

Section Solution

Solution 1: Look And Say

```
string LookAndSay(string before)
{
    string after;
    int len = before.size();
    int pos = 0;
    while (pos < len) {
        char ch = before[pos];
        int count = 1;
        while ((pos + count < len) &&
            (before[pos + count] == ch)) {
            count++;
        }
        after += count + '0';
        after += ch;
        pos += count;
    }
    return after;
}
```

Solution 2: <http://tinyurl.com>

```
static const int kTinyURLBase = 36;
string TinyURLConvertNumber(int num)
{
    string code; // default constructed to the empty string
    while (num > 0) {
        int digit = num % kTinyURLBase;
        code = ConvertDigit(digit) + code;
        num /= kTinyURLBase;
    }

    return code;
}

char ConvertDigit(int digit)
{
    if (digit < 10) return '0' + digit;
    return 'a' + digit - 10;
}

TinyURLRecoverNumber(string code)
{
    int num = 0;
    for (int i = 0; i < code.size(); i++) {
        num = kTinyURLBase * num + ConvertChar(code[i]);
    }

    return num;
}
```

```
int ConvertChar(char ch)
{
    if (isdigit(ch)) return ch - '0';
    return ch - 'a' + 10;
}
```

Solution 3: Maximizing Game Score

```
int ComputeMaxScore(Grid<int>& board)
{
    int prev = max(board[0][0], board[1][0]);
    if (board.numCols() == 1) return prev;
    int curr = max(prev, max(board[0][1], board[1][1]));
    if (board.numCols() == 2) return curr;
    for (int col = 2; col < board.numCols(); col++) {
        int next = max(curr, prev + max(board[0][col], board[1][col]));
        prev = curr;
        curr = next;
    }
    return curr;
}
```