```
#/*-------------------------------------------------------------------------+
#File Name: Makefile
#+-------------------------------------------------------------------------+
#
#+-------------------------------------------------------------------------+
#   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).
#        COMPILE: "gcc -Wall -w -lm readDate.c dateValidate.c -o validateDate"
#
#    Program execution:
#    From the terminal enter:
#     "./validateDate < dates.dat [X] | ./format > output.dat"
#
#     X: is the amount of validated dates
#
#
#
#
#+-------------------------------------------------------------------------*/
COMPILER = gcc
FLAGS =    -Wall -w
LIBS = -lm
READ_OBJECT = readDate.o
FORMAT_OBJECT = format.o
GENERAL_OBJECT = functions.o
```

```
70 HEADER = general.h
71 OBJS = functions.o readDate.o format.o
72
73 all: $(GENERAL_OBJECT) read format clean
74
75 # Link file and format file.
76 read:    $(READ_OBJECT)
77         $(COMPILER) $(FLAGS) $(READ_OBJECT) $(GENERAL_OBJECT) \
78         -o validateDate $(LIBS)
79
80 format:    $(FORMAT_OBJECT)
81         $(COMPILER) $(FLAGS) $(FORMAT_OBJECT) $(GENERAL_OBJECT) \
82         -o format $(LIBS)
83
84 #Compile each individual objects with headers
85 $(OBJS) : $(HEADER)
86
87 .PHONY:    clean
88 clean:
89         rm *.o
```