|  |
| --- |
| 形考任务一 |
| 一、单项选择题 |
| 1.把数据存储到计算机中，并具体体现数据元素间的逻辑结构称为（ ）。 |
| 【答案】物理结构 |
| 2.下列说法中，不正确的是（ ）。 |
| 【答案】数据项可由若干个数据元素构成 |
| 3.一个存储结点存储一个（ ）。 |
| 【答案】数据元素 |
| 4.数据结构中，与所使用的计算机无关的是数据的（ ）。 |
| 【答案】逻辑结构 |
| 5.在线性表的顺序结构中，以下说法正确的是（ ）。 |
| 【答案】逻辑上相邻的元素在物理位置上也相邻 |
| 6.对链表, 以下叙述中正确的是（ ）。 |
| 【答案】不能随机访问任一结点 |
| 7.下列的叙述中，不属于算法特性的是（ ）。 |
| 【答案】可读性 |
| 8.算法的时间复杂度与（ ）有关。 |
| 【答案】算法本身 |
| 9.设有一个长度为n的顺序表，要在第i个元素之前（也就是插入元素作为新表的第i个元素），插入一个元素，则移动元素个数为（ ）。 |
| 【答案】n-i+1 |
| 10.设有一个长度为n的顺序表，要删除第i个元素移动元素的个数为（ ）。 |
| 【答案】n-i |
| 11.在一个单链表中，p、q分别指向表中两个相邻的结点，且q所指结点是p所指结点的直接后继，现要删除q所指结点，可用语句（ ）。 |
| 【答案】p->next=q->next |
| 12.在一个单链表中p所指结点之后插入一个s所指的结点时，可执行（ ）。 |
| 【答案】s->next=p->next; p->next=s; |
| 13.非空的单向循环链表的尾结点满足（ ）（设头指针为head，指针p指向尾结点）。 |
| 【答案】p->next==head |
| 14.链表不具有的特点是（ ）。 |
| 【答案】可随机访问任一元素 |
| 15.带头结点的链表为空的判断条件是（ ）（设头指针为head）。 |
| 【答案】head->next==NULL |
| 16.在一个长度为n的顺序表中为了删除第5个元素，由第6个元素开始从后到前依次移动了15个元素。则原顺序表的长度为（ ）。 |
| 【答案】20 |
| 17.有关线性表的正确说法是（ ）。 |
| 【答案】除了一个和最后一个元素外，其余元素都有一个且仅有一个直接前驱和一个直接后继 |
| 18.向一个有127个元素的顺序表中插入一个新元素，并保持原来的顺序不变，平均要移动（ ）个元素。 |
| 【答案】63.5 |
| 19.一个顺序表第一个元素的存储地址是90，每个元素的长度为2，则第6个元素的地址是（ ）。 |
| 【答案】100 |
| 20. 在一个不带头结点的单循环链表中，p、q分别指向表中第一个结点和尾结点，现要删除第一个结点，且p、q仍然分别指向新表中第一个结点和尾结点。可用的语句是p=p->next;和（ ）。 |
| 【答案】q->next=p |
| 二、判断题 |
| 21.数据元素可以有一个或多个数据项组成。 |
| 【答案】对 |
| 22.数据元素之间的抽象关系称为物理结构。 |
| 【答案】错 |
| 23.数据的逻辑结构在计算机中的表示称为逻辑结构。 |
| 【答案】错 |
| 24.数据的逻辑结构是与存储该结构的计算机相关的。 |
| 【答案】错 |
| 25.数据结构中,元素之间存在多对多的关系称为树状结构。 |
| 【答案】错 |
| 26.通常可以把一本含有不同章节的书的目录结构抽象成线性结构。 |
| 【答案】错 |
| 27.通常可以把某城市中各公交站点间的线路图抽象成树型结构。 |
| 【答案】错 |
| 28. 设有一个不带头结点的单向循环链表，结点的指针域为next，指针p指向尾结点，现要使p指向第一个结点，可用语句p=p->next;。 |
| 【答案】对 |
| 29. 设有一个单向链表，结点的指针域为next，头指针为head，p指向尾结点，为了使该单向链表改为单向循环链表，可用语句p->next=head 。 |
| 【答案】对 |
| 30. 设有一个单向循环链表，结点的指针域为next，头指针为head，指针p指向表中某结点，若逻辑表达式p->next==head;的结果为真，则p所指结点为尾结点。 |
| 【答案】对 |
| 31. 要在一个单向链表中p所指向的结点之后插入一个s所指向的新结点，若链表中结点的指针域为next，可执行 p->next=s; s->next= p->next；的操作。 |
| 【答案】错 |
| 32. 要在一个单向链表中删除p所指向的结点，已知q指向p所指结点的直接前驱结点，若链表中结点的指针域为next，则可执行q->next= p->next； |
| 【答案】对 |
| 33. 要在一个带头结点的单向循环链表中删除头结点，得到一个新的不带头结点的单向循环链表，若结点的指针域为next，头指针为head，尾指针为p，则可执行head=head-> next; p->next=head；。 |
| 【答案】对 |
| 34. 设有一个单向循环链表，头指针为head，链表中结点的指针域为next，p指向尾结点的直接前驱结点，若要删除尾结点，得到一个新的单向循环链表，可执行操作p->next=head；。 |
| 【答案】对 |
| 三、程序填空题（请点击正确选项，然后拖拽至相应的方框上） |
| 35. 设线性表以不带头结点的单向链表存储，链表头指针为head，以下程序的功能是输出链表中各结点中的数据域data,完成程序中空格部分。 |
| #define NULL 0 |
| void main( ) |
| { NODE \*head ,\*p ; |
| p=head; /\*p为工作指针\*/ |
| do |
| {printf(“%d\n”, \_\_(1)\_\_; |
| \_\_(2)\_\_; |
| }while\_\_(3)\_\_; |
| } |
| 【答案】 |
| 选项1、p->data |
| 选项2、p=p->next |
| 选项3、p!=NULL |
| 36. 设有一个头指针为head的不带头结点单向链表，p、q是指向链表中结点类型的指针变量，p指向链表中结点a, （设链表中没有结点的数据域与结点a的数据域相同）,写出相关语句 |
| (1)使该单向链表成为单向循环链表 |
| (2)插入结点s,使它成为a结点的直接前驱 |
| q=p; x=p->data; |
| while \_\_(1)\_\_）q=q->next; |
| q->next=head; |
| q=p; p=p->next; |
| while(p->data!=x) |
| { q=p; |
| \_\_(2)\_\_ |
| } |
| s->next=p; |
| \_\_(3)\_\_ |
| 【答案】 |
| 选项1、q->next!=NULL |
| 选项2、p=p->next |
| 选项3、q->next=s |
| 形考任务二 |
| 一、单项选择题 |
| 1.若让元素1，2，3依次进栈，则出栈顺序不可能为（ ）。 |
| 【答案】3，1，2 |
| 2.一个队列的入队序列是1，2，3，4。则队列的输出序列是（ ）。 |
| 【答案】1，2，3，4 |
| 3.向顺序栈中压入新元素时，应当（ ）。 |
| 【答案】先移动栈顶指针，再存入元素 |
| 4.在一个栈顶指针为top的链栈中，将一个p指针所指的结点入栈，应执行（ ）。 |
| 【答案】p->next=top;top=p; |
| 5.在一个栈顶指针为top的链栈中删除一个结点时，用 x保存被删结点的值，则执行（ ）。 |
| 【答案】x=top->data;top=top->next; |
| 6.判断一个顺序队列（最多元素为m）为空的条件是（ ）。 |
| 【答案】front==rear |
| 7.判断一个循环队列为满的条件是（ ）。 |
| 【答案】(rear+1)%MaxSize==front |
| 8.判断栈满（元素个数最多n个）的条件是（ ）。 |
| 【答案】top==n-1 |
| 9.设有一个20阶的对称矩阵A（第一个元素为a1,1），采用压缩存储的方式，将其下三角部分以行序为主序存储到一维数组B中（数组下标从1开始）， 则矩阵元素a6,2在一维数组B中的下标是（ ）。 |
| 【答案】17 |
| 10.在解决计算机主机与打印机之间速度不匹配问题时通常设置一个打印数据缓冲区，主机将要输出的数据依次写入缓冲区中，而打印机则从缓冲区中取出数据打印，该缓冲区应该是一个（ ）结构。 |
| 【答案】队列 |
| 11.一个递归算法必须包括（ ）。 |
| 【答案】终止条件和递归部分 |
| 12.在一个链队中，假设f和r分别为队头和队尾指针，则删除一个结点的运算为（ ）。 |
| 【答案】f=f->next; |
| 13.在一个链队中，假设f和r分别为队头和队尾指针，则插入s所指结点的运算为（ ）。 |
| 【答案】r->next=s;r=s; |
| 14.数组a经初始化char a[ ]=“English”;a[7]中存放的是（ ）。 |
| 【答案】字符串的结束符 |
| 15.设主串为“ABcCDABcdEFaBc”，以下模式串能与主串成功匹配的是（ ）。 |
| 【答案】Bcd |
| 16.字符串 a1="AEIJING"，a2="AEI"，a3="AEFANG"，a4="AEFI"中最大的是（ ）。 |
| 【答案】a1 |
| 17.两个字符串相等的条件是（ ）。 |
| 【答案】两串的长度相等，并且对应位置上的字符相同 |
| 18.一维数组A采用顺序存储结构，每个元素占用6个字节，第6个元素的存储地址为100，则该数组的首地址是（ ）。 |
| 【答案】70 |
| 19.一个非空广义表的表头（ ）。 |
| 【答案】可以是子表或原子 |
| 20.对稀疏矩阵进行压缩存储,可采用三元组表，一个10 行8列的稀疏矩阵A，其相应的三元组表共有6个元素，矩阵A共有（ ）个零元素。 |
| 【答案】74 |
| 21.对稀疏矩阵进行压缩存储,可采用三元组表，一个10 行8列的稀疏矩阵A共有73个零元素，A的右下角元素为6，其相应的三元组表中的第7个元素是（ ）。 |
| 【答案】（10，8，6） |
| 22.对一个栈顶指针为top的链栈进行入栈操作，通过指针变量p生成入栈结点，并给该 结点赋值a，则执行: p=(struct node \*)malloc(sizeof(struct node);p->data=a;和（ ）。 |
| 【答案】p->next=top;top=p; |
| 23.头指针为head的带头结点的单向链表为空的判定条件是（ ）为真。 |
| 【答案】head->next==NULL |
| 24.设有一个对称矩阵A，采用压缩存储的方式，将其下三角部分以行序为主序存储到一维数组B中（数组下标从1开始），B数组共有55个元素,则该矩阵是（ ）阶的对称矩阵。 |
| 【答案】10 |
| 25.数组a经初始化char a[ ]=“English”;a[1]中存放的是（ ）。 |
| 【答案】字符n |
| 二、判断题 |
| 26. 设有一个链栈，栈顶指针为hs，现有一个s所指向的结点要入栈，则可执行操作。hs=s； |
| s-> next=hs; |
| 【答案】错 |
| 27. 设有一个非空的链栈，栈顶指针为hs，要进行出栈操作，用x保存出栈结点的值，栈 |
| 结点的指针域为next，则可执行hs=hs->next ;x=hs->data; |
| 【答案】错 |
| 28. 有一个链栈，栈顶指针为h，现有一个p所指向的结点要入栈，则可执行操作p->next=h; |
| 和h=p； |
| 【答案】对 |
| 29. 设有一个非空的链栈，栈顶指针为hs，要进行出栈操作，用x保存出栈结点的值，栈结点的指针域为next，数据域为data，则可执行hs= hs->next; x= hs->data; |
| 【答案】错 |
| 30. 在一个链队中，f和r分别为队头和队尾指针，队结点的指针域为next，则插入所指结点的操作为r->next=s；r=s； |
| 【答案】对 |
| 31. 在一个链队中，f和r分别为队头和队尾指针，队结点的指针域为next，s指向一个要入 队的结点，则入队操作为r=s；r->next=s； |
| 【答案】错 |
| 32. 在一个不带头结点的非空链队中，f和r分别为队头和队尾指针，队结点的数据域为data，指针域为next，若要进行出队操作，并用变量x存放出队元素的数据值，则相关操作为x=f->data; f=f->next; |
| 【答案】对 |
| 33.对稀疏矩阵进行压缩存储,可采用三元组表，一个6行7列的稀疏矩阵A相应的三元组表共有8个元素,则矩阵A共有34个零元素。 |
| 【答案】对 |
| 34. 循环队列的最大存储空间为MaxSize，队头指针为f，队尾指针为r，当（r+1）%MaxSize=f 时表明队列已满。 |
| 【答案】对 |
| 35.循环队列的队头指针为f，队尾指针为r，当r= =f时表明队列已满。 |
| 【答案】错 |
| 36.空串的长度是0；空格串的长度是空格字符的个数。 |
| 【答案】对 |
| 37. 对稀疏矩阵进行压缩存储，矩阵中每个非零元素对应的三元组包括该元素的行下标、列下标、和非零元素值三项信息。 |
| 【答案】对 |
| 38.循环队列的引入，目的是为了克服假上溢。 |
| 【答案】对 |
| 39.设有n阶对称矩阵A，用一维数组s压缩存储A的下三角元素，s的下标从零开始，元素 s[26]相应于A中的元素为a 7,5。 |
| 【答案】错 |
| 40. 循环队列的最大存储空间为MaxSize=6，采用少用一个元素空间以有效的判断栈空或栈满，若队头指针front=4，当队尾指针rear=3时队满。 |
| 【答案】对 |
| 41. 循环队列的最大存储空间为MaxSize=6，采用少用一个元素空间以有效的判断栈空或栈满，若队头指针front=4，队尾指针rear=3时，队列中共有5个元素。 |
| 【答案】对 |
| 三、程序选择填空题（请点击正确选项，然后拖拽至相应的方框上） |
| 42.以下函数为链栈的进栈操作，x是要进栈的结点的数据域，top为栈顶指针 |
| struct node |
| { ElemType data; |
| struct node \*next; |
| }; |
| struct node \*top ; |
| void Push(ElemType x) |
| { |
| struct node \*p; |
| p=(struct node\*)malloc \_\_(1)\_\_; |
| p->data=x; |
| \_\_(2)\_\_; |
| \_\_(3)\_\_; |
| } |
| 【答案】 |
| 选项1、sizeof (struct node) |
| 选项2、p->next=top |
| 选项3、top=p |
| 43. 以下函数为链队列的入队操作，x为要入队的结点的数据域的值，front、rear分别链队列的队头、队尾指针 |
| struct node |
| { ElemType data; |
| struct node \*next; |
| }; |
| struct node \*front，\*rear; |
| void InQueue(ElemType x) |
| { |
| struct node \*p; |
| p= (struct node\*) malloc \_\_(1)\_\_; |
| p->data=x; |
| p->next=NULL; |
| \_\_(2)\_\_; |
| rear= \_\_(3)\_\_; |
| } |
| 【答案】 |
| 选项1、(sizeof (struct node) |
| 选项2、rear->next=p |
| 选项3、p |
| 形考任务三 |
| 一、单项选择题 |
| 1.假定一棵二叉树中，双分支结点数为15，单分支结点数为30，则叶子结点数为（ ）。 |
| 【答案】16 |
| 2.二叉树第k层上最多有（ ）个结点。 |
| 【答案】2k-1 |
| 3.将含有150个结点的完全二叉树从根这一层开始，每一层从左到右依次对结点进行编号，根结点的编号为1，则编号为69的结点的双亲结点的编号为（ ）。 |
| 【答案】34 |
| 4.如果将给定的一组数据作为叶子数值，所构造出的二叉树的带权路径长度最小，则该树称为（ ）。 |
| 【答案】哈夫曼树 |
| 5.在一棵度具有5层的满二叉树中结点总数为（ ）。 |
| 【答案】31 |
| 6.一棵完全二叉树共有6层，且第6层上有6个结点，该树共有（ ）个结点。 |
| 【答案】37 |
| 7.利用3、6、8、12这四个值作为叶子结点的权，生成一棵哈夫曼树，该树中所有叶子结点中的最长带权路径长度为（ ）。 |
| 【答案】18 |
| 8.在一棵树中，（ ）没有前驱结点。 |
| 【答案】树根结点 |
| 9.设一棵采用链式存储的二叉树，除叶结点外每个结点度数都为2，该树结点中共有20个指针域为空,则该树有（ ）个叶结点。 |
| 【答案】10 |
| 10.在一个图G中，所有顶点的度数之和等于所有边数之和的（ ）倍。 |
| 【答案】2 |
| 11.邻接表是图的一种（ ）。 |
| 【答案】链式存储结构 |
| 12.图的深度优先遍历算法类似于二叉树的（ ）遍历。 |
| 【答案】先序 |
| 13.已知下图所示的一个图，若从顶点V1出发，按深度优先搜索法进行遍历，则可能得到的一种顶点序列为（ ）。 |
| 【答案】V1V2V4V8V5V3V6V7 |
| 14.已知如下图所示的一个图，若从顶点a出发，按广度优先搜索法进行遍历，则可能得到的一种顶点序列为（ ）。 |
| 【答案】aecbdf |
| 15.图状结构中数据元素的位置之间存在（ ）的关系。 |
| 【答案】多对多 |
| 16.在一棵二叉树中，若编号为i的结点存在右孩子，则右孩子的顺序编号为（ ）。 |
| 【答案】2i+1 |
| 17.一棵具有16个结点的完全二叉树，共有（ ）层。(设根结点在第一层) |
| 【答案】5 |
| 18.对二叉排序树进行（ ）遍历，可以使遍历所得到的序列是有序序列。 |
| 【答案】中序 |
| 19.已知一个图的边数为m，则该图的所有顶点的度数之和为（ ）。 |
| 【答案】2m |
| 二、判断题 |
| 20.一棵二叉树的叶结点（终端结点）数为5，单分支结点数为2，该树共有11个结点。 |
| 【答案】对 |
| 21.一棵有14个结点的完全二叉树，则它的最高层上有7个结点。 |
| 【答案】对 |
| 22.一棵二叉树有6个叶结点，则该树总共有11个结点。 |
| 【答案】错 |
| 23.根据搜索方法的不同，图的遍历有．先序；中序；后序三种方法。 |
| 【答案】错 |
| 24.对于一棵具有n个结点的二叉树，其相应的链式存储结构中共有n-1个指针域空。 |
| 【答案】错 |
| 25. 设一棵完全二叉树，其最高层上最右边的叶结点的编号为奇数，该叶结点的双亲结点的编号为10，该完全二叉树一共有21个结点。 |
| 【答案】对 |
| 26. 设一棵完全二叉树，其最高层上最右边的叶结点的编号为偶数，该叶结点的双亲结点的编号为9，该完全二叉树一共有19个结点。 |
| 【答案】错 |
| 27.按照二叉树的递归定义，对二叉树遍历的常用算法有深度优先遍历和深度优先遍两种方法。 |
| 【答案】错 |
| 28.一棵有8个权重值构造的哈夫曼数，共有17个结点。 |
| 【答案】错 |
| 29.一棵有7个叶结点的二叉树，其1度结点数的个数为2，则该树共有15个结点。 |
| 【答案】对 |
| 三、程序填空题（请点击正确选项，然后拖拽至相应的方框上） |
| 30. 以下程序是后序遍历二叉树的递归算法的程序，完成程序中空格部分（树结构中左、右指针域分别为left和right，数据域data为字符型，BT指向根结点）。完成程序中空格部分。 |
| void |
| Inorder (struct BTreeNode \*BT) |
| { |
| if( BT!=NULL) |
| { |
| Inorder(BT->left); |
| \_\_(1)\_\_ |
| \_\_(2)\_\_ |
| } |
| 利用上述程序对左图进行后序遍历，结果是\_\_(3)\_\_; |
| 【答案】 |
| 选项1、Inorder(BT-> right ) |
| 选项2、printf(“%c”,BT->data) |
| 选项3、d,e,b,f,c,a |
| 31. 以下程序是中序遍历二叉树的递归算法的程序，完成程序中空格部分（树结构中左、右指针域分别为left和right，数据域data为字符型，BT指向根结点）。 |
| void Inorder (struct BTreeNode \*BT) |
| { |
| if(BT!=NULL){ |
| Inorder(BT->left);} |
| \_\_(1)\_\_; |
| \_\_(2)\_\_; |
| } |
| 利用上述程序对右图进行中序遍历，结果是\_\_(3)\_\_; |
| 【答案】 |
| 选项1、printf(“%c”,BT->data) |
| 选项2、Inorder(BT->right) |
| 选项3、d,b,e,a,f,c |
| 四、综合应用题 |
| 32.（1）以3，4，5，8，9，作为叶结点的权，构造一棵哈夫曼树。该树的带权路径长度为 1 . |
| （2）权重为3的叶结点的哈夫曼编码为 2 。 |
| (1)【答案】65 |
| (2)【答案】 000 |
| 33.（1）以2，3，4，7，8，9作为叶结点的权，构造一棵哈夫曼树，该树的带权路径长度为 1 |
| （2）权重值为4的叶结点的哈夫曼编码为 2 。 |
| (1)【答案】80 |
| (2)【答案】 001 |
| 34.（1）已知某二叉树的后序遍历序列是debca，中序遍历序列是dbeac，该二叉树的根结点是 1 |
| （2）先序遍历序列是 2 。 |
| 【答案】 |
| (1)D. a |
| (2)C. a,b,d,e,c |
| 35.（1）已知某二叉树的先序遍历序列是aecdb，中序遍历序列是eadcb，该二叉树的根结点是 1 ; |
| （2）后序遍历序列为 2 。 |
| 【答案】 |
| (1)D. a |
| (2)A. e,d,b,c,a |
| 36.（1）以给定权重值5，6，17，18，25，30，为叶结点，建立一棵哈夫曼树,该树的中序遍历序列为 1 |
| （2）权重值为6的叶结点的哈夫曼为 2 . |
| 【答案】 |
| (1)5，11，6，28，17，58，30，101，18，43，25 |
| (2)0001 |
| 形考任务四 |
| 一、单项选择题 |
| 1.对线性表进行二分查找时，要求线性表必须（ ）。 |
| 【答案】以顺序存储方式，且数据元素有序 |
| 2.采用顺序查找方法查找长度为n的线性表时，每个元素的平均查找长度为（ ）。 |
| 【答案】(n+1)/2 |
| 3.有一个长度为10的有序表，按折半查找对该表进行查找，在等概率情况下查找成功的平均比较次数为（ ）。 |
| 【答案】29/10 |
| 4.已知一个有序表为{11,22,33,44,55,66,77,88,99}，则顺序查找元素55需要比较（ ）次。 |
| 【答案】5 |
| 5.有数据{53,30,37,12,45,24,96}，从空二叉树开始逐个插入数据来形成二叉排序树，若希望高度最小，应该选择的序列是（ ）。 |
| 【答案】37,24,12,30,53,45,96 |
| 6. 对于顺序存储的有序表{5,12,20,26,37,42,46,50,64}，若采用折半查找，则查找元素26的比较次数是（ ）。 |
| 【答案】4 |
| 7.在所有的排序方法中，关键字比较的次数与记录初始排列秩序无关的是（ ）。 |
| 【答案】直接选择排序 |
| 8.从未排序序列中依次取出元素与已经排好序的序列中的元素作比较。将其放入已排序序列的正确的位置上，此方法称为（ ）。 |
| 【答案】插入排序 |
| 9.依次将每两个相邻的有序表合并成一个有序表的排序方法称为（ ）。 |
| 【答案】归并排序 |
| 10.当两个元素出现逆序的时候就交换位置，这种排序方法称为（ ）。 |
| 【答案】交换排序 |
| 11.每次把待排序的区间划分为左、右两个子区间，其中左区间中记录的关键字均小于等于基准记录的关键字，右区间中记录的关键字均大于等于基准记录的关键字，这种排序称为（ ）。 |
| 【答案】快速排序 |
| 12. 一组记录的关键字序列为（46,20,30,79，56，38，40，84,90,110），利用快速排序，以第一个关键字为分割元素，经过一次划分后结果为（ ）。 |
| 【答案】40，20,30,38，46，56，79，84,90,110 |
| 13.在有序表{10,14，34，43，47，64，75，80，90}中，用折半查找法查找值80时，经（ ）次比较后查找成功。 |
| 【答案】3 |
| 14.对序列（49，38，65，97，76，13，47，50）采用直接插入排序法进行排序，要把第七个元素47插入到已排序中，为寻找插入的合适位置需要进行（ ）次元素间的比较。 |
| 【答案】5 |
| 15.排序方法中，从未排序序列中挑选元素，并将其依次放入已排序序列（初始为空）的一端的方法，称为（ ）排序。 |
| 【答案】选择 |
| 16.一组记录的关键字序列为（26，59，36，18，20，25），利用堆排序的方法建立的初始小根堆为（ ）。 |
| 【答案】18，20，25，59，26，36 |
| 17.一组记录的关键字序列为（25，48，16，35，79，82，23，40，36，72），其中，含有5个长度为2的有序表，按归并排序的方法对该序列进行一趟归并后的结果为（ ）。 |
| 【答案】16，25，35，48，23，40，79，82，36，72 |
| 18.已知10个数据元素为（54，28，16，34，73，62，95，60，26，43），对该数列从小到大排序，经过一趟冒泡排序后的序列为（ ）。 |
| 【答案】28，16，34，54，62，73，60，26，43，95 |
| 19.一组记录的关键字序列为（46，79，56，38，40，84），利用快速排序，以第一个关键字为分割元素，经过一次划分后结果为（ ）。 |
| 【答案】40，38，46，56，79，84 |
| 20.一组记录的关键字序列为（80,57,41,39,46,47），利用堆排序（堆顶元素是最小元素）的方法建立的初始堆为（ ）。 |
| 【答案】39，46，41，57，80，47 |
| 二、程序填空题（请点击正确选项，然后拖拽至相应的方框上） |
| 21. 以下函数是二叉排序树的查找算法，若二叉树为空，则返回根结点的指针，否则，返回值是指向树结点的结构指针p（查找成功p指向查到的树结点，不成功p指向为NULL）完成程序中的空格 |
| typedef struct Bnode |
| { int key; |
| struct Bnode \*left; |
| struct Bnode \*right; |
| } Bnode; |
| Bnode \*BSearch(Bnode \*bt, int k) |
| /\* bt用于接收二叉排序树的根结点的指针，k用以接收要查找的关键字\*/ |
| { Bnode \*p; |
| if(bt== \_\_(1)\_\_) |
| return (bt); |
| p=bt; |
| while(p->key!= \_\_(2)\_\_) |
| { if(k<p->key) |
| \_\_(3)\_\_; |
| else \_\_(4)\_\_; |
| if(p==NULL) break; |
| } |
| return(\_\_(5)\_\_; |
| } |
| 【答案】 |
| 选项1、NULL |
| 选项2、k |
| 选项3、p=p->left |
| 选项4、p=p->right |
| 选项5、p |
| 22. 以下程序是折半插入排序的算法 |
| 设待排序的记录序列存放在a[1],…a[n]中，以a[0]作为辅助工作单元，程序是要把a[i] 插入到已经有序的序列a[1],…a[i-1]中。 |
| void binsort (NODE a[ ],int n) |
| { int x,i,j,s,k,m; |
| for （i=2；i<=\_\_(1)\_\_;i++） |
| { a[0]=a[i]; |
| x= a[i].key; |
| s=1; |
| j=i-1; |
| while (s<=j) |
| { m=\_\_(2)\_\_ |
| if( x<a[m].key) |
| \_\_(3)\_\_ |
| else |
| \_\_(4)\_\_ |
| } |
| for ( k=i-1;k>=j+1;k- -) |
| \_\_(5)\_\_=a[k]; |
| a[j+1]=a[0]; |
| } |
| } |
| 【答案】 |
| 选项1、n |
| 选项2、(s+j)/2 |
| 选项3、j=m-1 |
| 选项4、s=m+1 |
| 选项5、a[k+1] |
| 三、综合题 |
| 23. （1）设查找表为(1,10,11,14,23,27,29,55,68) ，画出对上述查找表进行折半查找所对应的判定树，为了成功查找到元素14,需要依次与元素 1 进行比较。 |
| （2）在等概率条件下，成功查找的平均比较次数为 2 。 |
| 【答案】 |
| (1)23,10,11,14 |
| (2)25 /9 |
| 24. （1）一组记录的关键字序列为（47，80，57，39，41，46），利用堆排序的方法建立的初始堆为 1 （堆顶元素是最小元素,采用树的形式建堆）。 |
| （2）输出堆顶元素后,调整后的堆为 2 。 |
| 【答案】 |
| (1)39,41,46,80,47,57 |
| (2)41,47,46,80,57 |
| 25. （1）对关键字序列(56,51,71,54,46,106),利用快速排序，以第一个关键字为分割元素，经过一次划分后结果为 1 ； |
| （2）一组记录的关键字序列为（ 60,47，80，57， 39，41，46,30），利用归并排序的方法,经过(2,2)归并的结果序列为 2 。 |
| 【答案】 |
| (1)46,51,54,56,71,106 |
| (2)(47, 57, 60, 80, 30,39,41,46 ) |
| 26. （1）对关键字序列(36,69,46,28,30,74)采用快速排序，以第一个关键字为分割元素，经过一次划分后的结果序列为 1 |
| （2）用冒泡法对上述序列排序,经两趟冒泡的结果序列为 2 。 |
| 【答案】 |
| (1)30，28，36，46，69，74 |
| (2)36,28,30,46,69,74 |
| 27. （1）一组记录的关键字序列为{45，40，65，43，35，95}写出利用快速排序的方法，以第一个记录为基准得到的一趟划分的结果为 1 ； |
| （2）对上述序列 利用直接插入排序，逐次插入过程中，共进行了 2 次元素间的比较。 |
| 【答案】 |
| (1)35 40 43 45 65 95 |
| (2)10 |