Making Sense of Ones and Zeros

### Due Date: September 4, 2014 @ midnight Submission Subject: “Making Sense of Ones and Zeros”

## General Submission Criteria:

All laboratory assignments of must be submitted using git.[[1]](#footnote-0) In addition to the submission of the assignment the following criteria must be meet for all assignments.

1. Your laboratory software project must be stored within a faculty-accessible git repository.
2. This git repository must have been updated via a series of “commit”s and “push”es. There must be at least one commit/push after each laboratory meeting.
3. You must have a README file associated with your software project, and it must provide a log/journal of the activities you performed on your project and it must provide a summary of your project status.
4. An email is sent with the appropriate subject to [steve@my.csun.edu](mailto:steve@my.csun.edu) that
   1. announces that the assignment is complete and ready to be graded.
   2. a copy of your README document
   3. includes the git URL for download and testing
5. Your submission must include a Makefile

## Description:

In this software project, you are to write a command line tool that will read a series of 1’s and 0’s from a file (These 1’s and 0’s are ASCII values and not binary numbers). This file may be either explicitly named on the command-line or it may be **stdin**. For each 8 characters in this file, you are to output a single line of text onto **stdout**.

The output will consists of four (4) columns that consists of:

- the original value

- the corresponding ASCII character

- the corresponding decimal value

- the value of either EVEN or ODD depending on the number of 1’s set.

## Programming Requirements:

* The program must be written in C.
* You must check the command line arguments for the name of the file.  
  If the file name is “-” or not provided, you must use **stdin**.
* You are to use the low level I/O primitives for reading, e.g., read(2)
* You may use the standard I/O library for output, e.g., printf(3)

## Caveats:

* Any value that is not a printable ASCII character should be represented by the mnemonic given via “man ascii”.
* Spaces and newline characters may appear in the input file, but your program is not required to function appropriately with them included in the input.)
* If the last set of 1’s and 0’s is not a full complement of eight characters, the last value should be padded with the appropriate number of 0’s on the right. (E.g, 101101 ⇒ 10110100)

## Usage Example:

|  |
| --- |
| $ cat filename  01011101 10110111 11101011 111101  $ zero-one filename  Original ASCII Decimal Parity T.Error  -------- -------- -------- -------- --------  01011101 ] 93 ODD TRUE  10110111 7 55 EVEN FALSE  11101011 k 107 EVEN FALSE  11110100 t 116 ODD TRUE  $ |

## See Also:

* man ascii
* man 3 printf
* man 2 read
* <https://docs.google.com/a/my.csun.edu/document/d/14rWbPWIrOT6jDO-anvRPEsHsaiIO7ST3UiGVPM1nEYI/edit#heading=h.9k6pqhlgo9qt>
* man isascii

1. Git is a distributed revision control and source code management (SCM) system. [↑](#footnote-ref-0)