1	**Aim**	Illustrate and demonstrate the working model and principle of Find-S algorithm
	Program	For a given set of training data examples stored in a .CSV file, implement and demonstrate the Find-S algorithm to output a description of the set of all hypotheses consistent with the training examples

CONCEPT - FIND-S: FINDING A MAXIMALLY SPECIFIC HYPOTHESIS

FIND-S Algorithm

1. Initialize *h* to the most specific hypothesis in *H*
2. For each positive training instance *x*
    For each attribute constraint a_i in *h*
        If the constraint a_i is satisfied by *x*
            Then do nothing
        Else replace a_i in *h* by the next more general constraint that is satisfied by *x*
3. Output hypothesis *h*

To illustrate this algorithm, assume the learner is given the sequence of training examples from the ***EnjoySport*** task

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

- The first step of FIND-S is to initialize h to the most specific hypothesis in H

h - (Ø, Ø, Ø, Ø, Ø, Ø)

- Consider the first training example

x1 = <Sunny, Warm, Normal, Strong, Warm, Same>, +

Observing the first training example, it is clear that hypothesis *h* is too specific. None of the "Ø" constraints in h are satisfied by this example, so each is replaced by the next *more general constraint* that fits the example

h1 = <Sunny, Warm, Normal, Strong, Warm, Same>

- Consider the second training example

x2 = <Sunny, Warm, High, Strong, Warm, Same>, +

The second training example forces the algorithm to further generalize h, this time substituting a "?" in place of any attribute value in h that is not satisfied by the new example

h2 = <Sunny, Warm, ?, Strong, Warm, Same>

- Consider the third training example

x3 = <Rainy, Cold, High, Strong, Warm, Change>, -

Upon encountering the third training the algorithm makes no change to h. The FIND-S algorithm simply ignores every negative example.

h3 = < Sunny, Warm, ?, Strong, Warm, Same>

- Consider the fourth training example

x4 = <Sunny Warm High Strong Cool Change>, +

The fourth example leads to a further generalization of h

h4 = < Sunny, Warm, ?, Strong, ?, ? >

<u>The key property of the FIND-S algorithm</u>

- It is incremental learning i.e., algorithm learns by processing one training example at a time, updating its hypothesis based on each example
- FIND-S is computationally efficient, especially for small to medium-sized datasets and can handle noisy data and incomplete training sets
- FIND-S is guaranteed to output the most specific hypothesis within H that is consistent with the positive training examples
- FIND-S algorithm’s final hypothesis will also be consistent with the negative examples provided the correct target concept is contained in H, and provided the training examples are correct.

Training Instances: (The below data is saved as *enjoysport.csv* file)

Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

Program:

```
import csv
with open('enjoysport.csv', 'r') as file:
    data = [row for row in csv.reader(file)]
    print("The total number of training instances are:",len(data)-1,'\n',data[1:])

num_attribute = len(data[0])-1

# Initial hypothesis
hypothesis = ['0']*num_attribute

for i in range(0, len(data)):
    if data[i][num_attribute] == 'yes':
        for j in range(0, num_attribute):
            if hypothesis[j] == '0' or hypothesis[j] == data[i][j]:
                hypothesis[j] = data[i][j]
            else:
                hypothesis[j] = '?'
    print("\n The hypothesis for the training instance {} is : \n".format(i),hypothesis)

print("\n The Maximally specific hypothesis for the training instances is ", hypothesis)
```

Output:

```
The total number of training instances are: 4
['sunny', 'warm', 'normal', 'strong', 'warm', 'same', 'yes']
['sunny', 'warm', 'high', 'strong', 'warm', 'same', 'yes']
['rainy', 'cold', 'high', 'strong', 'warm', 'change', 'no']
['sunny', 'warm', 'high', 'strong', 'cool', 'change', 'yes']

The hypothesis for the training instance 0 is:
 ['0', '0', '0', '0', '0', '0']

The hypothesis for the training instance 1 is:
 ['sunny', 'warm', 'normal', 'strong', 'warm', 'same']

The hypothesis for the training instance 2 is:
 ['sunny', 'warm', '?', 'strong', 'warm', 'same']

The hypothesis for the training instance 3 is:
 ['sunny', 'warm', '?', 'strong', 'warm', 'same']

The hypothesis for the training instance 4 is:
 ['sunny', 'warm', '?', 'strong', '?', '?']

The Maximally specific hypothesis for the training instance is
['sunny', 'warm', '?', 'strong', '?', '?']
```