CS 61ASummer 2017

Structure and Interpretation of Computer Programs

Quiz 8 Solutions

INSTRUCTIONS

- You have 10 minutes to complete this quiz.
- The exam is closed book, closed notes, closed computer, closed calculator.
- The final score for this quiz will be assigned based on **effort** rather than correctness.
- Mark your answers on the exam itself. We will not grade answers written on scratch paper.
- For multiple choice questions,

 means mark all options that apply
 - \bigcirc means mark a single choice

Last name		
First name		
Student ID number		
CalCentral email (_@berkeley.edu)		
Teaching Assistant	Alex Stennet	○ Kelly Chen
	O Angela Kwon	Michael Gibbes
	Ashley Chien	O Michelle Hwang
	O Joyce Luong	O Mitas Ray
	Karthik Bharathala	O Rocky Duan
	○ Kavi Gupta	O Samantha Wong
Name of the person to your left		
Name of the person to your right		
All the work on this exam is my own. (please sign)		

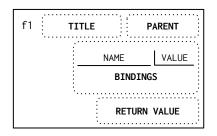
1. (5 points) All Summer Seventeen

(a) On the next page, fill in the environment diagram that results from executing the code below until the entire program is finished, an error occurs, or all frames are filled.

You may not need to use all of the spaces or frames.

(b) Then, for each region below, fill in the corresponding bubble. Leave a row blank if the space in the environment diagram should be left blank.

To receive credit, you must list your bindings in the order in which they are first bound in the frame.



FRAME	FIELD	Names Value							JUES						
	Binding 1	sum		func	func sum(lst) [parent=Global]										
Global	Binding 2	(a					$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Binding 3	$\bigcirc \ \mathrm{help}$	\bigcirc lst	\bigcirc me	\bigcirc total	○ you	$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
f1	Title	sum													
	Binding 1	\bigcirc help	lst	\bigcirc me	\bigcirc total	O you	$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	• fig.	
	Binding 2	$\bigcirc \ \mathrm{help}$	\bigcirc lst	\bigcirc me	total	○ you	$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Binding 3	help	\bigcirc lst	\bigcirc me	\bigcirc total	○ you	$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	\bullet α	\bigcirc fig.	
	Binding 4	$\bigcirc \ \mathrm{help}$	\bigcirc lst	• me	\bigcirc total	○ you	$\bigcirc 0$	\bigcirc 1	2	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Binding 5	$\bigcirc \ \mathrm{help}$	\bigcirc lst	\bigcirc me	\bigcirc total	○ you	$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Return						$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Title	help	O sur	n											
f2	Binding 1	$\bigcirc \ \mathrm{help}$	\bigcirc lst	\bigcirc me	\bigcirc total	you	• 0	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Binding 2	$\bigcirc \ \mathrm{help}$	\bigcirc lst	\bigcirc me	\bigcirc total	○ you	$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Binding 3	$\bigcirc \ \mathrm{help}$	\bigcirc lst	\bigcirc me	\bigcirc total	○ you	$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Return						$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	None	$\bigcirc \alpha$	\bigcirc fig.	
	Title	help	O sur	n											
f3	Binding 1	$\bigcirc \ \mathrm{help}$	\bigcirc lst	\bigcirc me	\bigcirc total	you	$\bigcirc 0$	1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Binding 2	$\bigcirc \ \mathrm{help}$	\bigcirc lst	\bigcirc me	\bigcirc total	○ you	$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Binding 3	$\bigcirc \ \mathrm{help}$	\bigcirc lst	\bigcirc me	\bigcirc total	O you	$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Return						$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	None	$\bigcirc \alpha$	○ fig.	
	Title	\bigcirc help	O sur	n											
f4	Binding 1	$\bigcirc \ \mathrm{help}$	\bigcirc lst	\bigcirc me	\bigcirc total	O you	$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Binding 2	$\bigcirc \ \mathrm{help}$	\bigcirc lst	\bigcirc me	\bigcirc total	○ you	$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Binding 3	$\bigcirc \ \mathrm{help}$	\bigcirc lst	\bigcirc me	\bigcirc total	○ you	$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	
	Return						$\bigcirc 0$	\bigcirc 1	$\bigcirc 2$	\bigcirc 6	\bigcirc 7	○ None	$\bigcirc \alpha$	\bigcirc fig.	

Name: _____

DO NOT TURN IN THIS PAGE.

3

Remember to fill out the choices on the previous page to receive credit for this quiz.

A complete answer will:

- Add all missing names and parent annotations to all local frames.
- Add all missing values created or referenced during execution.
- Show the return value for each local frame.

```
Global
    def sum(lst):
1
2
3
4
5
6
7
8
9
                                                                   sum
                                                                                   func sum(lst) [parent=Global]
        total = 0
def help(you):
            nonlocal total
total += lst[you]
                                                                      a
                                                                            7
            lst[you] = total - lst[you]
        me = 0
        while me < len(lst):
            help(me)
me += 1
10
11
                                             f1: Sum
                                                           _ [parent=<u>Global</u>]
                                                                                    list
        return total
12
13
                                                                   lst
    a = sum([6, 1])
                                                                total
                                                                                  → func help(you) [parent=f1]
                                                                  help
                                                                    me
                                                                            2
                                                          Return Value
                                             f2: help [parent= f1
                                                                   you
                                                          Return Value | None
                                             f3: help [parent= f1
                                                                   you
                                                          Return Value | None
                                             f4: __
                                                            [parent=_
                                                          Return Value
```