

CSCI 150

Exam 1 Solutions

1. What does this print? Read it carefully.

```
def change( s ):
    s = "Bob"
    return s
```

```
def main():
    s = "Carmen"
    change( s )
    print( s )
```

```
main()
```

This prints "Carmen". Calling change(s) doesn't alter variable s. To do that it would need to say s=change(s)

2. What will this print? Read it carefully.

```
def main():
    number = 27
    done = False
    d = 2
    while not done:
        if number % d == 0:
            done = True
        print(d)
        d = d+1

main()
```

This prints 2
3

The while loop doesn't stop until you get to the top of the loop. So when d is 3 the loop sets done to True, prints 3, sets d to 4, then goes back to the top and stops.

3. What will the following program print?

```
def A(n):  
    return n//2  
  
def B(n): # B(n) prints a line of n stars  
    for j in range(0, n):  
        print( "*", end = "")  
    print()  
  
def C(n): # C(n) calls B(1//2) B(2//2) B(3//2)...B(n//2)  
    for j in range(1, n+1):  
        k = A(j)  
        B(k)  
  
def main():  
    C(8) # C(8) calls B(0) B(1) B(1) B(2) B(3) B(3) B(4)  
  
main()
```

(blank line)

```
*  
*  
**  
**  
***  
***  
****
```

4. This program is supposed to get a range, print the first prime number in this range, and then halt. Instead it prints all of the primes in this range. How can we change main() so it stops after the first prime? I want you to change the code, not to explain the change in English.

```
def isPrime(x):
    # This returns True if x is prime, False otherwise
    # Don't change this; it works
    for d in range(2, x):
        if x%d == 0:
            return False
    return True

def main():
    lower, upper = eval(input( "Enter lower and upper bounds: "))
    for n in range( lower, upper+1 ):
        if isPrime(n):
            print(n)
            done = True

main()
```

The easy fix is to change the done=True line in main() to break

My first thought with this problem would be to change the for-loop to a while-loop, so main() becomes

```
def main():
    lower, upper = eval(input( "Enter lower and upper bounds: "))
    done = False
    n = lower
    while not done and n < upper+1:
        if isPrime(n):
            print(n)
            done = True
        n = n+1
```

5. Here is a sequence of numbers: 0, 1, 3, 6, 10, 15, 21, 28, ...

Note that the first pair (0 and 1) differ by 1, the second pair (1 and 3) differ by 2, the next pair (3 and 6) differ by 3, and so forth. Write a program that asks the user for a number n and then prints the first n elements of this sequence. You can print them horizontally or vertically; I don't care about the format as long as you print the correct numbers.

```
def main():  
    n = eval(input( "How many? "))  
    number = 0  
    for diff in range(0, n):  
        number = number+diff  
        print(number)
```

```
main()
```

Notice that there are n numbers in $\text{range}(0,n)$ so the loop prints n things.

An alternative way to write the loop is

```
for diff in range(1, n+1):  
    print( number )  
    number = number+diff
```

We need to reverse the order of the statements inside the loop to ensure that 0 is printed.

6. Write a program that lets the user repeatedly enter strings; the input stops when the user enters the empty string. The program should count and at the end print the total number of “*” characters from all of these lines of text, BUT it should ignore any * to the right of a comment (or hashtag) symbol #. For example:

```
string: Carmen **Ambar**  
string: This is ***fun!***  
string: bob # *****  
string: *** # *****  
string: *s are # stars  
string:  
I found a total of 14 *s
```

There are many ways to do this. I think the most straightforward is to use a function to count the number of *s on a line. This makes it easy to handle the # character, because when you see it you know the count for that line:

```
def main():  
    count = 0  
    done = False  
    while not done:  
        line = input( ">>> ")  
        if line == "":  
            done = True  
        else:  
            count = count + starCount( line )  
    print( 'I found %d *s'% count )  
  
def starCount( s ):  
    # Counts the number of *s in string s.  
    # this stops counting when it sees #.  
    count = 0  
    for letter in s:  
        if letter == "*":  
            count = count+1  
        elif letter == "#":  
            return count  
    return count  
  
main()
```