h")
    print(specific_h)
    #h=["#" for i in range(0,5)]
    #print(h)

    general_h = [["?" for i in range(len(specific_h))] for i in
range(len(specific_h))]
    print(general_h)
    # The learning iterations
    for i, h in enumerate(concepts):

        # Checking if the hypothesis has a positive target
        if target[i] == "Yes":
            for x in range(len(specific_h)):

                # Change values in S & G only if values change
                if h[x] != specific_h[x]:
                    specific_h[x] = '?'
                    general_h[x][x] = '?'

        # Checking if the hypothesis has a positive target
        if target[i] == "No":
            for x in range(len(specific_h)):
                # For negative hyposthesis change values only  in G
                if h[x] != specific_h[x]:
                    general_h[x][x] = specific_h[x]
                else:
                    general_h[x][x] = '?'

        print("\nSteps of Candidate Elimination Algorithm",i+1)
        print(specific_h)
        print(general_h)

    # find indices where we have empty rows, meaning those that are unchanged
    indices = [i for i, val in enumerate(general_h) if val == ['?', '?', '?',
'?', '?', '?']]
    for i in indices:
        # remove those rows from general_h
        general_h.remove(['?', '?', '?', '?', '?', '?'])
    # Return final values
    return specific_h, general_h
```

In [38]:

```
s_final, g_final = learn(concepts, target)
print("\nFinal Specific_h:", s_final, sep="\n")
print("\nFinal General_h:", g_final, sep="\n")
```

```
Initialization of specific_h and general_h
['Morning' 'Sunny' 'Warm' '         Yes' 'Mild' '      Strong']
[['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?',
'?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?',
'?', '?'], ['?', '?', '?', '?', '?', '?']]

Steps of Candidate Elimination Algorithm 1
['Morning' 'Sunny' 'Warm' '         Yes' 'Mild' '      Strong']
```

```
[['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?',
'?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?',
'?', '?'], ['?', '?', '?', '?', '?', '?']]

Steps of Candidate Elimination Algorithm 2
['Morning' 'Sunny' 'Warm' '         Yes' 'Mild' '       Strong']
[['Morning', '?', '?', '?', '?', '?'], ['?', 'Sunny', '?', '?', '?', '?'],
['?', '?', 'Warm', '?', '?', '?'], ['?', '?', '?', '         Yes', '?', '?'],
['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '       Strong']]

Steps of Candidate Elimination Algorithm 3
['Morning' 'Sunny' '?' '?' '?' '?']
[['Morning', '?', '?', '?', '?', '?'], ['?', 'Sunny', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?',
'?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?']]

Steps of Candidate Elimination Algorithm 4
['?' 'Sunny' '?' '?' '?' '?']
[['?', '?', '?', '?', '?', '?'], ['?', 'Sunny', '?', '?', '?', '?'], ['?',
'?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?',
'?', '?', '?'], ['?', '?', '?', '?', '?', '?']]

Final Specific_h:
['?' 'Sunny' '?' '?' '?' '?']

Final General_h:
[['?', 'Sunny', '?', '?', '?', '?']]
```

In [ ]: