

# APS 105S – Computer Fundamentals

Assignment #5: Recursion

Winter 1999

(To be completed for your lab period February 8/9.)

**Objective:** To understand how recursion works.

## 1 Printing A List of Numbers in Reverse Order

Design, implement and test a Java program that reads a list of numbers and prints them in reverse order. The number 0 will end the list, and is not to be included in the output. Use recursion to reverse the list. Do not use an array or any other data structure that has not yet been discussed in class.

*Hint:* Design a method "reverseNumbers" that reads an integer and returns if it's 0; otherwise the method calls itself to process the other integers in the list and then prints the integer.

Your code will look something like this:

```
public class Reverse {
    static int promptInt (String prompt) {
        System.out.print (prompt);
        return StdIn.getInt ();
    }
    static void reverseNumbers () {
        int i = promptInt ("Enter an integer: ");
        // PUT YOUR CODE HERE
    }
    public static void main (String[] args) {
        System.out.println ("Enter a list of integers, ending with 0.");
        reverseNumbers ();
    }
}
```

1

Execution of the program should be similar to the example that follows. The input supplied by the user is shown in **bold**.

Enter a list of integers, ending with 0.

Enter an integer: **8**

Enter an integer: **-2**

Enter an integer: **3**

Enter an integer: **0**

3

-2

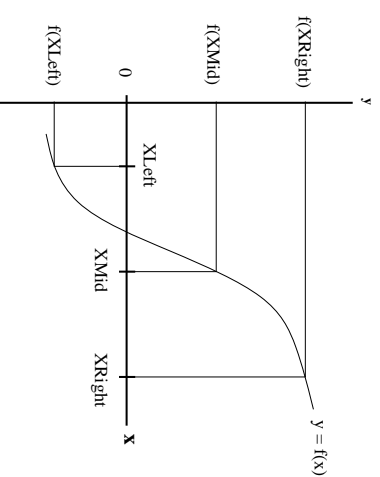
8

## 2 Finding Roots of Any Function

In assignment 2 you used an equation to determine the roots of a quadratic function. In general, it is more difficult to find roots of higher-order polynomials (those with high powers of x). In this section you will use a general method to directly search for the roots of a function, and you will use recursion to implement it. The method is called the "bisection method." The bisection method is a means of finding an approximate root for the equation  $f(x) = 0$  on the interval **xLeft** to **xRight** inclusive. (We will assume that the function is continuous in this interval.) The user inputs are:

- the interval endpoints: **xLeft** and **xRight**
- the tolerance for the approximation: **epsilon**

The figure below shows an example situation:



2

The goal is to find an interval  $[x_{\text{left}}, x_{\text{right}}]$  that is less than **epsilon** in length over which the function changes sign (and hence must be equal to zero, the root, somewhere between the two points). The midpoint of the interval

$$x_{\text{Mid}} = \frac{x_{\text{left}} + x_{\text{right}}}{2.0}$$

will then be an approximation of the root of the equation. If you happen to find a value of  $f(x_{\text{Mid}})$  that does equal zero then you are done and the program should stop.

If the sign change happens in the interval  $[x_{\text{left}}, x_{\text{Mid}}]$  then let **xRight** = **xMid**, and similarly if the change is in the other interval. Then repeat the process.

### Part A

Find an approximation to the equation

$$5x^3 - 2x^2 + 3 = 0$$

over the interval  $(-1, 1)$  using **epsilon** = 0.0001. Your program should use a *recursive* implementation of the bisection method.

### Part B

Modify your solution for Part A to find approximations to the three roots of the equation

$$2x^3 - x^2 - 13x - 6 = 0$$

You may assume that all roots are real numbers.