COURSE LABORATORY MANUAL

1. EXPERIMENT NO: 1

2. TITLE: **FIND-S ALGORITHM**

3. LEARNING OBJECTIVES:

- Make use of Data sets in implementing the machine learning algorithms.
- Implement ML concepts and algorithms in Python

4. AIM:

- Implement and demonstratethe FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.

5. THEORY:

- The concept learning approach in machine learning, can be formulated as “Problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples”.
- Find-S algorithm for concept learning is one of the most basic algorithms of machine learning.

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 in h 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
```

- It is Guaranteed to output the most specific hypothesis within H that is consistent with the positive training examples.
- Also Notice that negative examples are ignored.

Limitations of the Find-S algorithm:

- No way to determine if the only final hypothesis (found by Find-S) is consistent with data or there are more hypothesis that is consistent with data.
- Inconsistent sets of training data can mislead the finds algorithm as it ignores negative data samples.
- A good concept learning algorithm should be able to backtrack the choice of hypothesis found so that the resulting hypothesis can be improved over time. Unfortunately, Find-S provide no such method.

6. PROCEDURE / PROGRAMME :

FindS.py

```
import csv
def read_data(filename):
    with open(filename, 'r') as csvfile:
        datareader = csv.reader(csvfile, delimiter=',')
        headers = next(datareader)
        traindata = []
        for row in datareader:
            traindata.append(row)

    return (traindata,headers)
```

```
def print_hypothesis(h):
    print('<',end=' ')
    for i in range(0,len(h)-1):
        print(h[i],end=',')
    print('>')

def findS():
    dataset,features=read_data('data11_tennis6.csv')
    rows=len(dataset);
    cols=len(dataset[0]);

    flag = 0
    for x in range(0,rows):
        t=dataset[x]

        # Initialize h with first +ve sample
        if t[-1]=='1' and flag==0:
            flag=1
            h = dataset[x]
        # Update h with remaining +ve samples
        elif t[-1]=='1':
            for y in range(cols):
                if h[y]!=t[y]:
                    h[y]='?'
        #print("For Training instance {0} the hypothesis is :".format(x+1),end='  ')
        #print_hypothesis(h)

    print("The maximally specific Hypothesis for a given Training Examples")
    #print(h)
    print_hypothesis(h)

findS()
```

7. RESULTS & CONCLUSIONS:

Result-1

Dataset: data11_tennis6.csv

```
Sky,AirTemp,Humidity,Wind,EnjoySport
sunny,warm,normal,strong,warm,same,1
sunny,warm,high,strong,warm,same,1
rainy,cold,high,strong,warm,change,0
sunny,warm,high,strong,cool,change,1
```

Output:

```
    The Maximally Specific Hypothesis for a given Training Examples
    < sunny,warm,?,strong,?,?,>
```

Result-2

Dataset: data12_tennis4.csv

```
Sky,AirTemp,Humidity,Wind,EnjoySport
sunny,hot,high,weak,1
sunny,hot,high,strong,1
overcast,hot,high,weak,1
rain,mild,high,weak,0
rain,cool,normal,weak,1
rain,cool,normal,strong,0
overcast,cool,normal,strong,1
sunny,cool,normal,weak,1
rain,mild,normal,weak,1
sunny,mild,normal,strong,1
overcast,mild,high,strong,1
overcast,hot,normal,weak,1
rain,mild,high,strong,0
```

Output

```
The Maximally Specific Hypothesis for a given Training Examples
< ?,?,?,?,>
```

8. LEARNING OUTCOMES :

- Students will be able to apply Find-S algorithm to the real world problem and find the most specific hyposis from the training data.

9. APPLICATION AREAS:

- Classification based problems.

10. REMARKS:

-