```c
/*-------------------------------------------------------------------------+
File Name: General.h
+-------------------------------------------------------------------------+

+-------------------------------------------------------------------------+
   Programming III COP4338
   Author: Daniel Gonzalez P#4926400
   assignment 5: Date Validate / Format
   Date: 11/08/2016 ELLECTION DAY!!!!

  program description
     Input: Accept input for the first program via the command-line arguments.
             Input will be the number of valid entries to be redirected from
             the dates input file (dates.dat). A zero indicates to input all
             entries from the dates input file. Your program should validate
             (day, month & year - see page 111 for validation ideas) and skip
             corrupt dates in the dates.dat file (see page 159 for scanning
             ideas). This validated input will then be piped out to the
             second program.
            The second program will accept these validated dates in the
            month/day/year format and convert each of them to the day,
            abbreviated month & year format – both exhibited above. The
            abbreviated month should consist of the first three letters of
            the month, capitalized. These converted results will be redirected
            to the output file (output.dat), followed by a copy of the
            original (dates.dat) data.

     Output: Generates an output file (output.dat) that contains a
             converted list of dates in day, abbreviated month & year
             format (i.e. 1 JAN 1900), followed by the original list of
             dates in month/day/year format (i.e. 1/1/1900). This output file
             will be the result of appending the input file (dates.dat), which
             is accessed by the first program, with the result output file
             (output.dat), generated by the second program.


  +--------------------------------------------------------------------------+
  | I Daniel Gonzalez #4926400 hereby certify that this collective work is   |
  | my own and none of it is the work of any other person or entity.         |
  +--------------------------------------------------------------------------+


  how to compile and execute:
    1.Open the terminal
        Go to the program folder that contains all the files required for
        the program to compile including all header files(*.h).
        Run the following command "make"

    2.Open the terminal
        Go to the program folder that contains all the files required for
        the program to compile including all header files(*.h).
        "gcc -Wall -w -lm dateValidate.c Game.c Deck.c Card.c Player.c -o poker"

    Program execution:
    From the terminal enter:
     "./validateDate < dates.dat [X] | ./format > output.dat"

     X: is the amount of validated dates



+-------------------------------------------------------------------------*/

#include <stdio.h>  /* using fgets, fputs, fputc, sscanf */
#include <stdlib.h> /* using atoi, malloc, realloc, free*/
#include <string.h> /* using strcpy, strncat */
#include <ctype.h>  /* using isspace */
#include <limits.h> /* using INT_MAX */
```

```c
#include <math.h>   /* using fmod */



/* Defines */
#define MAX_LINE 256                      /* maximum line size*/
#define MAX_MONTH_NAME 10                 /*maximum amount of chars
                                           for a month name*/
#define MIN_FILE_LINES 70                 /* minimum amount of lines in file*/
#define HAND_SHAKING_SIGNAL '\t'          /* character to signal end of input*/
#define VALID_KEYS 3                      /* number of variables
                                             returned from valid scan*/
#define LB_ORIGINAL_DATA "Original Data:" /*Label for original data*/
#define MAX_AMOUNT_ALLOWED_ARGS 2         /* maximum amount of arguments */
#define LEAP_YEAR 4                       /* frequency in which a
                                            leap year happens*/
#define ENABLE_ERROR FALSE                /* enable to print errors
                                            to the console */

/* program constants */
static const char * ERROR_DESCRIPTIONS [] = {"No errors",
                                             "Error",
                                             "Unable to read date",
                                             "Date contains decimals",
                                             "Date values are huge",
                                             "Month is Invalid",
                                             "Day is invalid"};



static const char * MONTH_NAMES_SHORT [] = {
                                              "INVALID","JAN", "FEB", "MAR",
                                              "APR", "MAY", "JUN", "JUL",
                                              "AUG", "SEP", "OCT",
                                              "NOV", "DEC"};


/* type definitions and structs */
typedef enum {FALSE, TRUE} Boolean;

typedef enum {DATE_FORMAT_SLASH_DOUBLE,
               DATE_FORMAT_SLASH,
               DATE_FORMAT_SHORT_MONTH} Format;

typedef enum {NO_ERRORS,
               ERRORS,
               ERRORS_UNABLE_TO_READ,
               ERRORS_DECIMALS_IN_DATE,
               ERRORS_HUGE_DATE,
               ERRORS_INVALID_MONTH,
               ERRORS_INVALID_DAY} Error;

typedef struct Line{
   char content[MAX_LINE];
} Line;


typedef struct LineList{
   Line * lines;
   int size;
   int capacity;
} LineList;

typedef struct DateKey{
   int day;
   int month;
   int year;
   Error error;
} DateKey;
```

```
140
141 typedef struct Data{
142    char input [MAX_LINE];
143    char output [MAX_LINE];
144    char error [MAX_LINE];
145 } Data;
146
147
148
149 /* function prototypes */
150 Boolean appendToLineList(LineList * , const char *);
151 void freeLineList(LineList * );
152 DateKey getDateKeys(char *);
153 Boolean hasValidDateType(DateKey *, double, double, double);
154 void initializeLineList(LineList * );
155 Boolean isLeapYear(int );
156 Boolean isValidDate(DateKey *);
157 void printError(char *, char *, Error);
158 void printfDate(char * , Format, DateKey * );
```