```
Infixpostfix(exp)
{
  create a stack S
  for i<=0 to length(exp)-1
  {
    if exp[i] is operand
        res ← res + exp[i]
    elseif exp[i] is operator
        while(is.empty() && HasHigher Pre(s.top(), exp[i]))
        {
            res ← res+s.top()
            s.push(exp[i])
    elseif Is Opening Parenthises(exp[i])
            s.push(exp[i])
    elseif Isclosing parentheses (exp[i])
    {
        while(is.empty() && isopeningParanthesis(s.top()))
            res ← res + s.top()
            s.pop()
        }
            s.pop()
    }
  }
    while(is.empty())
    {
        res ← res + s.top()
        s.pop()
    }
  return res
}.
```