1200 Midterm Spring 2018 — Practice for Fall 2018

Code Tracing (Variables)

For each snippet of code below, write the value of each variable and its type.

`# Example` `x = 2` `y = 3` `y = 2 * 3 * 1.0`	x y *2/int* - *2/int* *3/int* *2/int* *6.0/float*
`a = "Nihar"` `b = "Peter"` `c = 2` `a = a * c + str(c)` `b = b + str(c - 1)`	a b c
`# user enters the number "7.2"` `n = input("Temperature: ")` `n = float(n) // 1` `judge = n < 40` `judge = not judge or n > 7` `r = "Temp is: " + str(judge)`	n judge r

Code Tracing (Output)

Write the output of the code snippets below:

`# Example` `print("hello")` `print(1 + 3)`	Output *hello* *4*
`s = 4` `e = 10`	Output

`` times = 0 for num in range(s, e): if num + s < e: print("pow!" + str(num)) times = times + 1 print(times) ```	
`` w = "purple" x = 0 for index in range(len(w)): print(w[x]) print(w[index]) ```	Output
`` p1 = "hermione" p2 = "hogwarts" if len(p1) == len(p2) and p1[0] == p2[0]: print("Here's your letter") p1 = "harry" if p1[0] == p2[0] and len(p1) < len(p2): print("Hoot! Owl! Hoot!") ```	Output

Function Mysteries

You are given the following functions. Call the function. What do you expect the results to be?
If there is a return value, remember to save it in a variable.
If there is no return value, leave the "Return value" column blank.
If there is no output printed to the console, leave the "Output" column blank.
You should pass the functions values that make sense and will not cause errors as parameters.

All functions must either return or print something.
You may not use any casting functions.

You may <u>only</u> use the following options as values for the parameters, <u>each one a maximum of one time</u>. You will not use all values in this table.

True **False**	**2** **5** **8** **11**	**"narwhal"** **"art"** **"dandelion"**

Function	Function call	Return value	Output
`def func1(x):` `  return x == 1`	*result = func1(7)*	*False*	
`def func2(name):` `  print("Hello " + name)`	*func2("Sal")*		*Hello Sal*
`def func3(m1, m2, a):` `  if m1 and m2:` `  return a * 1.0` `  elif m2 and a < 10:` `  return a / 2` `  else:` `  return (a + 1) / 2`			
`def func4(x, y):` `  for i in range(x, y):` `  j = (y - x) - i` `  print(str(i) + ":" + str(j))`			

def func5(b, c): s = b t = c if len(c) > len(b): s = c t = b r = "" for i in range(len(s) - len(t)): r = r + s[i + len(t) - 1] return r			

Writing Functions

1) Write a function, `name_box`, that takes one parameter, `name`, and prints out a name box of the specified size. Notice that when the name has an even length,there is no middle row so the name appears on the row “after” the middle.

Examples:

Function call	output	Function call	output
`name_box("")`	`**` `**`	`name_box("K")`	`***` `*K*` `***`
`name_box("An")`	`****` `*  *` `*An*` `****`	`name_box("Ace")`	`*****` `*   *` `*Ace*` `*   *` `*****`
`name_box("Fran")`	`******` `*    *` `*    *` `*Fran*` `*    *` `******`	`name_box("Frank")`	`*******` `*     *` `*     *` `*Frank*` `*     *` `*     *` `*******`

2) Write a function, `rotate`, that takes a string, `target`, and an integer, `n`, and rotates the string the desired amount before **returning** it. A rotation is performed by taking the `n` characters at the beginning of the string and placing them, in the same order, at the end of the string.

You may not use string slicing to implement this function. You may not use any string methods other than the `[]` accessor and `+` to concatenate strings.

Examples:

Function call (underlining used to emphasize how the letters change)	Return value (you do not need to reproduce underlines)
`rotate("hello", 1)`	`"elloh"`
`rotate("hello", 3)`	`"lohel"`
`rotate("Cat curling", 8)`	`"ingCat curl"`
`rotate("Peter Rock", 6)`	`"RockPeter "`
`rotate("hello", 0)`	`"hello"`

Notice that a rotation of 0 does not change the string. You may assume that `0 <= n < len(target)`.

Writing a Program

Write a complete program, including a `main` function and a call to `main` after you have defined it, that asks the user how many labels they would like to make, then, for each label, asks the user 1) what name to print on the label and 2) the rotation before printing the rotated name in a box of stars to create the label.

You may find calling one or more of your previous functions useful (you don't need to copy them down here again). You may assume you have access to correct implementations of all functions in the "Writing Functions" section.

Example 1: output (user input **underlined bold**)	Example 2: output
How many labels? **2** What name? **Chai** Rotation? **1** ****** * * * * *haiC* * * ****** What name? **Ben** Rotation? **0** ***** * * *Ben* * * *****	How many labels? **3** What name? **Cat** Rotation? **2** ***** * * *tCa* * * ***** What name? **T** Rotation? **0** *** *T* *** What name? **C** Rotation? **0** *** *C* ***