DEA IN WITH HANDS : 10.0 POINTS

Please read this problem entirel before ou begin coding the solution

EP ESENTIN HANDS

A hand is the set of letters held b a pla er during the game. The pla er is initiall dealt a set of random letters. For example the pla er could start out with the following hand: `a, q, l, m, u, i, l` **u am ,a a ill a a i i a a l a l a alu a um im a iula l i a i a a am l , a a ul a**

```
hand = {'a':1, 'q':1, 'l':2, 'm':1, 'u':1, 'i':1}
```

`'l'` is represented. em em ber that with a dictionary, the usual way to access a value is `hand['a']`, where `'a'` is the key we want to find. However, this only works if the key is in the dictionary otherwise, we get a `KeyError`. To avoid this, we can use the call `hand.get('a',0)`. This is the "safe" way to access a value if we are not sure the key is in the dictionary. `d.get(key,default)` returns the value for `key` if `key` is in the dictionary `d`, else `default`. If `default` is not given, it returns `None`, so that this m ethod never raises a `KeyError`. For e am ple:

```
>>> hand['e']
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'e'
>>> hand.get('e', 0)
0
```

`getFrequencyDict`, defined near the top of `ps4a.py`. hen given a string of letters as an input, it returns a dictionary where the keys are letters and the values are the num ber of tim es that letter is represented in the input string. For e am ple:

```
>>> getFrequencyDict("hello")
{'h': 1, 'e': 1, 'l': 2, 'o': 1}
```

`displayHand` function. Take a few m inutes right now to read through this function carefully and understand what it does and how it works.

ENE ATIN A AND HAND

The hand a player is dealt is a set of letters chosen at random . e provide you with the im plem entation of a function that generates this random hand, `dealHand`. The function takes as input a positive integer `n`, and returns a new ob ect, a hand containing `n` lowercase letters. Again, take a few m inutes (right now !) to read through this function carefully and understand what it does and how it works.

a, q, l, m, u, i, l T ha ul ll

quail T i ul la ll i l i ha a l, m u a i im lm

u i updateHand , which takes in two inputs – a hand and a word (string). updateHand uses letters from the hand to spell the word, and then returns a copy of the hand , containing only the letters remaining. For example:

```
>>> hand = {'a':1, 'q':1, 'l':2, 'm':1, 'u':1, 'i':1}
>>> displayHand(hand) # Implemented for you
a q l l m u i
>>> hand = updateHand(hand, 'quail') # You implement this function!
>>> hand
{'l': 1, 'm': 1}
>>> displayHand(hand)
l m
```

NOTE: In the above example after the call to `updateHand` , it is also acceptable for the value of `hand` to be the dictionary `{'a':0, 'q':0, 'l':1, 'm':1, 'u':0, 'i':0}` . The value of `hand` will depend on your implementation but the output of `displayHand()` will be the same in either case.)

Implement the `updateHand` function. Make sure this function has no side effects: i.e., it must not mutate the hand passed in. Before pasting your function definition here, be sure you've passed the appropriate tests in `test_ps4a.py` .

```
def updateHand(hand, word):
    """
    Assumes that 'hand' has all the letters in word.
    In other words, this assumes that however many times
    a letter appears in 'word', 'hand' has at least as
    many of that letter in it.

    Updates the hand: uses up the letters in the given word
    and returns the new hand, without those letters in it.

    Has no side effects: does not modify hand.

    word: string
    hand: dictionary (string -> int)
    returns: dictionary (string -> int)
    """
```

Test results

CORRECT

ull u u

Hints

Copying Dictionaries

You may wish to review the .copy method of Python dictionaries (review this and other Python dictionary methods here (http://docs.python.org/library/stdtypes.html#mapping types dict)).

Testing

Testing: Make sure the `test_updateHand()` tests pass. You will also want to test your implementation of `updateHand` with some reasonable inputs.

Check Save *You have used 2 of 30 submissions*

Show sc ss o **New Post**