Prior to having fancy iPhones, teenagers would wear out their thumbs sending SMS messages on candybar-shaped feature phones with 3x4 numeric keypads.

```
 ------- ------- -------
 |     | | ABC | | DEF |
 |  1  | |  2  | |  3  |
 ------- ------- -------
 ------- ------- -------
 | GHI | | JKL | | MNO |
 |  4  | |  5  | |  6  |
 ------- ------- -------
 ------- ------- -------
 |PQRS | | TUV | | WXYZ|
 |  7  | |  8  | |  9  |
 ------- ------- -------
 ------- ------- -------
 |     | |space| |     |
 |  *  | |  0  | |  #  |
 ------- ------- -------
```

Prior to the development of T9 (predictive text entry) systems, the method to type words was called "multi-tap" and involved pressing a button repeatedly to cycle through the possible values.

For example, to type a letter `"R"` you would press the `7` key three times (as the screen display for the current character cycles through `P->Q->R->S->7`). A character is "locked in" once the user presses a different key or pauses for a short period of time (thus, no extra button presses are required beyond what is needed for each letter individually). The zero key handles spaces, with one press of the key producing a space and two presses producing a zero.

In order to send the message `"WHERE DO U WANT 2 MEET L8R"` a teen would have to actually do 47 button presses. No wonder they abbreviated.

For this assignment, write a module that can calculate the amount of button presses required for any phrase. Punctuation can be ignored for this exercise. Likewise, you can assume the phone doesn't distinguish between upper/lowercase characters (but you should allow your module to accept input in either for convenience).

Hint: While it wouldn't take too long to hard code the amount of keypresses manufacturer who might be testing out different keyboard layouts, and you want to be able to test new ones rapidly.)