

## 22C:16 Practice Problem Set 2

Morning Section: Complete before Tuesday, 2-5-2013

Evening Section: Complete before Monday, 2-4-2013

---

These practice problems correspond roughly to the material covered in the second week of classes (1-28 to 2-1) and to the assigned readings.

1. Here is a program we discussed on converting a nonnegative integer to its equivalent binary representation. This is posted as `intToBinary2.py` on the course page.

```
n = int(raw_input("Type a nonnegative integer. "))

suffix = ""
while n > 0:
    suffix = str(n % 2) + suffix
    n = n/2

print suffix
```

- (a) What output do you get when you change Line 4 to

```
suffix = suffix + str(n % 2)
```

and run the program with input 35? Is the output the binary equivalent of 35? Explain in two sentences what this change is causing the program to do.

- (b) Change Line 4 to

```
suffix = n % 2 + suffix
```

and execute the program. What happens? Explain in one sentence why your program behaves in this manner.

2. Write a Python program that reads as input a positive integer and produces as output the *reverse* of the input integer. For example,

Input	Output
73	37
1024	4201
100	001

Your program should be very similar to `intToBinary2.py`. The basic idea of this program is to extract the decimal digits of the input, one digit at a time, and then put them together in reverse order.

3. For each of these programs, write down what the output will be.

```
(a) n = 10
    if n%3 != 0:
        print "Line 1", n
        print "Line 2", n
    else:
        print "Line 3", n
```

```
(b) n = 11
    if n < 2:
        n = n + 1
        print "Line 1", n
    else:
        print "Line 2", n
    print "Line 3", n
```

```
(c) n = 11
    if n < 11:
        n = n + 1
        print "Line 1", n
    else:
        n = n + 10
        print "Line 2", n
    print "Line 3", n
```

4. Write a program that prompts the user for a sequence of positive integers and then outputs the number of even integers and the number of odd integers input by the user. The user will input 0 to indicate that she is finished inputting her sequence of positive integers. The 0 is not considered part of the sequence that your program needs to process. Here is an example interaction between the user and the program.

```
Enter a number: 7
Enter a number: 9
Enter a number: 90
Enter a number: 3
Enter a number: 0
Even numbers: 1, Odd numbers: 3
```

5. Write down the output produced by the following program:

```
n = 10
while n < 13:
    if n%3 == 0:
        print "Line 1"
    else:
        print "Line 2"
    n = n + 1
```

6. For each expression below, specify its type and value. For expression (6), suppose that the user types `10 + 20` in response to the prompt. For expression (16), suppose that the user types `20` in response to the prompt.

	Expression	Type	Value
1	<code>17/2</code>		
2	<code>17/2.0</code>		
3	<code>float(17/2)</code>		
4	<code>str(17/2.0)</code>		
5	<code>"hello" + "100"</code>		
6	<code>raw_input("number?")</code>		
7	<code>"100" == 100</code>		
8	<code>float(1) + 1</code>		
9	<code>int("337")</code>		
10	<code>171/5</code>		
11	<code>171%5 != 0</code>		
12	<code>170/(2*5.0)</code>		
13	<code>float(17/2)</code>		
14	<code>17/2 &lt; 17.0/2</code>		
15	<code>"17/2" != "17 /2"</code>		
16	<code>"0 + " + raw_input("number?")</code>		
17	<code>float(1) + 1</code>		
18	<code>float("300." + "100")</code>		
19	<code>float("hello" != "hi")</code>		
20	<code>float(str(5%2) + str(5/2))</code>		

7. Write down the output produced by the following program:

```
n = 10
m = 15
while m >= n:
    if (m + n) % 5 == 0:
        print "Line 1", n, m
        m = m - 1
    else:
        print "Line 2", n, m
        n = n + 1
        m = m - 1
```

---