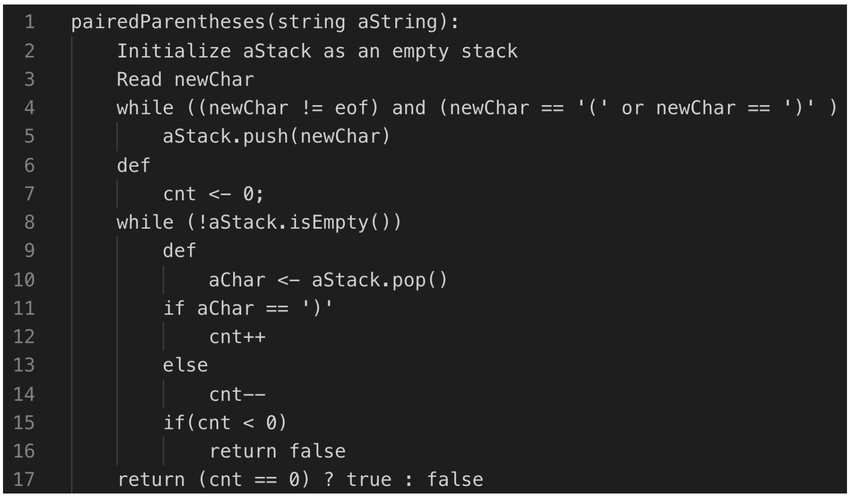
EE4, B04901158
Jing-Ping Cheng

Data Structure and Advanced Programming
Homework 4

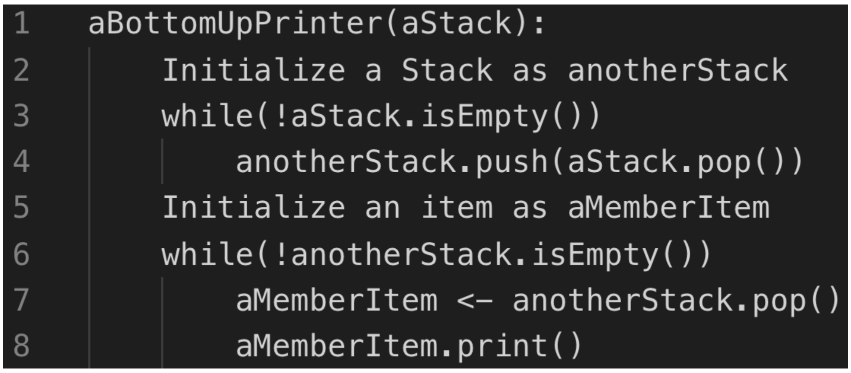
第一題

(a) Paired parentheses



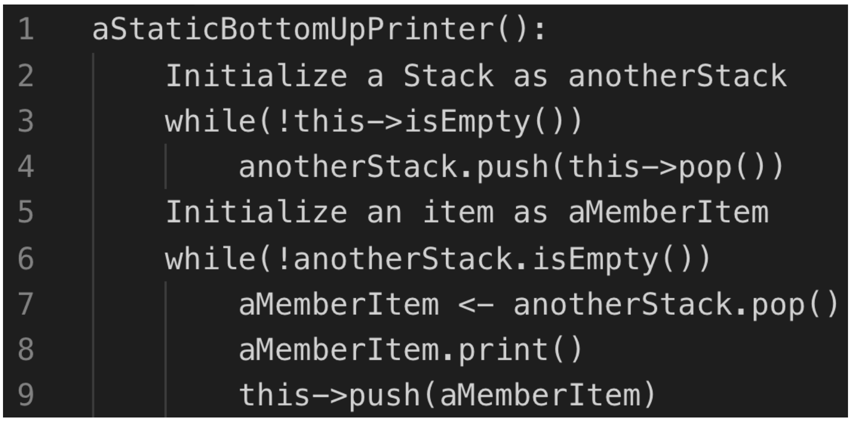
```
pairedParentheses(string aString):
    Initialize aStack as an empty stack
    Read newChar
    while ((newChar != eof) and (newChar == '(' or newChar == ')' )
        aStack.push(newChar)
    def
        cnt <- 0;
    while (!aStack.isEmpty())
        def
            aChar <- aStack.pop()
        if aChar == ')'
            cnt++
        else
            cnt--
        if(cnt < 0)
            return false
    return (cnt == 0) ? true : false
```

(b) A bottom-up printer



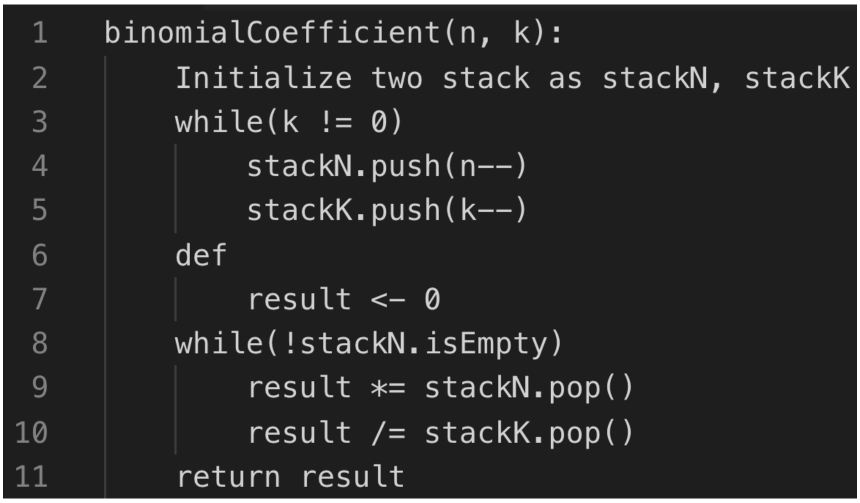
```
aBottomUpPrinter(aStack):
    Initialize a Stack as anotherStack
    while(!aStack.isEmpty())
        anotherStack.push(aStack.pop())
    Initialize an item as aMemberItem
    while(!anotherStack.isEmpty())
        aMemberItem <- anotherStack.pop()
        aMemberItem.print()
```

(c) A bottom-up printer as a member function



```
aStaticBottomUpPrinter():
    Initialize a Stack as anotherStack
    while(!this->isEmpty())
        anotherStack.push(this->pop())
    Initialize an item as aMemberItem
    while(!anotherStack.isEmpty())
        aMemberItem <- anotherStack.pop()
        aMemberItem.print()
        this->push(aMemberItem)
```

(d) Binomial Coefficient



```
binomialCoefficient(n, k):
    Initialize two stack as stackN, stackK
    while(k != 0)
        stackN.push(n--)
        stackK.push(k--)
    def
        result <- 0
    while(!stackN.isEmpty)
        result *= stackN.pop()
        result /= stackK.pop()
    return result
```