/\*

\* stable.h

\* Programmer - Patrick Fischer

\* This is the code for my symbol table.

\*

\*/

#include "stable.h"

#include "stdio.h"

// Global symbol table

struct symbol\_table \*table;

// Insert a new symbol\_table\_entry into table

void insert(struct symbol\_table\_entry entry) {

entry.address = table->memorySize;

table->table[table->count] = entry;

table->count++;

table->memorySize += entry.size;

// printf("name = %s, address = %d, kind = %d, type = %d, size = %d, count = %d\n", entry.name, entry.address, entry.kind, entry.type, entry.size, table->count);

}

// If name is present in table, then return its' array location.

int isPresent(char \*name) {

int i;

for(i = 0; i < 100; i++) {

if(table->table[i].name == name) {

return i;

}

}

return -1;

}

// Return the symbol\_table\_entry given by name

struct symbol\_table\_entry retrieve(char \*name) {

return table->table[isPresent(name)];

}

struct symbol\_table\_entry createEntry(char \*name, int address, int kind, int type, int size) {

struct symbol\_table\_entry entry;

entry.name = name; // This might need a malloc

entry.address = address;

entry.kind = kind;

entry.type = type;

entry.size = size;

return entry;

}