CS2302 Data Structures
Spring 2020
Python Warmup Exercise

Integers

1. Write the function divisible(a,b) that receives two positive integers a and b and determines if a is divisible by b.
2. Write the function prime(n) that receives a positive integer n and determines if it is a prime number.
3. Write the function sum_digits(n) that receives a positive integer n and returns the sum of the digits in n. For example, sum_digits(2302) should return 7.

Strings

4. Write the function reverse(s) that receives a string s and returns s backwards. For example, reverse('cat') should return 'tac'.
5. Write the function remove_vowels(s) that receives a string s and returns the string results from after removing all lowercase vowels from s. For example, remove_vowels('University') should return 'Unvrsty'.
6. Write the function pal(s) that receives a string s and determines if it is a palindrome (that it, it reads the same forward and backward).

1D Arrays

7. Write the function max_array(A) that receives a 1D array A and returns the maximum element in A.
8. Write the function find(A,x) that receives a 1D array A and a number x and returns position i of x in A, or -1,-1 if x is not in A.
9. Write the function sum_array(A) that receives a 1D array A and returns the sum of the elements in
10. Write the function replace_array(A,x,y) that receives a 1D array A and integers x and y, and returns the array resulting from replacing all occurrences of x by y in A (your function should not modify A).

2D Arrays

11. Write the function is_square(A) that receives a 2D array A and determines if A is square
12. Write the function diagonal_sum(A) that receives a square array A and returns the sum of the main diagonal of A.
13. Write the function sec_diagonal_sum(A) that receives a square array A and returns the sum of the secondary diagonal of A.
14. Write the function diagonal(A) that receives a square array A and returns a 1-D array containing the main diagonal of A.
15. Write the function sec_diagonal(A) that receives a square array A and returns a 1-D array containing the secondary diagonal of A.
16. Write the function swap_rows(A,i,j) that receives a 2D array A and integers i and j and returns the array resulting from swapping rows i and j in A (your function should not modify A).
17. Write the function swap_columns(A,i,j) that receives a 2D array A and integers i and j and returns the array resulting from swapping columns i and j in A (your function should not modify A).
18. Write the function replace_max_array(A,x) that receives a 2D array A and an integer x and returns the array resulting from replacing the largest element in A by x (your function should not modify A).

Lists

19. Write the function greater_than_list(L,x) that receives a list L and an integer x and returns a list containing the elements in L that are greater than x in the order they appear in L.
20. Write the function split(L) that receives a list L returns two lists, one containing the elements of L that have an even index ([L[0], L[2], etc.) and one containing the elements of L that have an odd index.
21. Write the function merge(L1,L2) that receives two sorted lists L1 and L2 and returns a sorted list with all the elements of L1 and L2 (this is the merge function in mergesort).
22. Write the function split_pivot(L) that receives a list of integers L and returns two lists, one containing the elements of L that are smaller than L[0] and one containing the elements of L that are greater or equal to L[0] appearing in the same order they appear in L (this is the split function in quick sort). For example, if L = [3,2,1,4,0], your function should return [2,1,0], [3,4].