1- Converting an expression from infix to postfix

```
opstk= the empty stack;
while (not end of input)
{
   symb=next input character;
   if (symb is an operand)
      add symb to the postfix string
   else
   {
      while (!empty()&&prcd(stacktop(), symb))
      {
            topsym=pop();
            add topsymbto the postfix string;
       } /*end while*/
      if (empty() II symb != ')')
           push(symb);
       else /* pop the open parenthesis and discard it */
            topsymb= pop();
   } /* end else*/
} /* end while*/
/*output any remaining operators*/
while (!empty())
 {
      topsymb=pop();
      add topsymb to the postfix string;
 } /* end while*/
```

2- Evaluating a postfix expression

```
Opndstk= the empty stack;
/* scan the input string reading one element at a time into
symb*/
while (not end of input)
 {
        symb=next input character;
        if (symb is an operand)
                push(symb);
        else
        {
                /* symb is an operator*/
                opnd2=pop();
                opnd1=pop();
                value=resulr of applying symb to opnd1 and
                opnd2;
                push(value);
         } /* end else*/
    } /* end while*/
return (pop());
```