#ifndef MONTY\_H

#define MONTY\_H

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <unistd.h>

#include <fcntl.h>

#include <string.h>

#include <ctype.h>

/\*\*

\* struct stack\_s - doubly linked list representation of a stack (or queue)

\* @n: integer

\* @prev: points to the previous element of the stack (or queue)

\* @next: points to the next element of the stack (or queue)

\*

\* Description: doubly linked list node structure

\* for stack, queues, LIFO, FIFO Holberton project

\*/

typedef struct stack\_s

{

int n;

struct stack\_s \*prev;

struct stack\_s \*next;

} stack\_t;

/\*\*

\* struct bus\_s - variables -args, file, line content

\* @arg: value

\* @file: pointer to monty file

\* @content: line content

\* @lifi: flag change stack <-> queue

\* Description: carries values through the program

\*/

typedef struct bus\_s

{

char \*arg;

FILE \*file;

char \*content;

int lifi;

} bus\_t;

extern bus\_t bus;

/\*\*

\* struct instruction\_s - opcode and its function

\* @opcode: the opcode

\* @f: function to handle the opcode

\*

\* Description: opcode and its function

\* for stack, queues, LIFO, FIFO Holberton project

\*/

typedef struct instruction\_s

{

char \*opcode;

void (\*f)(stack\_t \*\*stack, unsigned int line\_number);

} instruction\_t;

char \*\_realloc(char \*ptr, unsigned int old\_size, unsigned int new\_size);

ssize\_t getstdin(char \*\*lineptr, int file);

char \*clean\_line(char \*content);

void f\_push(stack\_t \*\*head, unsigned int number);

void f\_pall(stack\_t \*\*head, unsigned int number);

void f\_pint(stack\_t \*\*head, unsigned int number);

int execute(char \*content, stack\_t \*\*head, unsigned int counter, FILE \*file);

void free\_stack(stack\_t \*head);

void f\_pop(stack\_t \*\*head, unsigned int counter);

void f\_swap(stack\_t \*\*head, unsigned int counter);

void f\_add(stack\_t \*\*head, unsigned int counter);

void f\_nop(stack\_t \*\*head, unsigned int counter);

void f\_sub(stack\_t \*\*head, unsigned int counter);

void f\_div(stack\_t \*\*head, unsigned int counter);

void f\_mul(stack\_t \*\*head, unsigned int counter);

void f\_mod(stack\_t \*\*head, unsigned int counter);

void f\_pchar(stack\_t \*\*head, unsigned int counter);

void f\_pstr(stack\_t \*\*head, unsigned int counter);

void f\_rotl(stack\_t \*\*head, unsigned int counter);

void f\_rotr(stack\_t \*\*head, \_\_attribute\_\_((unused)) unsigned int counter);

void addnode(stack\_t \*\*head, int n);

void addqueue(stack\_t \*\*head, int n);

void f\_queue(stack\_t \*\*head, unsigned int counter);

void f\_stack(stack\_t \*\*head, unsigned int counter);

#endif