PP 13.6 Modify the solution to the PP 13.5 so that it represents eight
tellers and therefore eight customer queues. Have new custom-
ers go to the shortest queue. Determine which queue had the
shortest waiting time per customer on average.

PP 13.7 Design and implement an application that evaluates a postfix
expression that operates on integer operands using the arith-
metic operators +, −, *, /, and %. We are already familiar with
infix expressions, in which an operator is positioned between
its two operands. A postfix expression puts the operators after
its operands. Keep in mind that an operand could be the result
of another operation. This eliminates the need for parentheses
to force precedence. For example, the following infix expres-
sion:

\[(5 + 2) \times (8 - 5)\]

is equivalent to the following postfix expression:

\[5 \ 2 \ + \ 8 \ 5 \ - \ *\]

The evaluation of a postfix expression is facilitated by using a
stack. As you process a postfix expression from left to right,
you encounter operands and operators. If you encounter an
operand, push it on the stack. If you encounter an operator,
pop two operands off the stack, perform the operation, and
push the result back on the stack. When you have processed
the entire expression, there will be one value on the stack,
which is the result of the entire expression.

You may want to use a StringTokenizer object to assist in the
parsing of the expression. You can assume the expression will
be in valid postfix form.

PP 13.8 Design and implement a program that prompts the user
to enter a string and then performs two palindrome tests.
The first should use a single stack to test whether the string
is a palindrome. The second should use two stacks to test
whether the string is a palindrome when capitalization, spaces,
punctuation, and other non-alphanumeric characters are
ignored. The program should print the results of both tests.

PP 13.9 Design and implement a class named StringTree, a binary
tree for storing String objects in alphabetic order. Each node
in the tree should be represented by a Node class, which stores
the string value and pointers to the right and left child nodes.
For any node value in the tree, the value of its left child should