Problem Solving 1

When playing a game of solitaire, as with most card games, it is necessary to deal out some number of cards to start a game. Different games require different starting “hands”.

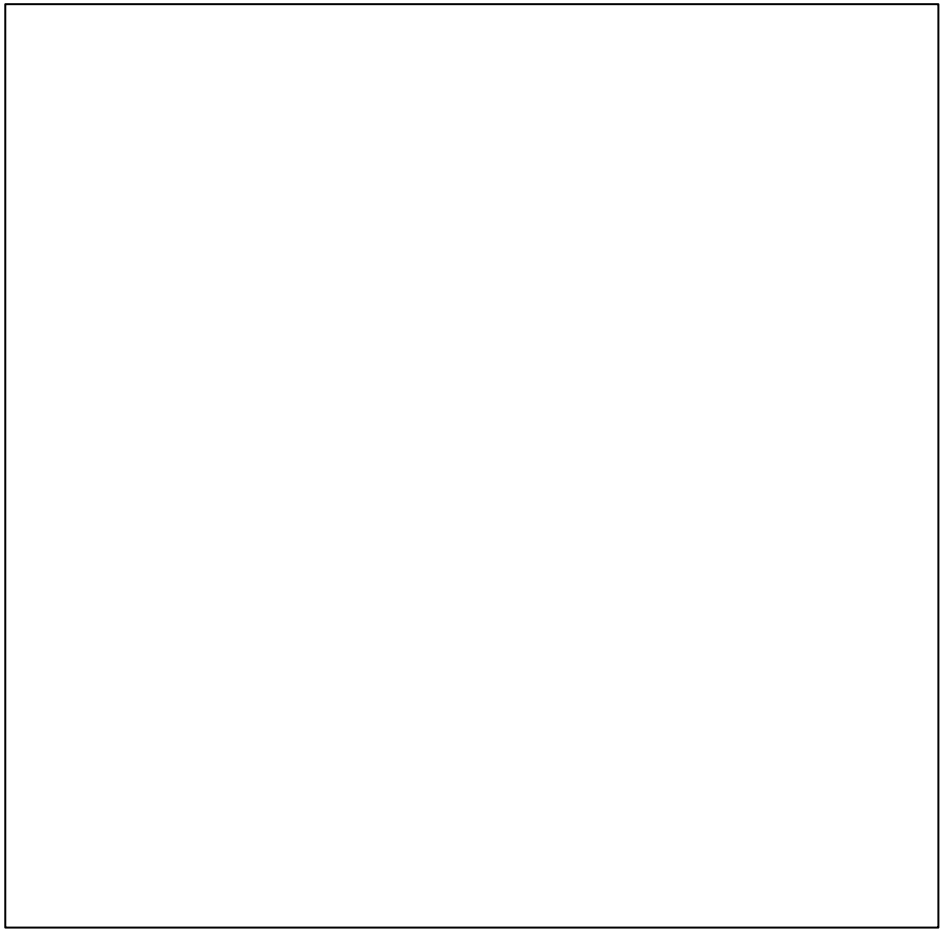
In the space to the right, write a function called “`deal_hand`” that makes a standard deck of cards, deals out 4 cards for the starting hand, and returns the hand.

Based on the first homework, what data structure(s) will you use to represent the hand? Indicate the type following your function.

```
def deal_hand():
     deck = []
     k = 0
     suits = (“hearts”,”diamonds”, etc
     ranks = (2,3,4,...”king”,”ace”)
     for j in range(4)
          suit = suits[j]
          for i in range(14)
               deck[k] = (ranks[i+2], j)
     deck.shuffle

     hand = []
     for i in range(4)
          pop = deck.pop(0)
          hand.append(pop)
     return hand

will use both lists and tuples for this
with hand and deck being lists
containing cards which are tuples
```



Problem Solving 2

When doing iterative development, it is important to create unit tests to confirm your code is not only working, but continues to work as your development progresses.

The function that you wrote in the previous problem created a standard deck of cards and dealt out some number of cards. What tests can you do at this point?

In the space to the left, write at least one test to confirm that your “`deal_hand`” function is working properly.

Problem Solving 3

Discarding is when you remove cards from your hand and place them in a “waste” or “discard” pile. Specific games vary in what card(s) get discarded when. You may even have to discard cards that are in a specific location.

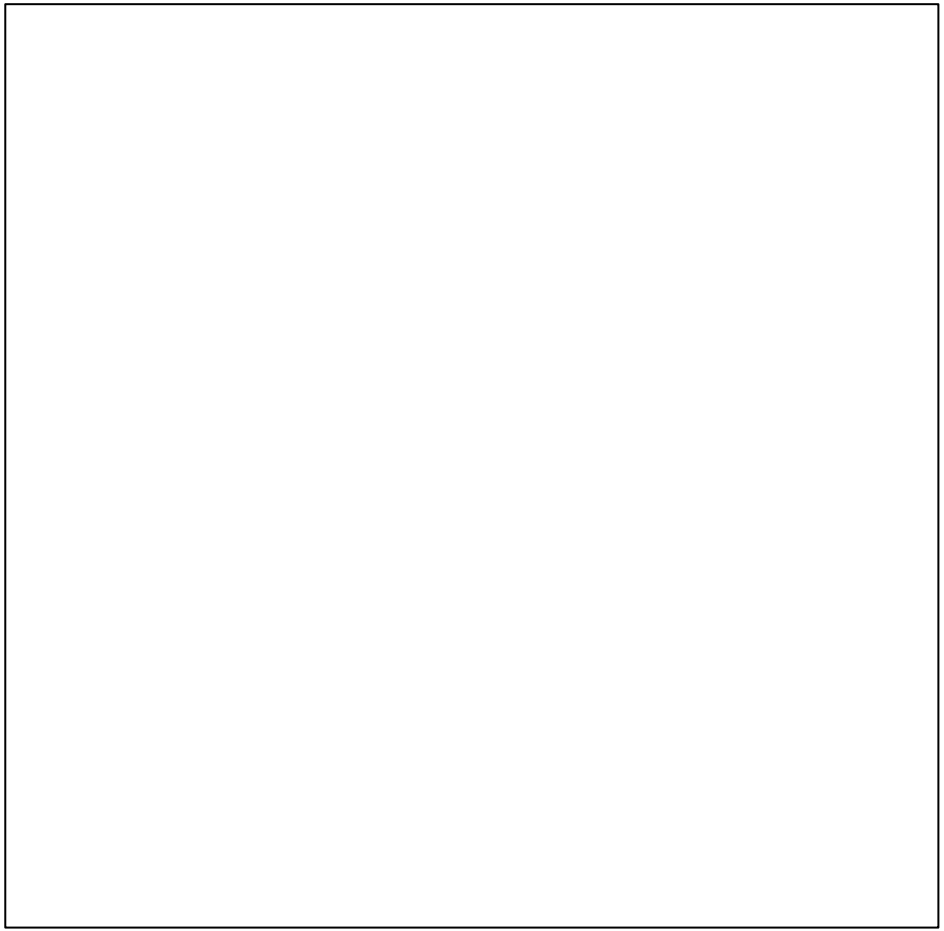
In the space provided, write a function called “`discard`”, that takes a `hand` and a `number` which represents how many cards to discard. You will discard the last n cards in the player’s hand using slicing. Return the hand.

Be sure to properly handle any potential errors. How would you avoid a runtime error?

```
import random

hand = random.randint(1,52)
discard_number = random.randint(1,52)
if discard_number

def discard(hand, discard_number):
      hand[]
```



Problem Solving 4

In the space to the left modify your “`discard`” function to pass a third parameter called `location` that specifies how many cards from the end (the right side of the hand) to keep before discarding the `number` of cards from the hand. As before, return the hand.

Problem Solving 5

What could you use to validate that your updated discard function is working properly?

Write at least one unit test to verify that your updated discard function works properly.