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

#include <stdio.h>
#include "parser.tab.h"
#include "main.h"
#include "tree.h"

FILE *yyin;
char *YY_FNAME;
extern struct tree *YY_TREE; //the tree the parser inserts into
extern int lineno;
extern int colno;
char yytext[YY_MAX_BUF];

int main(int argc, char *argv[])
{
        tree_init(); //inits YY_TREE
        yytext[0] = '\0';

        //the master tree that each yyparse will insert YY_TREE into
        struct tree *tree_master;

        int i;
        //for each filename given on the command line
        for(i = 1; i < argc; i++)
        {
                //open the file and store its reference in global variable yyin
                YY_FNAME = argv[i];
                //yyrestart for multiple file parsing
                FILE *yyfile = fopen(YY_FNAME, "r");
                yyrestart(yyfile); lineno = 1; colno = 1;
                //instead of - yyin = fopen(YY_FNAME,"r");
                if (yyin == NULL)
                {
                        fprintf(stderr, "ERROR: Cannot open '%s'. Ignoring.\n",\
                                                                YY_FNAME);
                        continue;
                }

                //print file name
                printf("%s\n", YY_FNAME);

                //parse file
                yyparse();

                //just set to, for now. Later, need to append each YY_TREE from
                // each parse to children of tree_master
                tree_master = YY_TREE;

                DEBUGMSG("DEBUG: done with file '%s'\n", YY_FNAME);
        }

        //generate code
        //tree_gen_tac(YY_TREE);
        tree_gen_tac(tree_master);

        #ifdef DEBUG_TREE
        //print our results
        treeprint(tree_master,0);
        #endif

        tree_del(tree_master);
        SymTab_free();
}
```