

一、填空题（前 17 空每 1 分，后面 4 空每空 2 分，共 25 分）

1. 根据数据元素之间关系的不同特性，通常有四类基本结构，即：集合、线性结构、树形结构和__ (1) __结构。
2. 数据类型是一个值的集合和定义在这个值集上的一组__ (2) __的总称。
3. 算法的性能主要由时间复杂度和__ (3) __复杂度表示。
4. 队列的操作特点是“先进先出”，堆栈的操作特点是__ (4) __。
5. 用长度为 m 个元素的 C 语言数组实现循环队列，若 f, r 分别表示队头和队尾下标，其中队头下标指向队头元素，队尾下标指向队尾元素后面的一个空闲位置，入队方向为下标加 1 方向，则在少用一个元素空间的前提下，队满的判断条件是__ (5) __。
6. 拥有 100 个结点的完全二叉树中，叶子结点数是__ (6) __。
7. 广义表 $A=(((), (a)))$ ，则 $HEAD(A)=$ __ (7) __， $TAIL(A)=$ __ (8) __。
8. 若 20 行 \times 20 列的下半三角矩阵(含主对角线元素)采用一维数组 A 以列序为主序顺序存储各元素。假设矩阵以及一维数组下标均从 0 开始，则一维数组元素 $A[121]$ 在矩阵中对应元素的行下标是__ (9) __，列下标是__ (10) __。
9. 已知两个带附加头结点的单链表，每个链表的数据结点按升序连接，下面的函数不另辟存储空间，实现将两个升序单链表归并为一个升序单链表，请填空。

已知结点结构定义为 `typedef struct node { int data; struct node *next; } LNode;`

`LNode *merge(LNode *h1, LNode *h2) // h1, h2 传入两个升序链表的附加头结点的指针`

```
{ p1=h1->next; p2=h2->next;
  last=h1; delete h2; //附加头结点*h1 作为归并后的链表附加头结点
  while(____ (11) ____ )
  { if(p1->data____ (12) ____ p2->data)
    { last->next=p1; p1=p1->next; }
    else
    { last->next=p2; p2=p2->next; }
    last=____ (13) ____;
  }
  if(p1) last->next=p1;
  if(p2) last->next=____ (14) ____;
  return h1;
}
```

10. 若二叉树结点指针类型定义如下：

`typedef struct bt_node { int data; struct bt_node *left, *right; } *BT;`

下面的 C++ 函数用先根遍历算法将所有叶子结点按 `right` 指针域从左向右串接成单向链表，请填空。

`void f(BT root, BT &p, BT &h) //h 返回最左边的叶子(单链表头)结点的指针`

```
{ if(root)
  { if(____ (15) ____ )
```

```

        { if(h==NULL) { h=____(16)____; p=h; /*处理最左边的叶子*/ }
          else { p->right=root; p=____(17)____; }
        }
    f(root->left, p, h);
    f(root->right, p, h);
}
}

```

已知主程序中，上述函数的调用方式为：

`h=NULL; f(bt, p, h);` /* bt 为二叉树根结点指针 */

(以下各小题每空 2 分)

11. 若有 C 语言二维数组定义 `double a[4][5]`; 已知 `sizeof(double)=8`, 数组按行优先存储, 若数组首地址为 0, 则元素 `a[3][2]` 的首地址是____(18)____。
12. 后缀表达式 `a, b, c, +, *, d, -, e, /` 对应的前缀表达式是____(19)____。
13. 考虑 KMP 算法, 子串为 "aaabaa", 则下标从 0 开始的改进的 next 数组元素值是____(20)____。
14. 执行以下程序后, 变量 `c` 的值为____(21)____。(用含 `n` 的数学式表示)
`a=1; c=0; while(a<n) { a*=3; c++; }` /* 已知 `n` 为整型变量, 且 `n` 的初值 > 1 */

二、单项选择题 (10 小题, 每小题 2 分, 共 20 分)

1. 逻辑上可以将数据结构分为____(1)____。
 (A) 动态结构和静态结构
 (B) 线性结构和非线性结构
 (C) 顺序结构和链式结构
 (D) 初等结构和组合结构
2. 以下对顺序表进行的操作中, 时间复杂度一定为 $O(1)$ 的是____(2)____。
 (A) 访问第 i 个元素的前驱
 (B) 插入一个元素到第 i 个元素之前
 (C) 删除第 i 个元素
 (D) 对元素进行排序
3. 若栈的容量为 3, 已知入栈序列为 1, 2, 3, 4, 5 且栈满时不会进行入栈操作, 则以下出栈序列正确的是____(3)____。
 (A) 5, 4, 3, 2, 1 (B) 1, 3, 5, 2, 4 (C) 3, 5, 4, 2, 1 (D) 2, 3, 5, 4, 1
4. 若 10 个结点的完全二叉树采用顺序表存储, 下标范围 $0 \sim 9$, 根结点下标为 0, 则下标为 3 的结点的左孩子的下标是____(4)____。
 (A) 5 (B) 6 (C) 7 (D) 8
5. 栈和队列的共同特点是____(5)____。
 (A) 只允许在端点处插入和删除元素
 (B) 都是先进后出

- (C) 都是先进先出
(D) 没有共同点
6. 用链接方式存储的队列，在进行插入运算时 (6)。
(A) 仅修改头指针 (B) 头、尾指针都要修改
(C) 仅修改尾指针 (D) 头、尾指针可能都要修改
7. 哈夫曼树二叉树叶子结点数为 m ，则该哈夫曼树中总共有 (7) 个空指针域。
(A) $2m-1$ (B) $2m$ (C) $2m+1$ (D) $4m$
8. 以下不属于存储结构的概念是 (8)。
A. 堆栈 B. 单向循环链表 C. 顺序表 D. 二叉链表
9. 设某棵二叉树中有 2000 个结点，则该二叉树的最小高度为 (9)。
(A) 9 (B) 10 (C) 11 (D) 12
10. 设二叉树的先序遍历结点访问次序和后序遍历结点访问次序正好相反，则该二叉树满足的条件是 (10)。
(A) 空或只有一个结点 (B) 高度等于其结点数
(C) 任一结点无左孩子 (D) 任一结点无右孩子

三、简答题（共 35 分）

- 若三个元素入栈次序为 ABC，写出所有可能的出栈次序。(此题 5 分)
- 已知二叉树的先序和中序遍历结点访问次序如下：
先序遍历：ACDEFBGH
中序遍历：CEDFABHG
试画出该二叉树。(此题 5 分)
- 分析以下递归算法。(此题 8 分)

```
int f(int n)
{ if(n==0) return 1;
  int k=n*f(n/2); printf("%d", k);
  return k;
}
```

 - 给出函数调用 $f(7)$ 的输出结果；
 - 若定 $n > 0$ ，试分析函数调用 $f(n)$ 的堆栈最大深度。(用 n 的数学式表示)
- 某森林由以下广义表给出，画出该森林；用孩子兄弟链表将该森林表示为二叉树形式，画出该二叉树。(此题共 7 分)
(A(B, C, D), E(F), G(H, I(J)))
- 8 个符号 A, B, C, D, E, F, G, H 出现的概率依次为 0.05, 0.29, 0.07, 0.08, 0.14, 0.23, 0.03, 0.11。试画出由这 8 个符号构造出的哈夫曼二叉树，给出各符号的二进制编码并计算平均编码长度。(此题共 10 分)

四、算法设计题（2 小题，每小题 10 分，共 20 分）

1. 已知带附加头结点单链表结点结构如本试卷第一大题第 9 小题。

写一个算法，形参传入附加头结点指针，然后删除所有 **data** 域值小于 0 的所有结点。

2. 已知二叉树根结点地址 **bt**，编写算法，求二叉树所有结点 **data** 域值的最小值。

二叉树数据结构可参考本试卷第一大题第 10 小题或自拟。