Sample Written Exam #2

CS 149 Spring 2022

Writing Loops

1. In the following code, fill in the blank using the **range** function so that the program prints:

```
[1, 3, 5]
```

```
numbers = []
for i in ______:
    numbers.append(i)
print(numbers)
```

2. Fill in the blank so that the following code prints the sum of the first 5 odd numbers:

```
i = 5
sum = 0
while i > 0:
    sum += 2 * i + 1
    ________
print(sum)
```

3. Fill in the blank so that the following code prints:

```
1
12
123
1234
12345
```

```
for i in range(1,6):
    for j in range(1, i + 1):
        _________________________
    print()
```

4. True/False

a. ______ An if statement may be followed by any number of else statements.

b. ______ A for loop is more appropriate than a while loop for iterating over a range of integers.

c. ______ "3" == 3

d. ______ You should indent blocks of code you consider important.

e. ______ The following expression is true if and only x is equal to 6 or equal to 15.

```
x == 6 or 15
```

f. ______ An equality operator always results in a boolean value.

g. ______ Some loops will never terminate (stop looping).

h. ______ '<' and '>' are examples of relational operators.

5. Evaluating Expressions

Complete the following table

EXPRESSION	VALUE	TYPE
`True and True or False`		
`5 > 4 or 4 > 5`		
`not (5 <= 4 and 4 <= 5)`		
`‘Tree’ in [‘Tree’, ‘House’]`		
`f’{7 + 14 < 20}’`		
`‘A’ in {20: ‘A’, 30: ‘B’}`		

Tracing Code

6. Given the two functions defined below, answer the following questions.

```
def f(x):
    if x < 0:
        return 0
    elif x < 10:
        return x
    elif x > 20:
        return 2 * x
    else:
        return 10


def g(x,y):
    a = 0
    while x < y:
        a += f(x + y)
        x += 1
        y -= 1
    return a
```

   a. What is the value of g(1, 4)?

   b. What is the value of f(13)?

   c. What is the value of g(5, 1)?

   d. What is the value of g(10, 11)?

More Tracing Code

7. Given the two functions defined below, answer the following questions.

```
def f(x):
    if len(x) > 0:
        return x[0]
    else:
        return 'x'

def g(x, y):
    if x > 0:
        return f(y) * x
    else:
        return f(y)

def h(x, list_y):
    a = ''
    for i in list_y:
        a += g(x, str(i))
        x -= 1

    return a
```

a. What is the value of h(0, [1, 2])?

b. What is the value of h(3, ['a', 'b', '', 'd'])?

c. What is the value of h(-1, [])?

d. What is the value of h(4, 'spam')?

Writing a Loop

8. Your former coworker left you with the following code in one of your products:

```
def print_squares():
      print(f"1 squared = 1")
      print(f"2 squared = 4")
      print(f"3 squared = 9")
      print(f"4 squared = 16")
      print(f"5 squared = 25")
      print(f"6 squared = 36")
      print(f"7 squared = 49")
      print(f"8 squared = 64")
```

This code works well enough -- it prints the first eight perfect squares as it should -- but it's pretty inelegant. In addition, you've decided it would be better if it could print *any number* of squares. Therefore, you'd like to rewrite this function to take a parameter `n`, which indicates the number of squares you wish to print, and then it should print those `n` squares in the same format it now prints the first eight squares.

Even More Tracing Code

9. What does the following code print?

```
a_list = [54, 2, 6, 14, 15]
b_list = []

last = None
for x in a_list:
    if last is not None and x > last:
        print(x)
    else:
        b_list.append(x)
    last = x

print(b_list)
```

10. What does the following code print?

```
def scramble(word, x, b):
    result = ''
    n = x
    index = 0
    while n > 0:
        result += word[index]
        n -= 1
        index = (index + b) % len(word)
    return result

print(scramble("floyd", 5, 3))
```

Style Issues

11. Each of the following functions is correct, but contains a significant style flaw. Rewrite the functions to remove the flaw.

```
def time_outdoors(wearing_hat, uv_index):
    if uv_index > 6:
        time = 1
    else:
        time = 4

    if wearing_hat == True:
        time += 1

    return time
```

```
def is_tall(age, height):
    if age <= 10 and height > 4.0 or height >= 6.0:
        return True
    else:
        return False
```