Computer Science 125
22 September 2020

1. Use a loop to create a list of random integers.

```
import random
nums = []
for i in range(10):
    nums.append(random.randrange(100))
print(nums)
```

2. Is 10 in the list?

```
if 10 in list:
    print("Found")
```

3. Is any multiple of 10 in the list?

loop over the list
inspect each element
keep track of whether we found a multiple of ten

```
for n in nums:
    if n % 10 == 0:
        print("Found")
```

Write a function that searches for a multiple of 10 in the list and stops when it finds one.

```
def findMultiple(alist):
    for n in alist:
        if n % 10 == 0
            return True   <- ends the function
    return False
```

4. Insert 100 at index 5 in the list.

4. Insert 100 at index 5 in the list.

example: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] becomes [0, 1, 2, 3, 4, 100, 5, 6, 7, 8, 9]

nums

Slices: nums[:5] + [100] + nums[5:]

[0, 1, 2, 3, 4] [5, 6, 7, 8, 9]

insert: nums.insert(5, 100)

index item

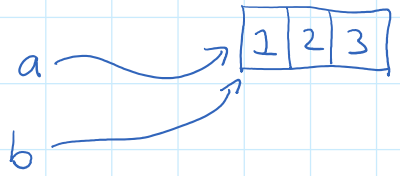
5. What will the following code print?

```
a = [1,2,3]
b = a
a[0] = 4
print(a)
print(b)
```

sets b as a pointer to list a, rather than making a copy of the list

[4, 2, 3]

[4, 2, 3] ~~or [1, 2, 3]~~

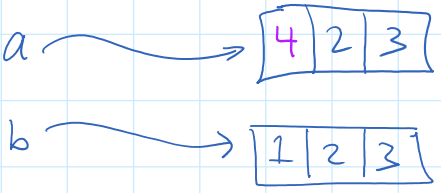


How would we make b a different list with the same values as a?

```
a = [1,2,3]
b = a[:]
a[0] = 4
print(a)
print(b)
```

[4, 2, 3]

[1, 2, 3]



PRACTICE WITH LISTS – SOLUTIONS

CS 125

Working with a partner/group, use the following steps to solve each of the following problems.

(a) Plan your function on the white board (either on the classroom wall or on Zoom). Write out your entire program. Think about what errors might occur and how to fix them.

(b) Plan multiple test cases for your function. What input will you send to your function? What value should the function return?

(c) *Only after you have completed steps (a) and (b) should you type your code in Python.*

(d) After you have typed your function, run your test cases. Does your function work? If not, how can you fix it?

1. Write a function `minimum(alist)` that returns the minimum of a list of numbers. Python has a built-in `min` function, but do not use it. Instead, iterate over the numbers in the list and keep track of the smallest number found.

   For example, `minimum([3, 7, 2, 4, 10])` returns `2`.

```
def minimum(alist):
    m = alist[0]
    for a in alist:
        if a < m:
            m = a
    return m
```

2. Write a function `isSorted(alist)` that determines whether a list of numbers is sorted in increasing order; that is, whether each number is less than or equal to the next number in the list. Your function should return `True` or `False`, depending on whether the list is sorted or not. For example:

   `isSorted([2, 5, 7, 8, 12, 15])` returns `True`
   `isSorted([2, 2, 2, 3, 3])` returns `True`
   `isSorted([2, 5, 10, 8, 12, 15])` returns `False`

```
def isSorted(alist):
    n = len(alist)
    for i in range(n-1):
        if alist[i] > alist[i+1]:
            return False
    return True
```

3. Write a program that asks the user to enter some text. Then use the Python's string `split` method to split the text into a list of words. Then print out all of the five-character words that the user entered.

```
text = input("Enter some text:")
words = text.split()
for w in words:
    if len(w) == 5:
        print(w)
```

4. A standard deck of playing cards contains 52 cards. Each card has a *suit* and a *value*. The suits are *spades*, *hearts*, *diamonds*, and *clubs*. The values are 2, 3, 4, 5, 6, 7, 8, 9, 10, *Jack*, *Queen*, *King*, *Ace*. Write a program that produces a shuffled deck of cards.

   Begin by writing a function `makeDeck`. This function should use loops to build a list representing a 52-card deck. In the list, each card should be represented by a string such as `"2Clubs"` or `"JackDiamonds"`.

   Then write a function `shuffle(deck)` that accepts a list as a parameter and shuffles it. One way to do this is to take each item in the list and swap it with another item at a randomly-chosen index. (Do not use Python's `random.shuffle` function.)

```
import random

def makeDeck():
    # set up lists of suits and values
    suits = ["Spades","Hearts","Diamonds","Clubs"]
    values = ["Jack","Queen","King","Ace"]
    for i in range(2,11):
        values.append(str(i))

    # now make the deck
    deck = []
    for s in suits:
        for v in values:
            deck.append(v+s)

    return deck

def shuffle(deck):
    n = len(deck)
    for i in range(n):
        j = random.randrange(n+1)
            temp = deck[i]
            deck[i] = deck[j]
            deck[j] = temp

    return deck

# testing
deck = makeDeck()
print(makeDeck())
print(len(deck))
print(shuffle(deck))
```