**求n×n阶螺旋方阵（要求用递归算法完成）**

#include<iostream>

#include<vector>

using namespace std;

// 递归函数，填充螺旋矩阵 round是表是第几轮

**bool function(vector<vector<int>>& general, int n, int rounds, int& num) {**

// 判断是否需要填充

if (n - rounds \* 2 <= 0) { //1\*1 2\*2

return true;

}

else if (n - rounds \* 2 == 1) { //3\*3的方阵

general[rounds][rounds] = num; // 填充中心

return true;

}

// 从左到右填充上边

for (int i = rounds; i < n - 1 - rounds; i++) {

general[rounds][i] = num++;

}

// 从上到下填充右边

for (int i = rounds; i < n - 1 - rounds; i++) {

general[i][n - 1 - rounds] = num++;

}

// 从右到左填充下边

for (int i = n - 1 - rounds; i > rounds; i--) {

general[n - 1 - rounds][i] = num++;

}

// 从下到上填充左边

for (int i = n - 1 - rounds; i > rounds; i--) {

general[i][rounds] = num++;

}

// 递归调用，填充下一层

**return function(general, n, rounds + 1, num);**

**}**

int main() {

int n;

cin >> n;

if (n <= 0) {

return 0;

}

vector<vector<int>> general(n, vector<int>(n, 0)); // 初始化矩阵为 0

int num = 1; // 从 1 开始填充

function(general, n, 0, num); // 开始填充

// 输出矩阵

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

cout << general[i][j] << ' ';

}

cout << endl;

}

return 0;

}

1

1 2

4 3

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 8 | **9** | 4 |
| 7 | 6 | 5 |

**单循环链表递增输出及清空**

**为单循环链表类添加方法，按递增序输出表中所有元素，并释放所有元素结点所占存储空间，不允许使用数组作为辅助空间。**

【输入形式】

第1行，1个整数n，表示将创建的单循环链表中元素个数

第2行，n个整数，表示将创建的单循环链表中的从前往后的所有元素，如果第1行输入的n小于等于0，则无需输入。

【输出形式】n个整数，单循环链表中按递增序排列的所有元素

例子：5 3 4 1 2

1 2 3 4 5

算法思路：

循环执行，直到first.next==first:

1. 从循环链表中找出最小的，打印输出该最小结点后，并删除该结点，
2. 继续定位到头结点，转到（1）

参考：

void CirLinkList<DataType> ::PrintOrdered\_clear()

{

Node<DataType>\* p = first, \* q, \* minp = nullptr, \* temp = nullptr;

DataType minValue;

while (first->next != first) {

p = first;

q = first->next;

minValue = q->data;

minp = p;

while (q != first) {

if (q->data < minValue) {

minValue = q->data;

minp = p;

}

p = q;

q = q->next;

}

temp = minp->next;

cout << temp->data << " ";

minp->next = temp->next;

delete temp;

}

}

**广义表运算的结果**

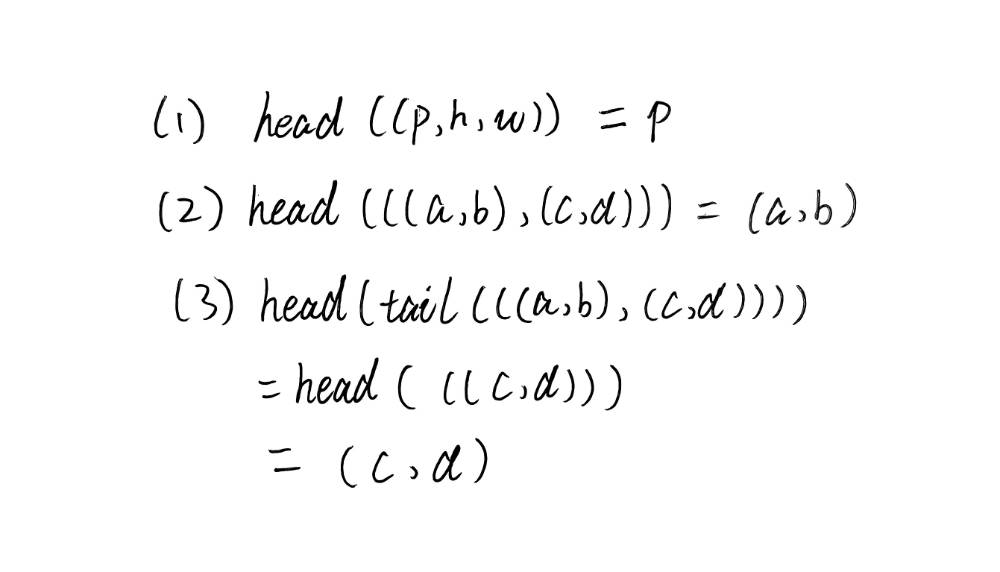
求下列广义表运算的结果

（1）head((p,h,w)) //注意：处理的广义表 (p,h,w) 和（(p,h,w)）不同

（2）head(((a,b),(c,d))) //((a,b),(c,d)) tail（((a,b),(c,d))）=（(c,d)）

（3）head (tail(((a,b),(c,d))))

**X=((a,b),(c,d)) Y=tail(X)= ((c,d)) head(Y)=(c,d)**

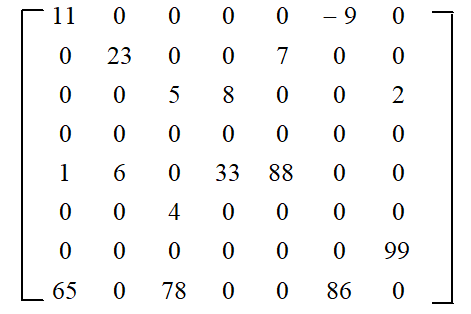


**转置矩阵的三元组表**

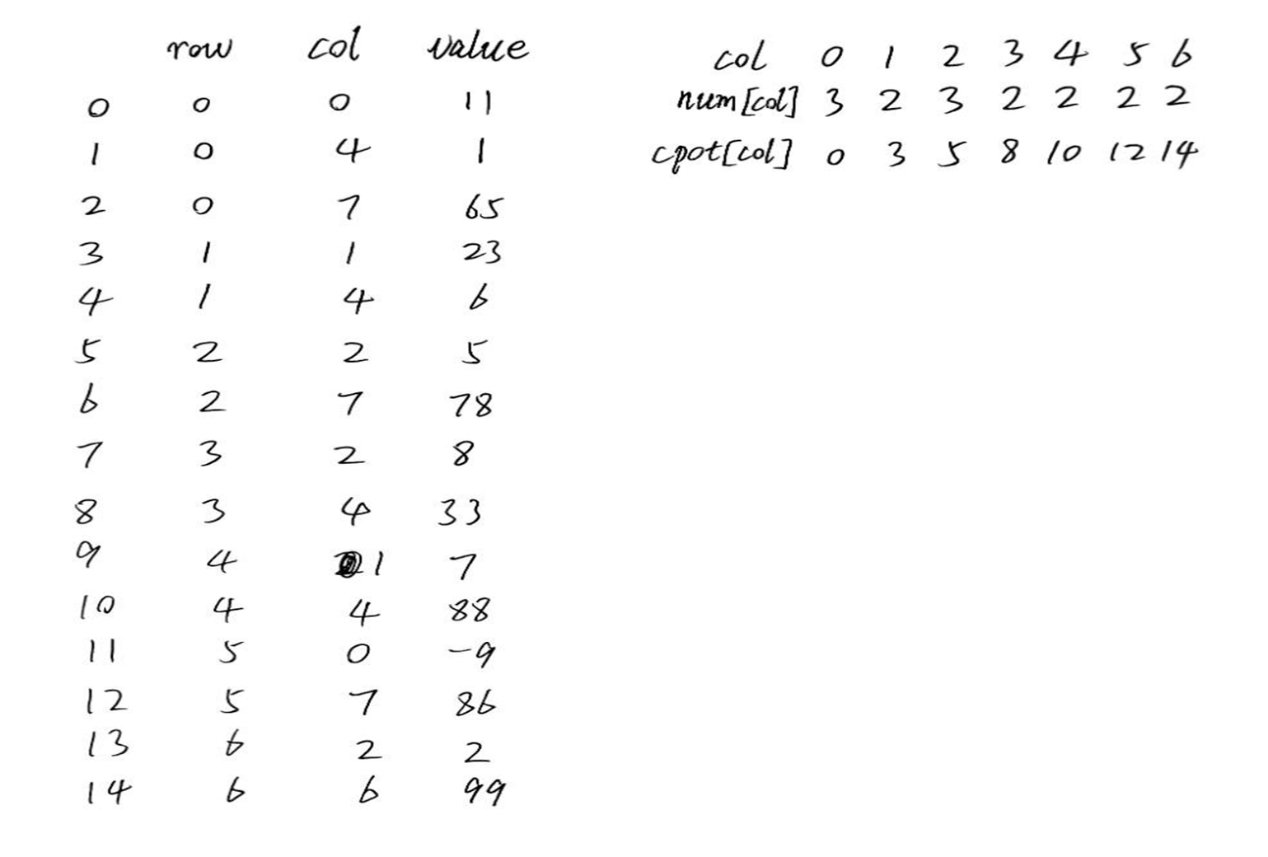
给定一个稀疏矩阵如下：

（1）画出它的转置矩阵的三元组表；

（2）给出num[col]和cpot[col]的内容。



|  |  |  |  |
| --- | --- | --- | --- |
|  | Row | Col | Value |
| 0 | 0 | 0 | 11 |
| 1 | 0 | 5 | -9 |
|  | 1 | 1 | 23 |
|  | 1 | 4 | 7 |
|  | 2 | 2 | 5 |
|  | 2 | 3 | 8 |
|  | 2 | 6 | 2 |
|  | 4 | 0 | 1 |
|  | 4 | 1 | 6 |
|  | 4 | 3 | 33 |
|  | 4 | 4 | 88 |
|  | 5 | 2 | 4 |
|  | 6 | 6 | 99 |
|  | 7 | 0 | 65 |
|  | 7 | 2 | 78 |
|  | 7 | 5 | 86 |



1.给定具有n个元素的顺序表，建立一个有序线性链表的时间复杂度为   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_     。

A. O（1）      B. O（n）      C. O（n2）    D. O（log2n）

**正确答案:**C

2.在一个长度为n、包含m个原子元素的广义表中 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 。

A. m=n    B. m<=n    C. m>=n    D. m与n无关

**正确答案:**D

3.在一个具有n个链结点的线性链表中查找某一个链结点，若查找成功，需要平均比较 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 个链结点。

A. n        B. n/2        C. (n+1)/2        D. (n-1)/2

**正确答案:**C

4.在一个单向循环链表中，若要在p所指向的结点之后插入一个新结点，则需要相继修改 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 个指针域的值

A. 1

B. 2

C. 3

D. 4

**正确答案:**B

5.在非空双向循环链表中由q所指的那个链结点前面插入一个由p所指的链结点的动作所对应的语句依次为：p一>rlink=q; p一>llink=q一>llink; q一>llink=p; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 。(空白处为一条赋值语句)

A. q一>rlink= p;

B. q一>llink一>rlink=p;

C. p一>rlink一>rlink= p;

D. p一>llink一>rlink=p;

**正确答案:**D (**很多错选了B**)

