4.6 Exercises

Write a complete Python program to solve each of the following problems.

4.1. Write a program with a function **next**(n) that returns the number after n (i.e, it returns n+1). Give your program the following main() function:

```
def main():
        print(next(35))
        print(next( next(23) ))
```

Your program should print the value 36, then 25.

4.2. A *perfect* number is one whose factors (including 1 but not the number) sum to the number itself. For example, the factors of 6 are 1, 2, and 3, which sum to 6. The factors of 28 are 1, 2, 4, 7, which sum to 28. The factors of 24, however, are 1, 2, 3, 4, 6, 8, and 12; these sum to 36, so 24 is not perfect. Write a function isPerfect (x) that returns True if x is a perfect number, then incorporate this in a program that finds all of the perfect numbers less than 10,000.

4.3. Write a function printTime(minutes) that inputs a number of minutes and prints this in terms of hours and minutes. For example, printTime(325) should print `"6 hours and 25 minutes."`

4.4. Write a program with area functions for squares, circles and rectangles. You might call these functions AreaSquare(side), AreaCircle(radius) and AreaRectangle(length, width). If you call these functions within the main() function:

```
def main():
       print( AreaSquare(4) )
       print( AreaCircle(10) )
       print( AreaRectangle(6, 7) )
```

the program should print 16, then 314.16, then 42

4.5. Write a function decade(year) that takes as argument a year and returns the start of the decade containing this year. For example, for argument 1968 the function should return 1960; for 1999 it should return 1900, and so forth. There are several ways to do this: you can either use the % operator (1968 % 10 is 8, so 1968 − (1968%10) is 1960) or you can have a loop that counts down by 10's from some upper limit (say 3000), and stops when it gets a decade smaller than or equal to the year. Alternatively, you could count down from the year you start, looking for a year that is divisible by 10.

Once you have function decade(year) written you can add the following main() function:

```
def main():
    y = 1
    while y != 0:
        y = eval(input( "Enter a year, 0 to exit: " ))
        if y != 0:
            print("%d was part of the %d's" %(y,decade(y)))
```

On an input like 1968 the program will print `"1968 was part of the 1960's."` Of course, for the 1960's it should add `"Now that was a groovy decade."`

4.6. Write a function deleteB(string) that removes all the instances of the letter `"b"` from a string and returns the result. Here are two hints. First, you can use a **for**-loop to run through the letters of a string:

```
for letter in "blah":
    print( letter )
```

will print first `"b"`, then `"l"`, then `"a"`, then `"h"`. Second, you can use concatenation to build up strings. Start variable result as the empty string:

```
result = ""
```

Then use the +-operator to add letters onto result. For example, you can copy a string with

```
result = ""
for letter in "blah":
    result = result + letter
```

Finally, give your program a main() function to test this out:

```
def main():
    print( deleteB("bob is a blob") )
```

will print `"o is a lo"`, a far nicer string.