Infix to Postfix algorithm

```
InfixtoPostfix(exp)
{create a Strack S
 for i <-0 to length(exp)-1
 {
    if exp[i] is operand
       res <-res + exp[i]
    else if exp [i] is operator
       while (!S.empty() && HasHigerPre(S.top(), exp[i])
           { res <-res +S. top()
             S.pop()

           }
          S.push (exp (i])
     else if IsOpeningParentheses(exp (i])
           S.push (exp [i])
     else if ISClosingParentheses(exp (i])
             S.push(exp [i]
     elseif IsClosingParentheses(exp [i])
        { while (!S.empty() && ! ISOpeningParentheses(S.
                                                  top()))
              { res <- res + S. top()
                 S.pop()
              }
              S.pop()
        }
  }
While (! S.empty())
{
  res <- res + S.top()
          S.pop()
  }
  return res
}
```