

EX 2

CODING:

```
data = [  
    ['Sunny','Warm','Yes'],  
    ['Sunny','Cold','Yes'],  
    ['Rainy','Warm','No']  
]
```

```
S = None
```

```
G = ['?','?']
```

```
for d in data:
```

```
    x, y = d[:-1], d[-1]
```

```
    if y == 'Yes':
```

```
        if S is None: S = x
```

```
    else:
```

```
        for i in range(len(S)):
```

```
            if S[i] != x[i]: S[i] = '?'
```

```
    else:
```

```
        for i in range(len(G)):
```

```
            if G[i] == '?' and S[i] != x[i]:
```

```
                G[i] = S[i]
```

```
print("S =", S)
```

```
print("G =", G)
```

OUTPUT:

The screenshot shows a code editor interface with a Python script named `main.py`. The code implements a function to calculate the most likely weather conditions based on historical data. It uses sets `S` and `G` to store possible values for sunny and rainy days respectively, and iterates through the data to update these sets based on observed outcomes.

```
1+ data = [
2+     ['Sunny', 'Warm', 'Yes'],
3+     ['Sunny', 'Cold', 'Yes'],
4+     ['Rainy', 'Warm', 'No']
5+ ]
6+
7 S = None
8 G = ['?', '?']
9
10 for d in data:
11     x, y = d[:-1], d[-1]
12
13+     if y == 'Yes':
14+         if S is None: S = x
15+         else:
16+             for i in range(len(S)):
17+                 if S[i] != x[i]: S[i] = '?'
18+     else:
19+         for i in range(len(G)):
20+             if G[i] == '?' and S[i] != x[i]:
21+                 G[i] = S[i]
22
23 print("S =", S)
24 print("G =", G)
25
```

The output window shows the execution results:

```
S = ['Sunny', '?']
G = ['Sunny', '?']

== Code Execution Successful ==
```