**COMP130 HW4: Functions and the call stack  
instructor: John MacCormick**

Consider the following Python code:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | def check\_string(s):  if s.isdigit():  print(s, 'consists only of digits')  elif s.isalpha():  print(s, 'is purely alphabetic')  else:  print(s, 'is neither purely digits nor purely alphabetic')  def concat\_then\_check(a, b):  c = a + b  check\_string(c)  p = 'COMP'  q = '131'  check\_string(p)  check\_string(q)  concat\_then\_check(p, q) |

Question 1. (10 points) Draw the complete stack diagram that results from the function call at line 18. The stack diagram must use the same format as our textbook. Hint: there should be three levels in the call stack. Suggestion: try to solve this without using the IDLE debugger, then later check your answer using the debugger if desired.

Question 2. (5 points) Suppose we add a new line at the end of this script, containing the statement print(c). When the program is run, it now produces an error. Explain the reason for this error in your own words.

Question 3. (10 points) Step through the code above in the IDLE debugger, until you reach a point where there are three frames on the call stack. Give the complete description of the lowest frame provided by the IDLE debugger at this point. Hint: It will begin with '\_\_main\_\_'.check\_string().

Question 4. (15 points)

Consider the program below. This program has been written so that it is confusing to read. Its purpose is not to do any useful computation, but rather to give you practice tracing the execution of a program and creating stack diagrams. Attempt this question by analyzing the code manually first, then check your answers by stepping through with the IDLE debugger.

def fun3(a,b):

a = 2\*b+a # Line D

print("fun3: a=" + str(a) + " b=" + str(b))

def fun2(b, a):

b = a+b

a = b-a # Line C

fun3(b,a)

def fun1(a):

a = a+1 # Line B

fun2(3\*a, a)

a=3 # Line A

fun1(a)

(a) Draw a stack diagram for the program just after Line A has executed. This diagram should include only the \_\_main\_\_ frame.

(b) Draw a new stack diagram that shows the state of the program's execution just after Line B has executed.

(c) Draw a new stack diagram that shows the state of the program's execution just after Line D has executed.

Question 5. (5 points) Indicate whether each of the following functions is *void* or *fruitful*. Hint: Conduct experiments by printing the output of each function.

(a) random.seed  
(b) random.randint  
(c) turtle.pos  
(d) turtle.forward  
(e) print

Question 6. (15 points) Write a function named print\_centered\_text takes two paramters: a string text and an integer width. The function then prints out the text centered in the specified width. For example, the program

print\_centered\_text("This text is", 20)

print\_centered\_text("centered", 20)

print\_centered\_text("in twenty spaces", 20)

will generate the output:

This text is

centered

in twenty spaces

Hints:

* You will need the len function, which returns the length of a string. Example: len('abcd') is 4.
* It may help to know how to print on the same line from multiple print functions. See the supplementary study guide for details, but the basic idea is to set the end parameter of print to the empty string, as in this example: print('hi ', end='')

Question 7. (30 points) Based on the assigned reading and in-class discussion for our class on Computing and Algorithms, write 200-300 words responding to both parts of the following prompt:

(a) In what ways is an algorithm similar to a cooking recipe, and in what ways is it different? (b) Give at least one example of an algorithm that affects our society but is not mentioned in the assigned reading.

Note: You are not expected to do additional research or use external sources other than the assigned reading and your notes from our discussion when answering the above question. If you do use an external source, you must cite it. This includes generative AI programs.

Total points on assignment: 90