Advanced Input Validation

Input Validation Using String Methods

1. Ask the user to enter a seven-letter word, all in capital letters, and ensure that they do this.
2. Obtain a string from the user that represents a Canadian postal code, and check if it is valid or not. All valid postal codes have the form *A*1*A* 1*A*1, where *A* is a letter, and 1 is a number, with the following two restrictions:
   - No postal code includes the letters D, F, I, O, Q or U, and
   - the first letter is neither W nor Z.

   If the postal code is invalid, indicate that it is not acceptable and obtain a new one.
3. Ask the user to enter an email address. For our purposes, an email address will have the form xxx@yyy.zzz, with the following specifications:
   - xxx and yyy may consist of any sequence of letters, digits, periods, or underscores.
   - zzz must be one of the following: com, net, org.

   If the email address is invalid, indicate that it is not acceptable and obtain a new one.
4. Write a function, *get_integer*, that repeatedly prompts the user to enter an integer. For our purposes, we can define an integer as one of two cases:
   - the input consists solely of numeric digits (no letters or special characters), or
   - the input consists of a single hyphen, followed only by numeric digits.

   Your function should return the integer value, typecast as an integer.
5. Write a function, *get_float*, that repeatedly prompts the user to enter a floating point value. For our purposes, we can define a floating point value as one of X cases:
   - the input consists solely of numeric digits (no letters or special characters), or
   - the input consists of a single period, optionally preceded by one or more numeric digits and followed by one or more numeric digits, or
   - input from one of the previous two cases is preceded by a single hyphen

   Your function should return the floating point value, typecast as a float.