Part 1:

a) Report the predicted class labels of each instance in the test set using the kNN with k=1, and the classification accuracy on the test set of kNN with k=1. Make sure you keep the same order as in the test set file. Also, remember to apply the Min-Max normalisation for each feature.

Predicted class labels for k = 1:
[3 3 3 1 1 1 1 2 1 2 2 3 3 3 1 2 3 3 1 1 3 2 2 3 2 3 2 3 2 1 2 1 2 1 2 2 2
2 2 1 2 2 3 1 2 1 3 2 2 1 3 1 1 3 3 1 1 3 1 3 3 2 2 3 2 3 3 1 1 2 2 3 2 2
1 1 1 3 1 1 2 2 3 1 2 1 1 2 1]
Classification accuracy: 94.38202247191011%

b) Report the classification accuracy on the test set of the k-nearest neighbour method where k=3, and compare and comment on the predictive performance of the two classifiers (k=1 and k=3).

Predicted class labels for k = 3:
[3 3 3 1 1 1 1 2 1 2 3 3 3 3 1 2 3 3 1 1 3 2 2 3 2 3 2 3 2 1 2 1 2 1 2 2 2
2 2 1 2 2 3 1 2 1 3 2 2 1 3 1 1 3 3 1 1 3 1 3 3 1 2 3 2 3 3 1 1 2 1 3 2 2
1 1 1 3 1 1 2 2 3 1 2 1 1 2 1]
Classification accuracy: 95.50561797752809%

k = 3 is slightly better at predicting classes than k = 1, as with k = 1 it can be susceptible to randomness.

c) Discuss the main advantages and disadvantages of k-Nearest Neighbour method. Also, discuss the impact of increasing and decreasing k, for example, what happens when k=total size of the dataset?

For kNN, training the data is actually quite efficient, as all that is needed to be done is to load the data in. The predicting part is much more costly, as that is when we have to get the machine to do expensive calculations for distance. Oftentimes this is bad, as we would like to have a quick prediction made when we need it.
kNN is good when calculating distance between data that has continuous features (i.e. with numerical values), but it is not well-adapted for nominal features that are not numerical, as then the distance calculation does not work.
While distance makes sense with lower dimensions, it loses meaning in higher dimensions.
kNN can be easily adapted for regression.
Lower values of k can be noisy, and are subject to effects of outliers. Larger values of k are better as they are less susceptible to outliers, but if you go too large, then groups with few members will always be voted out. If you use k = size of dataset, then the prediction will always just be the mode of the entire dataset.
It is best to choose odd numbers for k, as we want to avoid ties

d) Describe the steps to apply the k-fold cross validation method for this dataset where k=5 (the number of folds). State the major steps

The first step is to split the dataset into 5 groups.
For each group:

We take the group as a test data set, and take the remaining groups as training sets
We do kNN on the test set, using our training sets.
Keep track of the accuracy, and move on to the next group

Then we summarise the skill of the model by taking an average of accuracy.

Part 2

a) You should first apply your program to the hepatitis-training and hepatitis-test files and report the classification accuracy in terms of the fraction of the test instances that it classified correctly. Report the constructed decision tree classifier printed by your program. Compare the accuracy of your decision tree program to the baseline classifier (which always predicts the most frequent class in the training set), and comment on any difference.

The accuracy for the baseline classifier is 80%. Below is my outputted decision tree, which has an accuracy of 76%. My decision tree is of course slightly lower than that of the baseline classifier.

```
ASCITES  = True:
    SPIDERS  = True:
      VARICES  = True:
        STEROID  = True:
          Class  live , prob= 1.0
        STEROID  = False:
          SPLEENPALPABLE  = True:
            FIRMLIVER  = True:
              Class  live , prob= 1.0
            FIRMLIVER  = False:
              BIGLIVER  = True:
                SGOT  = True:
                  Class  live , prob= 1.0
                SGOT  = False:
                  FEMALE  = True:
                    Class  live , prob= 1.0
                  FEMALE  = False:
                    ANOREXIA  = True:
                      Class  die , prob= 1.0
                    ANOREXIA  = False:
                      Class  live , prob= 1.0
              BIGLIVER  = False:
                Class  live , prob= 1.0
          SPLEENPALPABLE  = False:
            ANOREXIA  = True:
              Class  live , prob= 1.0
```

```
        ANOREXIA  = False:
          Class  die , prob= 1.0
    VARICES  = False:
      Class  die , prob= 1.0
  SPIDERS  = False:
    FIRMLIVER  = True:
      ANOREXIA  = True:
        SGOT  = True:
          Class  live , prob= 1.0
        SGOT  = False:
          Class  die , prob= 1.0
      ANOREXIA  = False:
        Class  live , prob= 1.0
    FIRMLIVER  = False:
      SGOT  = True:
        BIGLIVER  = True:
          Class  live , prob= 1.0
        BIGLIVER  = False:
          Class  die , prob= 1.0
      SGOT  = False:
        Class  live , prob= 1.0
ASCITES  = False:
  BIGLIVER  = True:
    VARICES  = True:
      FIRMLIVER  = True:
        STEROID  = True:
          Class  die , prob= 1.0
        STEROID  = False:
          BILIRUBIN  = True:
            Class  live , prob= 1.0
          BILIRUBIN  = False:
            Class  die , prob= 1.0
      FIRMLIVER  = False:
        Class  live , prob= 1.0
    VARICES  = False:
      Class  die , prob= 1.0
  BIGLIVER  = False:
    Class  live , prob= 1.0
```

b) ”Pruning” (removing) some of the leaves of the decision tree will make the decision tree less accurate on the training set. Explain:

   i) What criteria would you use to decide which leaves can be pruned from your decision tree? Why would you choose this criteria?

   ii) Why pruning reduces accuracy on the training set?

   iii) When pruning, should we expect the accuracy on the test set to decrease as well?

We want to prune the redundant leaf nodes. One simple way to choose which leaves to prune, is to try replacing it with just the most popular class. If the accuracy on the test set isn't affected, then we keep it removed. This criteria is naive, but also simple and fast to implement.
You can do either pre-pruning or post-pruning. Pre-pruning is done by just stopping the tree early before it is done classifying the training set, while post-pruning fully classifies the training set and then prunes the tree.
The accuracy on the training set will decrease because we are reducing the complexity. Even though accuracy on the training set decreases, the accuracy on the test set shouldn't and may even increase, as we are specifically pruning less pure leaf nodes that are redundant.

Part 3

a) Report on the accuracy of your perceptron. For example, did it find a correct set of weights? Did its performance change much between different runs?

Accuracy for perceptron: 88.88% after 100 iterations. The accuracy remains a consistent amount between runs.
Even after further iterations (i.e. 1000), hovers around 88% - 92% and does not reach 100% accuracy, so the algorithm does not find a fully correct set of weights.
The data is not linearly separable, so it the weights keep changing to get a good accuracy

b) Explain why evaluating the perceptron's performance on the training data is not a good measure of its effectiveness. You should split the dataset and perform a fairer evaluation.

Splitting the dataset into a training and test set should yield better results.