We completed the trace of the program by building the run-time stack. (from Lecture 7’s in-class handout)

```
# Program to demonstrate function calls

def main():
    value = 99  # local variable
    print 'In main before call: value is', value
    change_me(value)  # location (**)
    print 'Back in main: value is', value

def change_me(arg):
    value = 10  # local variable
    arg = 3
    print 'In change_me: arg is', arg,
    print 'and value is', value
    value = more_change(value, arg)  # location (***)
    print 'Back in change_me: arg is', arg,
    print 'and value is', value

def more_change(arg, arg2):
    print 'In more_change: value is', value
    print 'In more_change: arg is', arg,
    print 'and arg2 is', arg2
    arg = arg2
    print 'Here, value', value
    return arg2 * 5

# Global variable
value = 5
# Call the main function.
main()  # location (*)
print 'Here, value', value
```

Actual Output of Program:

```
IDLE 1.2.1      ==== No Subprocess ====
>>> In main before call: value is 99
In change_me: arg is 99
In change_me: arg is 3 and value is 10
In more_change: value is 5
In more_change: arg is 10 and arg2 is 3
in more_change: arg is 3 and arg2 is 3
Back in change_me: arg is 3 and value is 15
Back in main: value is 99
Here, value is 5
```
2. We worked together to design a program to solve the following program (Project 11 from Chapter 3.)

In the game of Lucky Sevens, the player rolls a pair of dice. If the dice add up to 7, the player wins $4; otherwise, the player loses $1. Suppose that, to entice the gullible, a casino tells players that there are lots of ways to win: (1, 6), (2, 5), etc. A little mathematical analysis reveals that there are not enough ways to win to make the game worthwhile; however, because many people’s eyes glaze over at the first mention of mathematics, your challenge is to write a program that demonstrates the futility of playing the game. Your program should take as input the amount of money that the player wants to put into the pot, and play the game until the pot is empty. At that point, the program should print the number of rolls it took to break the player, as well as maximum amount of money in the pot.

a) What would the user’s interaction with the program look like?

Welcome to Lucky Sevens! The rules are....

How much money do you want to put into the pot? 20
< a trace/log of the play would go here >

You lost all of your money after 34 rolls.
The maximum size of pot during play was $25.

b) We want the main function to act as an outline of the program and contain at most:
• the “main loop”: What would be the main loop for this program?

We decided (well actually I did) that a loop in the main would be too cluttered.

• function calls to perform difficult subproblems. What high-level subproblems does our program need to perform? (Think about what arguments each subprogram needs to be passed or user input needed, and what type of information is returned to the caller)

We pretty much wrote the following program with the function playTheGameUntilPotIsGone remaining to be completed. (I am guessing at the actual function name we used in class--I did not write them down...)

SEE NEXT PAGE FOR PARTIAL PROGRAM
File: luckySevens.py
Author: Mark Fienup with design help from Intro. class
Description: Simulates a user playing games of Lucky Sevens until they lose all of their money. In Lucky Sevens the player rolls a pair of dice. If the dice add up to 7, the player wins $4; otherwise, the player loses $1.

Input: the user will be asked to enter the amount of money in the initial pot.

Output: A trace/log of play until the pot is empty. At that point, the program prints the number of rolls it took to break the player, as well as maximum amount of money in the pot.

```python
import random

def main():
    """ Simulates the playing of Lucky Seven until the user breaks""
    printWelcomeAndRules()
    initialPot = input("Enter the amount of the initial pot: ")
    rolls, maxPot = playTheGameUntilPotIsGone(initialPot)
    print "The number of rolls it took to lose all $", initialPot, 
    "was", rolls, "."
    print "The maximum amount of money in the pot during play was", maxPot

def printWelcomeAndRules():
    """Prints a welcome and rules of Lucky Seven""
    print "Welcome to Lucky Sevens Simulator!\n"
    print "We'll simulate you playing games of Lucky Sevens"
    print "until you lose all of your money.\n"
    print "After the simulation, we'll print the number of rolls"
    print "it took for you to lose all your money, and "
    print "the maximum amount of money in the pot.\n"

def sumOfPairOfDice():
    """ Random rolls a pair of 6-sided dice and returns their sum""
    die1 = random.randint(1,6)
    die2 = random.randint(1,6)
    return die1 + die2

def playTheGameUntilPotIsGone(pot):
    """ Simulates playing Lucky Seven until the pot is gone""

main()
raw_input("Hit any key to close")
```