```
// Salvador Romero
//K Nearest Neighbor Algorithm
// HW 2 Problem 1

#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <cmath>


using namespace std;

struct plant {
float sepLen;
float sepWid;
float petLen;
float petWid;
string name;
};

int main() {

vector<plant> iris;
iris.push_back(plant());

ifstream dataFile;
dataFile.open("irisdataset.txt");

//if the file is unable to open
if(!dataFile){
cout<<"unable to open file";
exit(1);
}

int plantCount = 0;
float First, Second, Third, Fourth, Fifth;
float normalize;
char comma1, comma2, comma3, comma4;
string plantname;

while(true){

iris.push_back(plant());

dataFile >> First;
First = (First - 4.3)/(...);
iris[plantCount].sepLen = First;

dataFile >> comma1;

dataFile >> Second;
Second = (Second - 2)/(...);
iris[plantCount].sepWid = Second;

dataFile >> comma2;

dataFile >> Third;

Third = (Third - 1)/(...);
iris[plantCount].petLen = Third;
dataFile >> comma3;

dataFile >> Fourth;
Fourth = (Fourth - 0.1)/(...);
iris[plantCount].petWid = Fourth;

dataFile >> comma4;

dataFile >> plantname;
iris[plantCount].name = plantname;

plantCount = plantCount + 1;
if(dataFile.eof()) break;
}

```

```
//normalize = (iris[].sepLen-4.3)/(7.9-4.3);
//cout<< normalize << endl;
//cout << iris[128].sepLen << endl;
/*for(int i = 0; i<plantCount-1;i++){


        cout << setw(12) <<iris[i].sepLen<<" ";
        cout << setw(12) <<iris[i].sepWid<<" ";
        cout << setw(12) <<iris[i].petLen<<" ";
        cout << setw(12) <<iris[i].petWid<<" ";
        cout << iris[i].name << endl;
    }
    */

float userSepLen
float userSepWid
float userPetLen
float userPetWid
float sepal
float sepal
float petal
float petal
float distance
float nearest
int index
int found
int kValue
int test


bool false
vector foundVals

cout "Enter kValue "
cin kValue
cout "Enter the sepal length in cm: "
cin userSepLen
userSepLen
cout "Enter the sepal width in cm: "
cin userSepWid
userSepWid
cout "Enter petal lenth in cm: "
cin userPetLen
userPetLen
cout "Enter petal width in cm: "
cin userPetWid
userPetWid

for kValue
{
nearest

for i plantCount

for k foundVals
foundVals
true Vec

if Vec

userSepLen
userSepWid
userPetLen
userPetWid

pwal
dwal
pwal
dwal

distance

distance nearest
nearest distance
index
}
```