Project 7

You will be divided into groups of 4 students for this project.

You must use GitHub to collaborate with your classmates. Designate a project manager and add every member as a collaborator (click on repository, settings, collaborators). Add me trishduce as a collaborator. Everyone will be able to merge to the master branch without a pull request – there is no forking.

Create a Binary Search Tree (ADT). It will be a tree of integers – integers will be inserted according to their value (integer is the key).

binarytree.h

The type binary_tree is a struct with 2 data members

- an integer - number of elements in the tree
- a pointer to the root of the tree

Function declarations for the binary tree should look like the following.

```
void initialize(binary_tree* bt);
bool search(binary_tree* bt, int key);
void insert(binary_tree* bt, int item);
void printinorder(binary_tree* bt);
void printpreorder(binary_tree* bt);
void printpostorder(binary_tree* bt);
int btsize(binary_tree* bt);
int treeheight(binary_tree* bt);
```

Create a main.c that reads integers from a file specified on the command line and inserts them into the binary tree.

```
myfile.txt
8 34 2 67 76 45 33 1 23 18
```

Output should look like the following:

```
Print in order
1 2 8 18 23 33 34 45 67 76
Print pre order
8 2 1 34 33 23 18 67 45 76
Print post order
1 2 18 23 33 45 76 67 34 8
FOUND 33
Number of elements in tree: 10
Tree height: 4
```

What code should look like in main.c:

```
printf("Print in order\n");
printinorder(&bt);
printf("\nPrint pre order\n");
printpreorder(&bt);
printf("\nPrint post order\n");
printpostorder(&bt);
printf("\n");
found = search(&bt,33);
if (found){
   printf("FOUND 33\n");
}
found = search(&bt,38);
if (found){
   printf("FOUND 38\n");
}
printf("Number of elements in tree: %d\n", btsize(&bt));
printf("Tree height: %d\n", treeheight(&bt));
```

Program should exit with code 1 and print an error message if the number of command line arguments is wrong.

Program should exit with code 1 and print an error message if the input file does not exist.

Create a script that:

- tests 3 different files (each file with a different set of integers)
- tests to see if error message displays when number of command line arguments is wrong
- tests to see if error message displays when input file does not exist

Create a makefile with modular compilation.

Create a **README** file that describes the project.

Comment your code.

There will be a strict deadline on this project.   Saturday, March 16th at midnight.   I will be more than happy to talk about this project with you but will not look at your code.