**C Braille Activity**

**What is Braille?**

Braille is a tactile [writing system](https://en.wikipedia.org/wiki/Writing_system) used by people with [visual impair](https://en.wikipedia.org/wiki/Visually_impaired)ment. It is traditionally written with embossed paper. Braille users can read computer screens using [refreshable braille displays](https://en.wikipedia.org/wiki/Refreshable_braille_display). They can write braille with the original [slate and stylus](https://en.wikipedia.org/wiki/Slate_and_stylus) or type it on a [braille writer](https://en.wikipedia.org/wiki/Braille_writer), such as a portable [braille notetaker](https://en.wikipedia.org/wiki/Braille_notetaker) or computer that prints with a [braille embosser](https://en.wikipedia.org/wiki/Braille_embosser).

**Requirements**

Write a program that translates ASCII text to Braille output. Use getchar() to read the input. An EOF (CTRL-D) terminates the program. You may assume that no input line is over 80 characters long. Output should be recognisable as Braille, the form of output could be o for a raised dot and . for a non-raised dot. Only the 26 alphabet characters and space are required for a minimal solution. All input characters not supported by your solution should be ignored. You are not allowed to use Unicode Braille characters. Vertical display of letters in Braille is acceptable. Bonus points for displaying Braille letters horizontally.

**Example output:**  


To test your code, compare its output to this online Braille translator output: http://www.byronknoll.com/braille.html

**Problem Breakdown**

1. Braille was the first writing system with [binary](https://en.wikipedia.org/wiki/Binary_numeral_system) [encoding](https://en.wikipedia.org/wiki/Character_encoding). The system as devised by Braille consists of two parts:

* [Character encoding](https://en.wikipedia.org/wiki/Character_encoding) that mapped characters of the [alphabet](https://en.wikipedia.org/wiki/French_alphabet) to [tuples](https://en.wikipedia.org/wiki/Tuple) of six [bits](https://en.wikipedia.org/wiki/Bit) (the dots),
* The physical representation of those six-bit characters with raised dots in a braille cell.

1. Within an individual cell, the dot positions are arranged in two columns of three positions. A raised dot can appear in any of the six positions, producing sixty-four (26) possible patterns, including one in which there are no raised dots.
2. A pattern is commonly described by listing the positions where dots are raised, the positions being universally numbered, from top to bottom, as 1 to 3 on the left and 4 to 6 on the right. For example, dot pattern 1-3-4 describe a cell with three dots raised, at the top and bottom in the left column and at the top of the right column: that is, the letter ⠍ *m*.
3. The first ten letters of the alphabet, *a–j,* use the upper four dot positions: ⠁⠃⠉⠙⠑⠋⠛⠓⠊⠚ (black dots in the table below).
4. The next ten letters, *k–t,* are identical to *a–j,* respectively, apart from the addition of a dot at position 3 (red dots in the table): ⠅⠇⠍⠝⠕⠏⠟⠗⠎⠞
5. The next ten letters (the next "[decade](https://en.wiktionary.org/wiki/decade#English)") are the same again, but with dots also at positions both 3 and 6 (green dots). Here *w* was left out as not being a part of the official French alphabet at the time of Braille's life.
6. Space in Braille is represented by a cell with no dots raised **( ⠀ )**.

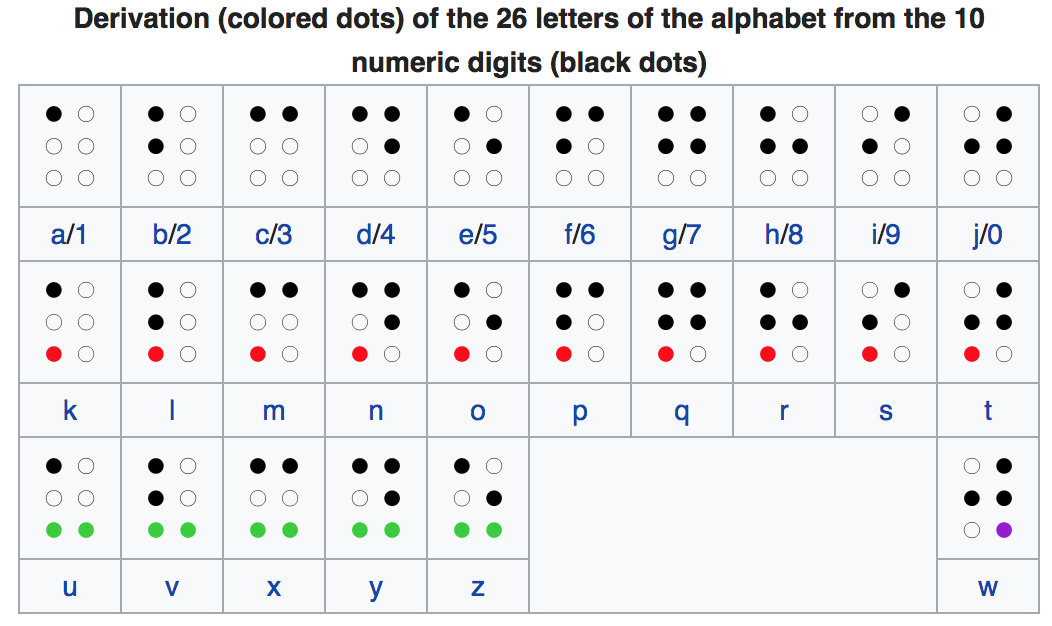


Image source: <https://en.wikipedia.org/wiki/Braille>

### **Submission**

Submit your source file braille\_translator.c and your ActivityJournal.txt in a top level directory named BrailleTranslator to your Git pushbox.

**Notes:**

An example for more full project for potential future assignments

<https://github.com/LazoCoder/Braille-Translator/blob/master/alphaToBraille.py>