# 6.189 Sample Exam

## Introduction

The purpose of the test is to give you an idea of the kind of comfort with programming that we'd you to get to very quickly in 6.01. If you do fine on this test, you should have no difficulty with the programming aspects of 6.01 on start-up.

If you want additional practice, please visit the 6.01 webpage at `http://web.mit.edu/6.01`. There, you can sign up for a *free* Python tutor account, where you'll get more practice working with Python functions and types. Additionally, the 6.01 course staff will be hosting open office hours during the last week of IAP and throughout the day on Reg Day (Feb. 1st) to assist any students who wish for some more structured help.

The test consists of short programming problems that you should do in any language you find convenient. Whichever language you pick, please actually write the programs - and *run and debug* them. Please also work on these problems alone. The purpose of the diagnostic is to gauge your ability, not the ability of a team.

## Programming Problems

1. (a) Write a procedure that solves quadratic equations using the quadratic formula: It should take as arguments three numbers a, b, and c. It should print error messages if a is zero, or if the roots are complex. Otherwise it should print the two roots. (b) Modify your procedure to handle the case of complex roots.

2. Write a procedure that evaluates polynomials. It should take two arguments. The first is a number $x$. The second is a list of of coefficients ordered from highest to lowest:

$$a_n, a_{n-1}, \ldots, a_2, a_1, a_0$$

   Your procedure should return the value of the polynomial evaluated at $x$:

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

3. A clerk works in a store where the cost of each item is a positive integer number of dollars. So, for example, something might cost \$21, but nothing costs \$9.99. In order to make change a clerk has an unbounded number of bills in each of the following denominations: \$1, \$2, \$5, \$10, and \$20. Write a procedure that takes two arguments, the cost of an item and the amount paid, and prints how to make change using the smallest possible number of bills.

4. (a) Write a procedure that takes a string of words separated by spaces (assume no punctuation or capitalization), together with a "target" word, and shows the position of the target word in the string of words. For example, if the string is:

```
we dont need no education we dont need no thought control no we dont
```

and the target is the word "dont" then your procedure should return the list 1, 6, 13 because "dont" appears at the 1st, 6th, and 13th position in the string. (We start counting positions of words in the string from 0.) Your procedure should return False if the target word doesn't appear in the string.

(b) Suppose you are repeatedly looking up targets in a fixed long list. It might help to build an "inverted index", that shows the positions of all targets, so that this information can be retrieved quickly. For example, an inverted index for the above string of words would be:

```
we: 0, 5, 12
dont: 1, 6, 13
need: 2, 7
no: 3, 8, 11
education: 4
thought: 9
control: 10
```

Use an appropriate data structure to represent an inverted index. Write a procedure that, given a string of words, constructs an inverted index, and show how to use the index to look up target words as in part (a).

6.189 A Gentle Introduction to Programming Using Python
January (IAP) 2010