EX. NO: 01	**FIND S ALGORITHM**
DATE:	

AIM:

To implement the Find-S concept learning algorithm on the Student Performance dataset to determine the most specific hypothesis that classifies students with good performance.

PROCEDURE:

STEP 1: Open Google Colab and create a new notebook
STEP 2: Upload the StudentPerformance.csv dataset
STEP 3: Load dataset using pandas
STEP 4: Display dataset head and verify column names
STEP 5: Convert numeric attributes into categorical values
STEP 6: Convert Performance Index into binary class (Good/Poor)
STEP 7: Select only positive (Good) examples
STEP 8: Take first few positive samples to reduce variation
STEP 9: Initialize hypothesis with first positive example
STEP 10: Compare with remaining positives and update using Find-S rule
STEP 11: Print the final hypothesis

DATASET DESCRIPTION:

The Student Performance dataset contains 10,001 student records with the following attributes:

- Hours Studied - number of hours studied
- Previous Scores - previous exam marks
- Extracurricular Activities -Yes/No participation
- Sleep Hours - daily sleep duration
- Sample Question Papers Practiced - number of papers solved
- Performance Index - final performance score

Since Find-S requires categorical attributes and a binary target:

- Numeric attributes were converted into categorical ranges
- Performance Index was converted into class label:
    - ≥ 80 = Good
    - < 80 = Poor

To avoid over-generalization, a small subset of positive examples was used for hypothesis learning.

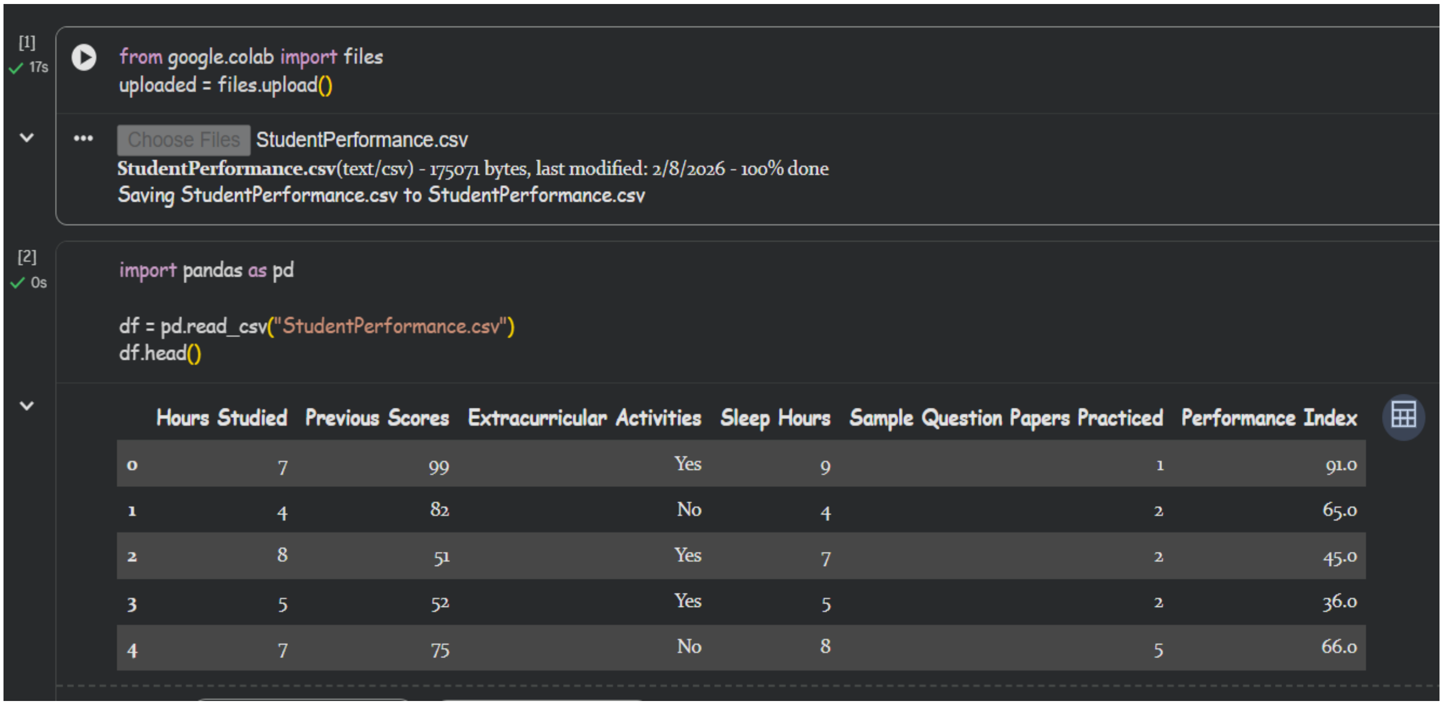
PROGRAM:

```
from google.colab import files
uploaded = files.upload()

import pandas as pd
df = pd.read_csv("StudentPerformance.csv")

# --- Convert numeric columns to categorical ---
def hours_cat(x):
    return "High" if x >= 5 else "Low"
def score_cat(x):
    return "Strong" if x >= 75 else "Weak"
def sleep_cat(x):
    return "Adequate" if x >= 6 else "Less"
def practice_cat(x):
    return "Practice" if x >= 3 else "NoPractice"
df["Hours"] = df["Hours Studied"].apply(hours_cat)
df["Prev"] = df["Previous Scores"].apply(score_cat)
df["Sleep"] = df["Sleep Hours"].apply(sleep_cat)
df["Practice"] = df["Sample Question Papers Practiced"].apply(practice_cat)
# --- Create binary target ---
df["Target"] = df["Performance Index"].apply(
    lambda x: "Good" if x >= 80 else "Poor"
)

# --- Find-S Algorithm ---
features = ["Hours","Prev","Extracurricular Activities","Sleep","Practice"]
positive_data = df[df["Target"] == "Good"][features].values[:10]
```

```
hypothesis = list(positive_data[0])
for instance in positive_data:
    for i in range(len(hypothesis)):
        if hypothesis[i] != instance[i]:
            hypothesis[i] = "?"
print("Final Hypothesis:", hypothesis)
```

OUTPUT SCREENSHOTS:

DATASET UPLOAD IN COLAB:



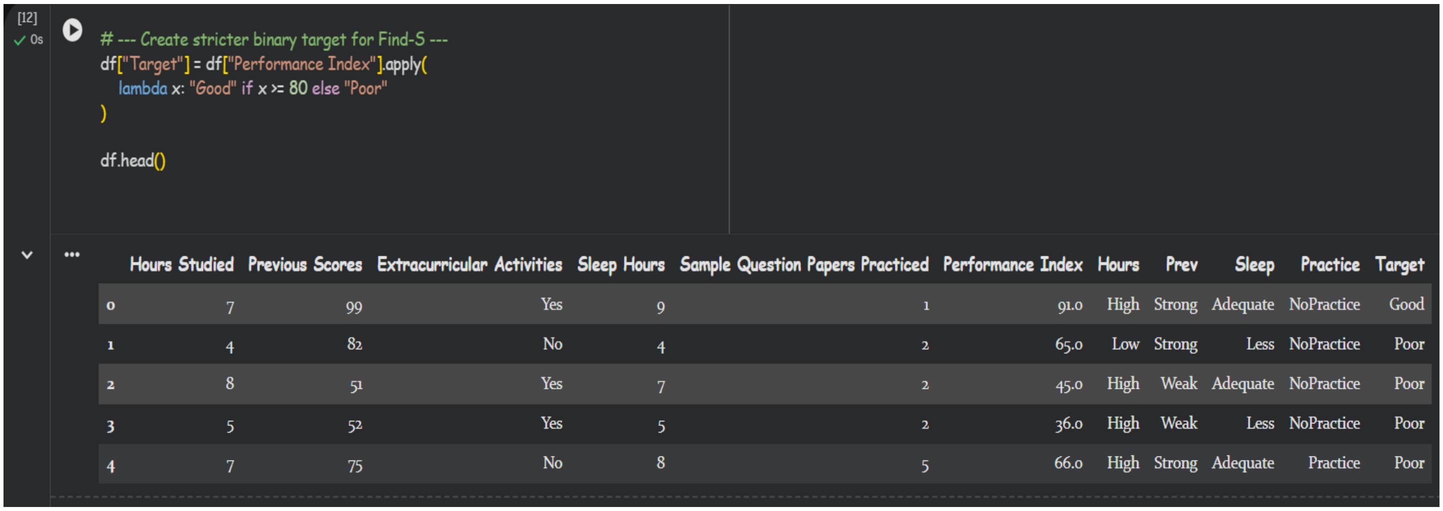
```
[1]
from google.colab import files
uploaded = files.upload()
```

```
StudentPerformance.csv
StudentPerformance.csv(text/csv) - 175071 bytes, last modified: 2/8/2026 - 100% done
Saving StudentPerformance.csv to StudentPerformance.csv
```

```
[2]
import pandas as pd

df = pd.read_csv("StudentPerformance.csv")
df.head()
```

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	Yes	9	1	91.0
1	4	82	No	4	2	65.0
2	8	51	Yes	7	2	45.0
3	5	52	Yes	5	2	36.0
4	7	75	No	8	5	66.0

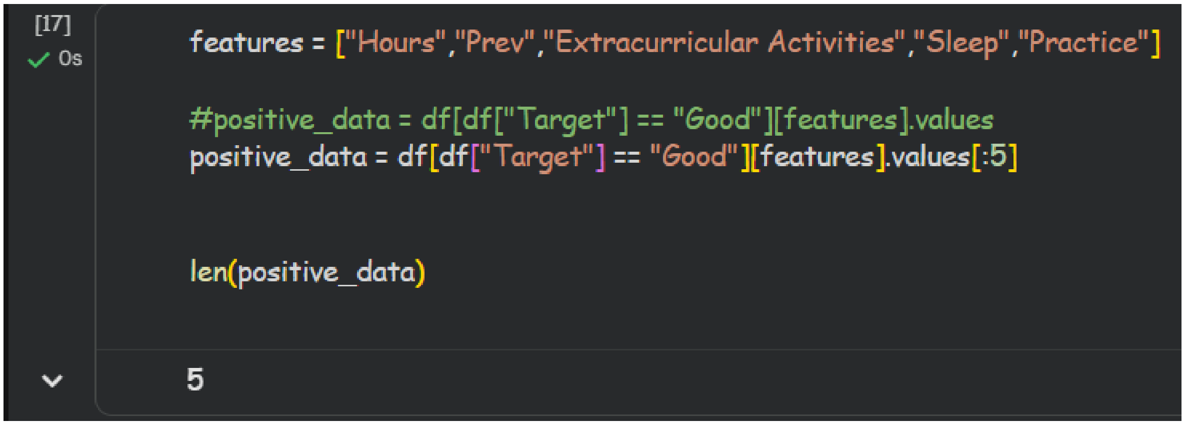
PREPROCESSED DATA:



```
[12]
# --- Create stricter binary target for Find-S ---
df["Target"] = df["Performance Index"].apply(
    lambda x: "Good" if x >= 80 else "Poor"
)

df.head()
```

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index	Hours	Prev	Sleep	Practice	Target
0	7	99	Yes	9	1	91.0	High	Strong	Adequate	NoPractice	Good
1	4	82	No	4	2	65.0	Low	Strong	Less	NoPractice	Poor
2	8	51	Yes	7	2	45.0	High	Weak	Adequate	NoPractice	Poor
3	5	52	Yes	5	2	36.0	High	Weak	Less	NoPractice	Poor
4	7	75	No	8	5	66.0	High	Strong	Adequate	Practice	Poor

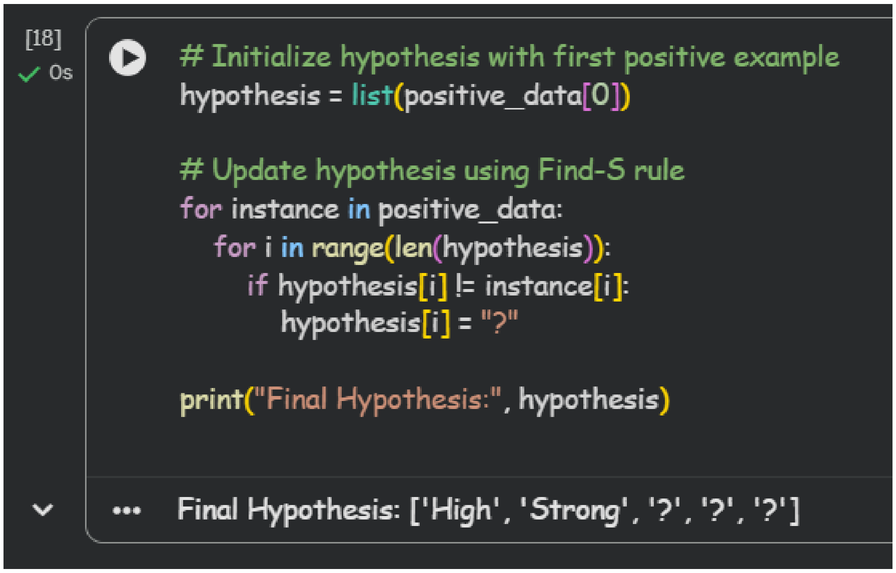
POSITIVE SAMPLE SELECTION:



```
features = ["Hours","Prev","Extracurricular Activities","Sleep","Practice"]

#positive_data = df[df["Target"] == "Good"][features].values
positive_data = df[df["Target"] == "Good"][features].values[:5]


len(positive_data)
```

```
5
```

FINAL OUTPUT:



```
# Initialize hypothesis with first positive example
hypothesis = list(positive_data[0])

# Update hypothesis using Find-S rule
for instance in positive_data:
    for i in range(len(hypothesis)):
        if hypothesis[i] != instance[i]:
            hypothesis[i] = "?"

print("Final Hypothesis:", hypothesis)
```

```
Final Hypothesis: ['High', 'Strong', '?', '?', '?']
```

<u>RESULT:</u>

The Find-S algorithm was successfully implemented on the Student Performance dataset. After preprocessing and selecting positive training examples, the algorithm generated the most specific hypothesis:

['High', 'Strong', '?', '?', '?']

This indicates that high study hours and strong previous scores are consistently present among good-performance students, while other attributes vary. The experiment confirms the working of the Find-S concept learning method.