/\*

\* Stack containing race conditions

\*/

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

// Linked list node

typedef int value\_t;

typedef struct Node

{

value\_t data;

struct Node \*next;

} StackNode;

// Stack function declarations

void push (value\_t v, StackNode \*\*top);

value\_t pop ( StackNode \*\*top);

int is\_empty( StackNode \*top);

int main(void)

{

StackNode \*top = NULL;

push(5, &top);

push(10,&top);

pop ( &top);

push(15,&top);

pop ( &top);

pop ( &top);

push(20,&top);

push(-5, &top);

pop ( &top);

push(-10,&top);

pop ( &top);

pop ( &top);

push(-15,&top);

pop ( &top);

push(-20,&top);

return 0;

}

// Stack function definitions

void push(value\_t v, StackNode \*\*top)

{

StackNode \* new\_node = malloc(sizeof(StackNode));

new\_node->data = v;

new\_node->next = \*top;

\*top = new\_node;

}

value\_t pop(StackNode \*\*top)

{

if (is\_empty(\*top)) return (value\_t)0;

value\_t data = (\*top)->data;

StackNode \* temp = \*top;

\*top = (\*top)->next;

free(temp);

return data;

}

int is\_empty(StackNode \*top) {

if (top == NULL) return 1;

else return 0;

}