```
///////////////////////////////////////////////////////////////////////////////
////Class:        CS 445
////Semester:     Fall 2011
////Assignment:   Homework 4
////Author:       Colby Blair
////File name:    tree.h
////////////////////////////////////////////////////////////////////////////////

#ifndef _TREE_H
#define _TREE_H

#define MAX_KIDS 9

#include <stdarg.h>
#include "symtabc.h"

struct tree {
        //int prodrule; //not sure on how to set the int yet
        char *prodrule;
        int nkids;
        struct tree *kids[MAX_KIDS];
        struct tree_token *leaf; //points to tree_token, terminals only
};

struct tree_token {
        int cat;
        char *text;
        int lineno;
        char *fname;
};

//symbol table stuff
static int maxTable_;          // this is how big the table is now since it can grow
static SymTabEntry *table_;    // the table is simply a fancy stack of SymTabEntries


//tree  functions
int tree_init();
int tree_del(struct tree *);
int treeprint(struct tree *, int);
struct tree *tree_create_node(char *, int n_args, ...);
struct tree *tree_create_node_from_token(int, char*, int, char*);
int tree_gen_tac(struct tree *);


//Internal only
//getters of subtree members
void tree_get_subtree(char *, struct tree*, struct tree**);
char *tree_get_opt_type(struct tree*);
char *tree_get_opt_aux_flag(struct tree*);
char *tree_get_ident(struct tree*);

//general symbol table things
//int tree_update_sym_tab(struct tree *);

//general string stuff
//void tree_import_ident_to_path(char *, char **);

#endif
```