一、这是一道已知权重数组输出相应字符的哈夫曼编码的试题，其中用堆来完成排序和取最小权重哈夫曼树的操作。数据类型 void*是一种通用指针，它可以通过强制类型转换与所有类型的指针交换数据。源代码存在附件目录 haffman 中。工程中一共有 5 个函数没有完成，它们是 buildHeap()，push()，pop()，join(), createCode()。请根据以下说明完成所有这 5 个函数：

（1） buildHeap()是用筛选法建立大根堆的函数。R 是用来存储大根堆的数组，有效元素从下标 1 开始存储。函数 sift()完成一次筛选操作；（10 分）

（2） push()是用来向大根堆插入新元素的函数。新插入的元素首先会被放入数组中所有有效元素的最末尾，然后通过向上调整的操作使数组重新恢复成一个最大堆。注意参数 n 会在函数内部被修改；（10 分）

（3） pop()是取走堆顶元素的操作。堆顶元素被取走之后，数组最后一个元素会拷贝至堆顶，然后通过一次筛选操作使数组恢复成一个大根堆。注意参数 n 会在函数内部被修改；（10 分）

（4） join()函数是用来合并两棵小哈夫曼树的操作，根节点权值较小的子树要求被放在左边。在合并两棵小哈夫曼树时，会创建一个新的 HaffmanNode 节点，原来的两棵树分别成为这个新的根节点的左右子树。在两个传引用的参数中，t1 存放合并后的二叉树，t2 被释放；（10 分）

（5） createCode()是一个递归函数，它利用已经建好的哈夫曼树创建哈夫曼编码。函数的大部分代码已经给出，请完成有关递归的部分。注意：沿着某个节点向左遍历时，输出代码‘0’，向右遍历时，输出代码‘1’。（10 分）

二、这是一道用图的广度优先遍历求迷宫问题的最短路径的问题。迷宫本身用一个二维数组表示，0 表示可以通行，1 表示有障碍。求解方案是先将迷宫视为一个图，图的顶点表示迷宫中的一个方格。边表示两个方格之间有通路。第一步将迷宫转换成图的邻接表，然后利用广度优先遍历解决路径搜索的问题。源代码存在附件目录 graphmg 中。工程中一共有 5 个函数没有完成，它们是 buildAdjGraph ()，queueEmpty ()，dequeue ()，destroyQueue (), mgpath ()。请根据以下说明完成所有这 5 个函数：

（1） buildAdjGraph ()是将迷宫二维数组转换成图的邻接表表示的函数。邻接表由顶点列表和边节点链表两大部分组成。与一般的邻接表不同的是，这里的顶点列表是一个二维数组。相关数据类型定义已经完成，请注意在转换过程中空指针的处理；（20 分）

（2） queueEmpty ()是用来判断队列是否为空的函数。注意这里构造的是循环队列，请参考其他几个已经给出的函数；（5 分）

（3） dequeue ()是取出队头元素的操作。请参考入队列的操作来完成此函数；（10 分）

（4） destroyQueue ()函数用来释放循环队列的空间；（5 分）

（5） mgpath()执行图上的广度优先遍历。在这个过程中发现的第一条从入口到出口的路径，就是迷宫问题的最短路径。该函数的部分实现已经给出，请补充完成缺失的部分。（10 分）

三、这是一道在链表结构上进行二路归并排序的问题。输入数据被组织到一个单向链表中。为了方便进行二路归并排序，需要另外建立一个单向链表 SequenceList sl，它的每一个节点代表一个已经排好序的子序列，连续两个子序列会进行归并操作，结果会再次放入 sl 中。工程中一共有 5 个函数没有完成，它们是 buildList ()，initSequenceList ()，merge ()，insertNode (), mergeSort ()。请根据以下说明完成所有这 5 个函数：

（1） buildList ()是将数组中的元素转储到链表中去的操作。链表中的节点是动态创建的，

要求链表当中数据的排列顺序同数组中的顺序相同；（5 分）

（2） initSequenceList ()是用来为二路归并排序做准备的函数。原始的数据链表会被分成 n 个独立的包含单个节点的子序列，每一个序列的首节点地址保存在链表 sl 的节点中；（10 分）

（3） merge()是合并两个有序子序列的操作。链表 sl 中相邻的两个节点指向的子序列会被合并成一个子序列，结果存放在第一个节点中，第二个节点将来在完成其参与的所有功能后被释放；（10 分）

（4） insertNode ()函数在函数 merge()内部被调用，它将某个特定的数据节点插入到合并之后的子序列链表中去。注意各个传引用的参数会在函数内部被修改；（5 分）

（5） mergeSort ()是完成二路归并操作的函数，它是通过不断调用 merge()函数来完成排序的。这个函数带两个参数，二路归并排序是通过链表 sl 完成的，最终的排序结果会被重新归还给链表 l。（20 分）