SY201 Twelve Week Exam Study Guide
Fall AY19
10/25/18

1. Given the code segments below, explain what each one does.

   a. Assume `list1` is a list of integers

```
for i in range(len(list1)-1):
   if list1[i] in list1[i+1:]:
      has_dups = True
      break
   else:
      has_dups = False
print(has_dups)
```

   b.

```
s1 = "Cyber Operations"
s2 = ""
i = 0
while i < len(s1):
   s2 += s1[i:]
   s2 += "#"
   i += 1
print(s2)
```

   c.

```
def main():
    x = 1
    y = [1, 2, 3]
    m(x,y)
    print("x is",x)
    print("y[0] is", y[0])

def m(number,numbers):
    number = 1001
    numbers[0] = 5555

main()
```

2. Write a ***function*** called `switch` that has two parameters, `first` and `last`. It returns a string in the form: "`last,first`", where `last` and `first` are capitalized. For example, if `first` were "`alan`" and `last` were "`turing`", then `switch` would return "`Turing,Alan`". (*note: capitalized with no spaces*). If `first` and `last` were "`Florean`" and "`fortescue`", then `switch` would return "`Fortescue,Florean`".

3. Write a ***function*** that returns the longest common prefix of two strings. For example, the longest common prefix of `distance` and `disinfection` is `dis`. The header of this function is `def prefix(s1,s2):` If the two strings have no common prefix, the function returns an empty string. Also write a function called `main()` that prompts the user to enter two strings and displays the common longest prefix using the `prefix` function.

4. Secure Websites impose rules for passwords. Write a ***function*** that checks whether a string is a valid password which conforms to the follow rules:

   a. Must have at least eight characters
   b. Must consist of only letters and digits
   c. Must contain at least two digits

   Now write a program that prompts the user to enter a password and displays the string *Valid Password* if it meets the rules above, or *Invalid Password* if it does not.

5. Given `L1 = [1,2,3,1,2,3]` and `L2 = [2,4,6,8,10,12]`, write a ***function*** that is called using `L1` and `L2` as arguments and replaces every item in `L2` with itself, raised to the power of its positional companion in `L1`. When complete, `L2` will be `[2,16,216,8,100,1728]`. Your function must return no value. *Explain why we don't need this function to create and return a third list with the new computed values.*

6. Given two list variables, `lst1` and `lst2`, write a ***function*** that is called using these two lists as arguments and creates and returns a new, sorted list consisting of all the items in `lst1` that also appear in `lst2`. For example, if `lst1` is `[4,3,2,6,2]` and `lst2` is `[1,2,4]`, then the new list would be `[2,2,4]`. *Note: duplicate elements in `lst1` that appear in `lst2` are also duplicated in the new list.*

7. You have a variable named `incompletes` that contains a list of Midshipman alpha codes, and a variable `midID` which is a single alpha code (represented as a string). Write a ***function*** that takes `incompletes` and `midID` as parameters and counts and returns the number of times `midID` appears in `incompletes`.

8. Write a program that counts the number of characters, words and lines in a file. Words are separated by spaces. Your program should prompt the user to enter the name of the file. *Note: spaces are the only separators you need to process, so "`greetings!`" (with the exclamation point) would count as one word.*

9. Write a ***function*** that returns the greatest common divisor of integers in a list named `numbers`. Use the following function header: `def gcd(numbers):` Also write a test program that prompts the user to enter five numbers, appends each to a list, and calls the `gcd` function to find the GCD of these numbers.

10. The following function searches through a list of integers (called `nums`) until the target value (`x`) is found.

```
def search(x,nums):
   result = -1
   for i in range(len(nums)):
      if nums[i] == x: # item found, save the index value
         result = i
         break
   return result # loop finished, return the result
```

    If `x` is not found, then `-1` is returned. The difference between this approach and the use of `list.index(x)` is that `list.index(x)` raises an exception if the target value does not appear in the list. Change the code above to implement the search using `list.index(x)` and `try…except` to manage the operation of the function.

11. Write a program that prompts the user to enter a Social Security number in the format `ddd-dd-dddd`, where `d` is a digit. The program displays *Valid SSN* for a correct Social Security number or *Invalid SSN* otherwise.

12. Complete any of the class activities in: `cw25`, `cw26`, `cw27`, `cw28`, `cw35`