Lab #7 (15 points)

1. Write a program that reads in a set of integers (one at a time) and prints out the entire set backward, then forward. You may use as many Stacks as you wish, but you may not use any other objects (including Strings), just simple variables. If you find it necessary to explicitly box or unbox your numbers, you are permitted to use an Integer object.

2. Write a program that uses a stack or stacks to implement a prefix calculator for integers. Assume that the user inputs a well-formed prefix arithmetic expression.

Your program should prompt for an expression, evaluate it and show the result. Continue to do this until the user signals a desire to quit.

Sample run:

Enter a well-formed prefix expression:
- + 16 4 11
The result is 9
Do you want to continue? (y/n): y

Enter a well-formed prefix expression:
/ * 2 3 4 5
The result is 4
Do you want to continue? (y/n): y

Enter a well-formed prefix expression:
% + * 3 3 3 – 11 4
The result is 5
Do you want to continue? (y/n): n
OK, thanks for playing. Goodbye.

Extra credit options (worth 5 points each):

- For part 1: Divide each number read into its individual digits, and print each number both forward and backward as well
- For part 2: Add syntax checking to your calculator, and refuse to accept expressions that are not well-formed. Your program should recover from such an error, not just throw an exception and halt.