**第六周作业**

**姓名：陈岳阳**

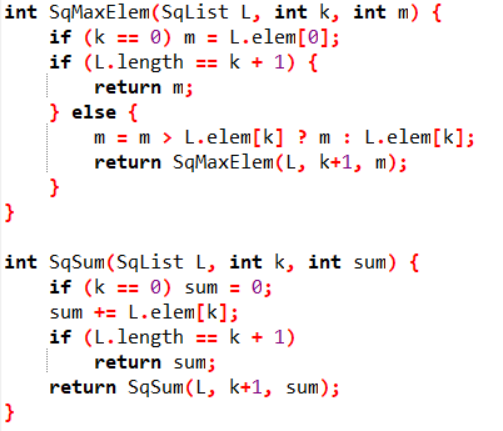
**学号：21020007009**

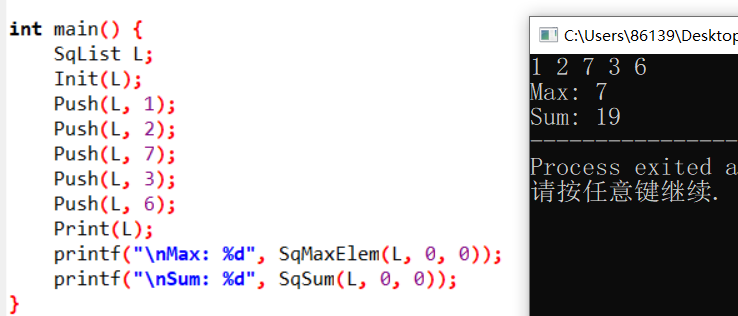
**一、已知顺序表L含有n个整数，试分别以函数形式写出下列运算的递归算法。（1）求表中的最大整数（3）求表中n个整数之和。**

**答：**

设计思路：递归需要输入一个参数k，代表现在访问的元素下标。可以考虑加入一个结果m/sum，让递归变为尾递归，减少空间复杂度。也可以不将m/sum作为参数，但是这样空间复杂度会上升。

算法分析：两个算法均为线性时间复杂度。由于是尾递归，且没有额外空间开销，空间复杂度为常量。

****



**二、若矩阵Am×n中的某个元素aij是第i行中的最小值，同时又是第j列的最大值，则称此元素为矩阵中的一个马鞍点。假设以二维数组存储矩阵Am×n，试编写求出矩阵中所有马鞍点的算法，并分析你的算法在最坏情况下的时间复杂度。**

**答：**

设计思路：先求出每行最小值，以及每列最大值，分别用数组存储，然后将矩阵中的每个元素进行匹配，找到所有SaddlePoint。

算法分析：由于算法对整个矩阵进行遍历，时间复杂度为O(mn)。最坏的情况下，矩阵中所有点都是马鞍点，空间复杂度为O(mn)。

伪代码如下：

|  |
| --- |
| typedef struct {  int x, y;  }Point;  Point\* GetSaddlePoint(Matrix mat, int m, int n) {  Point ans[m\*n]; //马鞍点构成的数组  int row\_min[m], col\_max[n]; //存储每行最小值以及每列最大值  int temp;  //每行最小值  for (int i = 0; i < m; ++i) {  temp = mat[i][0];  for (int j = 1; j < n; ++j)  temp = min(temp, mat[i][j]);  row\_min[i] = temp;  }  //每列最大值  for (int j = 0; j < n; ++j) {  temp = mat[0][j];  for (int i = 1; i < m; ++i)  temp = max(temp, mat[i][j]);  col\_max[j] = temp;  }    //逐个检查  int k = 0;  for (int i = 0; i < m; ++i) {  for (int j = 0; j < n; ++j) {  if (mat[i][j] == row\_min[i] && mat[i][j] == col\_max[j]) {  ans[k].x = i;  ans[k].y = j;  k++;  }  }  }    return ans;  } |

**注：**

**1、红字部分是需要修改的内容；**

**2、题目描述部分使用：宋体 小四 加粗**

**解答内容部分使用：宋体 小四；**

**3、英文内容字体使用Times New Roman；**

**4、全文使用单倍行距；**

**5、尽量保证作业整洁美观。**