Computer Science 1 — CSci 1100 — Spring 2017
Exam 1
February 13, 2017

SOLUTIONS

1. (**15 points total; 5 points each**) Assume each of the following is an individual program. Write the **exact** output of these programs in the area provided. Use the scratch area as you wish. There are **no syntax errors** in the code provided in this question.

Part a:	**Solution:**	Scratch area:

`import math`
`y = 125`
`y = int(y // 5 / 3) // 4`
`print(y)`
`print(7 + 1 * 2 ** 2 ** 3)`
`z = 12 + 2 // 7 - 4`
`z += 12`
`print(z)`
`print(math.sqrt(2 * 2 ** 3))`
`x = 17`
`z = 18`
`print(x // z * (7 - 4))`
	`2`
	`263`
	`20`
	`4.0`
	`0`	

Part b:	**Solution:**	Scratch area:

`a = 'b'`
`b = 'c'`
`c = 'd'`
`print(a + b + c)`
`print(a, b, c)`
`print(4 * a + b)`
`print('onomatopaeia'.`
`  replace('on','oon').count('o'))`
`print('wat\nerf\nall' + 3 * 'o')`
	`bcd`
	`b c d`
	`bbbbc`
	`4`
	`wat`
	`erf`
	`allooo`	

Part c:	**Solution:**	Scratch area:

`def f(a):`
`  print('f')`
`  return 4*a`
 	
`def g(a):`
`  if len(a) < 3:`
`  return a+'bababa'`
`  elif len(a) == 3:`
`  return(a)`
`  else:`
`  return a.count('oo')`
`  return "Done"`
 	
`s = g('goo')`
`print(s)`
`t = f(s)`
`print(t)`
`print(g(t))`
	`goo`
	`f`
	`googoogoogoo`
	`4`	

2. (**10 points total; 2 points each**) Assume each of the following is a complete, separate program. For each program find the **first** error that prevents the program from generating output, or decide there is no error. If there is an error, write in the solution box that there is an error, write the line number **and** very briefly describe the error. If there program would run and generate output, write **No Error**.

Part a:

```
def f123_45(x)                          #1
    print(x, sep="", end="")            #2
    return x.count('9')                 #3
print(f123_45('678'))                   #4
```

Solution:

```
#1 def needs a : at the end
```

Part b:

```
import math                             #1
def f1(a, b, c):                        #2
    p = a * b * c                       #3
    p = math.sqrt(p)                    #4
    return p                            #5
print("{:d}".format(f1(3,3,3,3)))       #6
```

Solution:

```
#6 function f1 expects 3 arguments,
but was given 4
```

Part c:

```
def a(b, c, d):                         #1
    return b + c + d                    #2
print(a(1, 2, 3)*'#')                   #3
print(a(4/2, 3, 5)*'8')                 #4
```

Solution:

```
Error on #4 cannot multply a
string by a non-integer
```

Part d:

```
def v(l, w, h):                         #1
    if l*w < w*h:                       #2
        return l * w                    #3
    elif w*h < h*l:                     #4
        return w*h                      #5
    else:                               #6
    return h*l                          #7
print(v(7, 7, 7))                       #8
```

Solution:

```
Error on #7: Expecting an indented
block
```

Part e:

```
' "Right,",he said."We\'ll call." '     #1
'Right, he said."We\'ll call.'          #2
"Right,\t\nhe said." 'We\'ll call\\it.' #3
"\"Right,\"he said.\"We'll call.\" "    #4
""" "Right,"he said."We'll call."""     #5
"Right, he said." 'We\'ll call.'        #6
```

Solution:

```
No errors
```

3. (**15 points total; 3 points each**) Given the string assignment:

```
a = "This is an ex parrot!!"
```

Give a single line of code to accomplish each of the following. You can assume you are typing into the Python interpreter and can omit print statements. Be sure you use the string variable `a` in your code and not the hardcoded string value. Make sure your solution is only one line. Answers of more than one line will not get credit.

Part a:	**Solution:**
Return the string in all capital letters. For **a**, this would give: `'THIS IS AN EX PARROT!!'`	`a.upper()`
Part b:	**Solution:**
Return the ratio (as a float) of the length of **a** divided by the number of times 'is' appears in the string. For **a**, this would give: `11.0`	`len(a)/a.count('is')`
Part c:	**Solution:**
With each word capitalized and all spaces removed. For **a**, this would give: `'ThisIsAnExParrot!!'`	`a.title().replace(' ', '')`
Part d:	**Solution:**
Return a new string using only the first, twelfth, thirteenth, ninth, and fourth letters. Repeat the resulting string 3 times. For **a**, this would give: `'TexasTexasTexas'`	`(a[0]+a[11]+a[12]+a[8]+a[3])*3`
Part e:	**Solution:**
Return a string all in lower case, with the exception of all occurrences of 'i' or 'r' which should be in upper case. For **a**, this would give: `'thIs Is an ex paRRot!!'''''`	`a.lower().replace('i', 'I').replace('r','R')`

4. (**10 points**) A quart is about 946 cubic centimeters. Write a short but complete segment of code that calculates and outputs the height of a cylinder – accurate to 2 decimal places – that has a radius of 3 centimeters and that holds a quart. It must use the `math` module and the string `format` method. Your output should be exactly

   ```
   The cylinder would have to be 33.46 cm tall
   ```

   Remember that the volume of a cylinder is `v = pi * r**2 * h`

   Solution:

```
import math
h = 946 / (3**2 * math.pi)
print("The cylinder would have to be {:.2f} cm tall".format(h))
```

5. (**12 points**) Suppose you are renting a limousine for a day of celebration – maybe to take in a Yankee's game. The cost for the limo is $93 per whole hour and $2 per minute up to the maximum per minute charge of $93. The trip from RPI to the venue takes `h1` hours and `m1` minutes, the game runs `h2` hours and `m2` minutes and the trip back to RPI takes `m3` minutes more than the trip down. Assuming that the variables `h1`, `m1`, `h2`, `m2`, and `m3` have already been assigned integer values, write code to calculate and print the total number of hours and minutes you had the limousine and the amount you owe the driver.

   For example, if you are given

```
h1 = 2
m1 = 42
h2 = 3
m2 = 11
m3 = 22
```

   then your output should be

```
Outing took 8 hours and 57 minutes
Total cost is $837
```

   You can use a hardcoded values for $93 and $2, but be sure to use the variables names for the hours and minutes.

   Solution:

```
total_minutes = 2 * (h1 * 60 + m1) + h2 * 60 + m2 + m3
hours = total_minutes // 60
minutes = total_minutes % 60
cost = 93 * hours + min(2 * minutes, 93)
print("Outing took {:d} hours and {:d} minutes".format(hours, minutes))
print("Total cost is ${:d}".format(cost))
```

6. (**12 points**) Write a function called `calc_bias` that takes as parameters five numbers. The function should calculate the average of the five values and then determines bias as follows:

   (a) if the average is closer to the largest of the five numbers than to the smallest, the function should print `Biased High`.

   (b) if the average is closer to the smallest of the five numbers, the function should print `Biased Low`.

   (c) Otherwise, the function should print `No Bias`.

   In any event the function must return the average of the five numbers.

```
print(calc_bias(5, 2, 1, 4, 3))
print(calc_bias(2, 1, 6, 4, 3))
print(calc_bias(4, 2, 3, 1, 4))
```

would output

```
No Bias
3.0
Biased Low
3.2
Biased High
2.8
```

In the first case, 5 and 1 are both 2 units from the average of 3. In the second case, 1 is 2.2 units and 6 is 2.8 from the average of from the average of 3.2, so the result is biased low. In the third case 1 is 1.8 units and 4 is 1.2 units from the average of 2.8, so the result is biased high.

Solution:

```
def calc_bias(a, b, c, d, e):
    avg = (a+b+c+d+e) / 5
    if (max(a, b, c, d, e) - avg < avg - min(a, b, c, d, e)):
        print('Biased High')
    elif (max(a, b, c, d, e) - avg > avg - min(a, b, c, d, e)):
        print('Biased Low')
    else:
        print('No Bias')
    return avg
```

7. (**14 points**) Bring the mouse to the cheese. Write a program that asks the user for a height and a length, of two corridors of a maze, and then prints out the maze using `'*'` for the corridors. Do not use loops.

   Here is an example of running the program:

```
Enter the horizontal length ==> 5
Enter the vertical length ==> 7
Mouse*****
         *
         *
         *
         *
         *
         *
         *
         Cheese
```

Solution:

```
hor = int(input("Enter the horizontal length ==> "))
vert = int(input("Enter the vertical length ==> "))
print("Mouse", "*"*hor, sep='')
spaces = (hor+len("Mouse")-1) * " "
print((spaces + "*\n") * vert, end='')
print(spaces + "Cheese", sep='')
```

8. (**12 points**) Write a program that asks the user for a string and then decodes the string into an integer as follows:

   (a) the hundreds digit is encoded as the number of `*`s in the string

   (b) the tens digit is encoded as the number of `!`s in the string

   (c) the ones digit is encoded as the number of `@`s in the string

   You can assume that the string is valid (there will not be more than 9 occurences of `*`, `!`, or `@` in the string.) All characters other than `*`, `!`, and `@` can be ignored. Here is an example of running the program:

```
Enter a code: 6@@r8w*8^^*73**56!
Your number is 412
```

Solution:

```
string = input("Enter a code: ")
val = 0
val += 100 * string.count('*')
val += 10 * string.count('!')
val += 1 * string.count('@')
print("Your number is", val)
```