3 Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate dataset for building the decision tree and apply this knowledge to classify a new sample.

```
import pandas as pd
from pandas import DataFrame
df_tennis = pd.read_csv('C:/Users/hp/Desktop/4MT17CS044_JOVITA/
                        PlayTennis.csv')
attribute_names = list(df_tennis.columns)
attribute_names.remove('Play Tennis')
print(attribute_names)
def entropy_of_list(list):
    from collections import counter
    count = counter(x for x in lst)
    num_instances = len(lst) * 1
    probs = [x/num_instances for x in count.values()]
    return entropy(probs)
def entropy(probs):
    import math
    return sum([-prob * math.log(prob, 2) for prob in probs])

total_entropy = entropy_of_list(df_tennis['Play Tennis'])

def information_gain(df, split_attribute_name, target_attribute_name, trace=0):
    df_split = df.groupby(split_attribute_name)
    nobs = len(df.index)+1
    df_agg_ent = df_split.agg({target_attribute_name : [entropy_of_list,
                              lambda x : len(x)/nobs]})
```

Teacher's Signature _____

```
df_agg_ent.columns = ['Entropy', 'proportionvalues']
new_entropy = sum[df.agg_ent['Entropy'] *df_agg_ent['proportionvalues'])
old_entropy = entropy_of_list(df[target_attribute_name])
print(split attribute_name, 'IG:', old_entropy - new entropy)
return old_entropy - new_entropy

def id3(df, target_attribute_name, attribute_names, default_class = None):
  from collections import counter
  count = Counter(x for x in df[target_attribute_name])
  if len(count) == 1:
     return next[iter(count)]
  elif df.empty or (not attribute_names):
     return default_class
  else:
     default_class = max(count.keys())
     gain = [information_gain(df, attr, target_attribute_name) for
             attr in attribute_names]
     index_of_max = gain.index[max(gain)]
     best_attr = attribute_names[index_of_max]
     tree = {best_attr : {}}

     remaining_attribute_names = [i for i in attribute_names if i!=best_attr]

for attr_val, data_subset in df.groupby(best_attr):
     subtree = id3(data_subset, target_attribute_name, remaining_attribute_
             names, default_class)
     tree[best_attr][attr_val] = subtree
return tree
```

Teacher's Signature

```
from pprint import pprint
tree = id3(df_tennis, 'Play_Tennis', attribute_names)
print("\n The Resultant decision tree is :\n")
pprint(tree)
```

Teacher's Signature

Output

['outlook', 'temperature', 'humidity', 'wind']

outlook IG: 0.24674981977443911

Temperature IG: 0.029922565895 4647

Humidity IG: 0.15183550136239136

Wind IG: 0.048127030408 26927

Temperature IG: 0.019973094021 97489

Humidity IG: 0.019973094021 97489

Wind IG: 0.970950599454 6686

Temperature IG: 0.570950594459 6686

Humidity IG: 0.970950594454 6686

Wind IG: 0.019973094021 97489

The resultant decision tree is:

{'outlook': {'overcast': 'yes',
'Rain': {'wind': {'strong': 'NO', 'weak': 'Yes'}},
'Sunny': {'humidity': {'high': 'NO', 'Normal': 'Yes'}}}}