**Problem 1**

Midterm due Jul 8, 2020 05:30 +06

 Bookmark this page

Problem 1-1

1.0/1.0 point (graded)

Suppose x = "pi" and y = "pie". The line of code x, y = y, x will swap the values of x and y, resulting in x = "pie" and y = "pi".

True

False

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Problem 1-2

1.0/1.0 point (graded)

Suppose x is an integer in the following code:

def f(x):

while x > 3:

f(x+1)

For any value of x, all calls to f are guaranteed to never terminate.

True

False

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Problem 1-3

1.0/1.0 point (graded)

A Python program always executes every line of code written at least once.

True

False

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Problem 1-4

1.0/1.0 point (graded)

Suppose you have two different functions that each assign a variable called x. Modifying x in one function means you always modify x in the other function for any x.

True

False

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Problem 1-5

0.0/1.0 point (graded)

The following code will enter an infinite loop for all values of i and j.

while i >= 0:

while j >= 0:

print(i, j)

True

False

incorrect

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Problem 1-6

0/1 point (graded)

It is always possible and feasible for a programmer to come up with test cases that run through every possible path in a program.

True

False

incorrect

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Incorrect (0/1 point)

Review

Problem 1-7

1.0/1.0 point (graded)

Assume f() is defined. In the statement a = f(), a is always a function.

True

False

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Problem 1-8

1.0/1.0 point (graded)

A program that keeps running and does not stop is an example of a syntax error.

True

False

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Problem 1-9

1.0/1.0 point (graded)

Consider the following function.

def f(x):

a = []

while x > 0:

a.append(x)

f(x-1)

A new object of type list is created for each recursive invocation of f.

True

False

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Problem 1-10

1.0/1.0 point (graded)

A tuple can contain a list as an element.

True

False

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

**Problem 2**

Midterm due Jul 8, 2020 05:30 +06

 Bookmark this page

Problem 2-1

1/1 point (graded)

Consider the statement: L = {'1':1, '2':2, '3':3}. Which is correct?

L is a list

L is immutable

L contains 6 elements

L has integer keys

L maps strings to integers

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Correct (1/1 point)

Review

Problem 2-2

0/1 point (graded)

Assume a break statement is executed inside a loop and that the loop is inside a function. Which of the following is correct?

The program might immediately terminate.

The function might immediately terminate.

The loop will always immediately terminate.

All of the above.

None of the above.

incorrect

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Incorrect (0/1 point)

Review

Problem 2-3

1/1 point (graded)

In Python, which of the following is a mutable object?

a string

a tuple

a list

all of the above

none of the above

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Correct (1/1 point)

Review

Problem 2-4

1/1 point (graded)

Assume the statement s[1024] = 3 does not produce an error message. This implies:

type(s) can be str

type(s) can be tuple

type(s) can be list

All of the above

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Correct (1/1 point)

Review

Problem 2-5

0/1 point (graded)

Consider the code:

L = [1,2,3]

d = {'a': 'b'}

def f(x):

return 3

Which of the following does NOT cause an exception to be thrown?

print(L[3])

print(d['b'])



for i in range(1000001, -1, -2):

print(f)

print(int('abc'))

None of the above

incorrect

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Incorrect (0/1 point)

Review

Problem 2-6

3/3 points (graded)

Examine the following code snippet:

stuff = \_\_\_\_\_

for thing in stuff:

if thing == 'iQ':

print("Found it")

Select all the values of the variable "stuff" that will make the code print "Found it".

["iBoy", "iGirl", "iQ", "iC","iPaid","iPad"]

("iBoy", "iGirl", "iQ", "iC","iPaid","iPad")

[ ( "iBoy", "iGirl", "iQ", "iC","iPaid","iPad") ]

( [ "iBoy", "iGirl", "iQ", "iC","iPaid","iPad" ], )

["iQ"]

"iQ"

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Correct (3/3 points)

Review

Problem 2-7

2/2 points (graded)

The following Python code is supposed to compute the square of an integer by using successive additions.

def Square(x):

return SquareHelper(abs(x), abs(x))

def SquareHelper(n, x):

if n == 0:

return 0

return SquareHelper(n-1, x) + x

Not considering recursion depth limitations, what is wrong with this implementation of procedure Square? Check all that apply.

It is going to return a wrong value.

The term Square is a reserved Python keyword.

Function names cannot start with a capital letter.

The function is never going to return anything.

Python has arbitrary precision arithmetic.

This function will not work for negative numbers.

The call SquareHelper(abs(x), abs(x)) won't work because you can't have abs(x) as both parameters.

Nothing is wrong; the code is fine as-is.

correct

Submit

You have used 1 of 1 attemptSome problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

Correct (2/2 points)

Review

Problem 3

0/10 points (graded)

Write a recursive Python function, given a non-negative integer N, to count and return the number of occurrences of the digit 7 in N.

For example:  
count7(717) -> 2  
count7(1237123) -> 1  
count7(8989) -> 0

Hint: Mod (%) by 10 gives you the rightmost digit (126 % 10 is 6), while doing integer division by 10 removes the rightmost digit (126 // 10 is 12).

This function has to be recursive; you may not use loops! This function takes in one integer and returns one integer.

def count7(N):

'''

N: a non-negative integer

'''

# Your code here

def count7(N):

'''

N: a non-negative integer

returns: no. of occurrences of digit 7 in the number N

'''

if N == 0:

return 0

elif N % 10 == 7:

return 1 + count7(N // 10)

else:

return 0 + count7(N // 10)

Write a Python function that returns the sum of the pairwise products of listA and listB. You should assume that listA and listB have the same length and are two lists of integer numbers. For example, if listA = [1, 2, 3] and listB = [4, 5, 6], the dot product is 1\*4 + 2\*5 + 3\*6, meaning your function should return: 32

This function takes in two lists of numbers and returns a number.

def dotProduct(listA, listB):

'''

listA: a list of numbers

listB: a list of numbers of the same length as listA

'''

# Your code here

def dotProduct(listA, listB):

'''

listA: a list of numbers

listB: a list of numbers of the same length as listA

'''

# Your code here

s=0

for i in range(len(listA)):

s +=listA[i]\*listB[i]

return s

**Problem 5**

Midterm due Jul 8, 2020 05:30 +06

 Bookmark this page

Problem 5

0.0/20.0 points (graded)

Write a Python function that returns a list of keys in aDict that map to integer values that are unique (i.e. values appear exactly once in aDict). The list of keys you return should be sorted in increasing order. (If aDict does not contain any unique values, you should return an empty list.)

This function takes in a dictionary and returns a list.

def uniqueValues(aDict):

'''

aDict: a dictionary

'''

# Your code here

### def uniqueValues(aDict):

### '''

### You are given a dictionary aDict that maps integer keys to integer values.

### Write a Python function that returns a list of keys in aDict that map to dictionary values that appear exactly once in aDict.

### This function takes in a dictionary and returns a list.

### Return the list of keys in increasing order.

### If aDict does not contain any values appearing exactly once, return an empty list.

### If aDict is empty, return an empty list.

### For example:

### If aDict = {1: 1, 3: 2, 6: 0, 7: 0, 8: 4, 10: 0} then your function should return [1, 3, 8]

### If aDict = {1: 1, 2: 1, 3: 1} then your function should return []

### '''

### uList = []

### tempList = []

### lastList = []

### for value in aDict.values():

### if value not in uList:

### uList.append(value)

### else:

### tempList.append(value)

### for i in tempList:

### if i in uList:

### uList.remove(i)

### for key, value in aDict.items():

### if value in uList:

### lastList.append(key)

### lastList.sort()

### return lastList

### Problem 6

18.571428571428573/20 points (graded)

Implement a function that meets the specifications below.

def max\_val(t):

""" t, tuple or list

Each element of t is either an int, a tuple, or a list

No tuple or list is empty

Returns the maximum int in t or (recursively) in an element of t """

# Your code here

For example,

* max\_val((5, (1,2), [[1],[2]])) returns 5.
* max\_val((5, (1,2), [[1],[9]])) returns 9.

Paste your entire function, including the definition, in the box below. Do not leave any debugging print statements.

def max\_val(t):

'''

def max\_val(t):

t, tuple or list

Each element of t is either an int, a tuple, or a list

No tuple or list is empty

Returns the maximum int in t or (recursively) in an element of t

For example,

max\_val((5, (1,2), [[1],[2]])) returns 5.

max\_val((5, (1,2), [[1],[9,5,7]])) returns 9.

Paste your entire function, including the definition, in the box below. Do not leave any debugging print statements.

'''

assert t == ("") or [""], "t can not be empty"

nList = []

for sub in t:

if type(sub) == int:

nList.append(sub)

else:

for item in sub:

if type(item) != int:

for elem in item:

nList.append(elem)

else:

nList.append(item)

return max(nList)

### Problem 7

20/20 points (graded)

Write a Python function called satisfiesF that has the specification below. Then make the function call run\_satisfiesF(L, satisfiesF). Your code should look like:

def satisfiesF(L):

"""

Assumes L is a list of strings

Assume function f is already defined for you and it maps a string to a Boolean

Mutates L such that it contains all of the strings, s, originally in L such

that f(s) returns True, and no other elements. Remaining elements in L

should be in the same order.

Returns the length of L after mutation

"""

# Your function implementation here

run\_satisfiesF(L, satisfiesF)

For your own testing of satisfiesF, for example, see the following test function f and test code:

def f(s):

return 'a' in s

L = ['a', 'b', 'a']

print(satisfiesF(L))

print(L)

Should print:

2

['a', 'a']

Paste your entire function satisfiesF, including the definition, in the box below. **After you define your function, make a function call to run\_satisfiesF(L, satisfiesF). Do not define f or run\_satisfiesF.**Do not leave any debugging print statements.

**For this question, you will not be able to see the test cases we run. This problem will test your ability to come up with your own test cases.**

def satisfiesF(L):

"""

Assumes L is a list of strings

Assume function f is already defined for you and it maps a string to a Boolean

Mutates L such that it contains all of the strings, s, originally in L such

that f(s) returns True, and no other elements. Remaining elements in L

should be in the same order.

Returns the length of L after mutation

"""

# Your function implementation here

i=0

while len(L)> i:

if f(L[i]): i +=1

else: L.pop(i)

return len(L)

run\_satisfiesF(L, satisfiesF)