(a)

```
bool isCorrect(string str)
{
        for i in str
                if i == '('
                        right++
                else if i == ')'
                        left++
                if left > right
                        return false
        return (right == left)
}
```

(b)

```
void print(Stack s)
{
        while s is not empty
                push s's top in stacktoPrint
                pop s
        while stacktoPrint is not empty
                print stacktoPrint's
                pop stacktoPrint
}
```

(c)

```
//assume linked stack
void Stack<Item>::print() const
{
        nodePtr = topPtr
        while nodePtr is not nullptr
                push nodePtr->getItem in stacktoPrint
                nodePtr getNext

        nodePtr = stacktoPrint.topPtr
        print nodePtr->getItem
        while(nodePtr's Next is not nullptr)
                nodePtr getNext
                print nodePtr->getItem
}
```

(d)

```
int CnGetk(int n, int k)
{
    Stack<int> nS;
    Stack<int> kS;
    int ans = 0;
    nS.push(n);
    kS.push(k);

    //assume n >= k
    while(!nS.isEmpty() && !kS.isEmpty())
    {
        int a = nS.peek();
        nS.pop();
        int b = kS.peek();
        kS.pop();
        if(b == 1)
        {
            ans += a;
        }
        else if(a == b)
        {
            ans += 1;
        }
        else
        {
            nS.push(a - 1);
            kS.push(b - 1);

            nS.push(a - 1);
            kS.push(b);
        }
    }
    return ans;
}
```