a) [25 marks] Present a pseudo-code for a simple Decision Tree with error reduction pruning:

----

i) The information gain is used as a split criterion. [5 marks]

```
info_gain(data, split_attribute, target_attribute):
    entropy_1 := calculate entropy of target attribute in dataset (data)
    elements := get unique elements of the split attribute from dataset(data)

    for i := 0; i < size(elements); i++:
        target_data_rows := get the rows of the target attribute from dataset(data)
        entropy_2 := calculate the entropy based on entropy of target data and element
count

    gain = entropy_1 - entropy_2
    return gain
```

----

ii) The tree is grown deep, i.e. it is grown until all training examples corresponding to

a leaf node belong to the same class. [5 marks]

no answer

iii) Works on categorical data. [5 marks]

```
get_clean_data():
    dataset = read_csv(path)

    dataset.remove_nulls()
    dataset.remove_weird_symbols()

    columns = [income, marial-status]
    dataset.rename_columns()
    columns = [income, marial-status]

    dataset.encode_categorical_columns(columns)
```

iv) Works on numerical data. [5 marks]

```
get_clean_data():

    columns = [age, education-num, capital-gain, capital-loss, hours-per-week, marital-
status]

    // convert columns from string to type int

    data[age] = data[age].astype(int)

    // do the same as above for other columns
```

v) Error reduction pruning using validation data. [5 marks]

```
prune(data, best_idx, best_attr, attributes, target_attr, parent_attr):

    unique_vals := get unique values for the best attribute in the dataset


    if count(unique_vals[0]) == size(unique_vals):

        return unique_vals[0]   // return first element

    else:

        tree := dict()      // create object


        for v in unique_vals:

            new_data := get data for the best attribute from the dataset(data)

            new_attributes := remove previous best attribute from the attribute list


            // recursively build the subtree and connect with the tree

            subtree = grow(new_data, new_attributes, target_attr)

            tree[best_attr][v] := subtree


    return tree
```