1. Use Python to implement a decision tree using ID3 algorithm. For the data, see the link below.

   https://archive.ics.uci.edu/ml/datasets/lenses

2. The attached dataset is divided into three sets: the training set, the validation set and the test set. Columns A to T show features (XB to XU) and column U shows class (1 or 0). It is a binary tree (meaning features take only two values 1 or 0). Implement the decision tree algorithm (ID3) using information gain to select attributes to split. The Python code must output the accuracy on the test set.

   Also implement the post pruning algorithm (shown below). Report the accuracies for the post pruned-decision trees for 10 combinations of L and K values.

```
Input: An integer L and an integer K
Output: A post-pruned Decision Tree
begin
    Build a decision tree using all the training data. Call it D;
    Let D_Best = D;
    for i = 1 to L do
        Copy the tree D into a new tree D';
        M = a random number between 1 and K;
        for j = 1 to M do
            Let N denote the number of non-leaf nodes in the decision
             tree D'. Order the nodes in D' from 1 to N;
            P = a random number between 1 and N;
            Replace the subtree rooted at P in D' by a leaf node.
             Assign the majority class of the subset of the data at P to
             the leaf node.;
            /* For instance, if the subset of the data at P
               contains 10 examples with class = 0 and 15
               examples with class = 1, replace P by class = 1
               */
        end
        Evaluate the accuracy of D' on the validation set;
        /* accuracy = percentage of correctly classified
           examples                                          */
        if D' is more accurate than D_Best then
            D_Best = D';
        end
    end
    return D_Best;
end
```