**华清远见创客学院《数据结构》闯关试题**

1. **选择题(每题2分，共40分**)
2. 下列结构，属于线性结构的是（C）
   1. 二叉树
   2. 有向无环图
   3. 循环队列
   4. 霍夫曼树
3. 对于层次结构，下列描述正确的是（C）
   1. 具有唯一前驱和唯一后继
   2. 具有多个前驱
   3. 后继可以有多个，前驱只有一个
   4. 前驱可以有多个，后继只有一个
4. 下列不属于描述数据逻辑结构的是（C）
   1. 线性表
   2. 树
   3. 图
   4. 链表
5. 数据结构包含的内容不包括（D）
   1. 线性结构
   2. 存储结构
   3. 逻辑结构
   4. 数据运算
6. 下列关于存储结构和逻辑结构描述正确的是（B）
   1. 存储结构和程序具体实现的语言没有关系
   2. 算法的实现需要考虑逻辑结构
   3. 逻辑结构就是实际存储的结构
   4. 算法的实现依赖于存储结构
7. 下列不是一个好的算法应该具有的特点的是（C）
   1. 算法消耗时间少
   2. 算法消耗空间少
   3. 算法的时间复杂度高
   4. 算法的结构简单。
8. 一个算法的时间复杂度为T(n) = 2n3+3n2+4n+5n4+2,则该T(n)应记为（D）
   1. O(n)
   2. O(n2)
   3. O(n3)
   4. O(n4)
9. 对于冒泡排序来说，其算法时间复杂度为（B）
   1. O(n)
   2. O(n2)
   3. O(1)
   4. O(nlogn)
10. 向一个单链表linklist中的节点t后面插入一个节点p，下列操作正确的是（A）
    1. p->next = t->next;t->next = p;
    2. t->next = p->next;t->next = p;
    3. t->next = p;p->next = t->next;
    4. t->next = p;t->next = p->next;
11. 在一个完全二叉树中，编号为i的节点存在左孩子，则左孩子的编号是（C）

设根节点编号为0.

* 1. 2i
  2. 2i - 1
  3. 2i +１
  4. 2i + 2

1. 二叉树的第k层最多有几个节点（D）
   1. 2k-1
   2. 2k+1
   3. 2k+1
   4. 2k-1
2. 设指针变量p指向单链表中节点A，若删除单链表中的节点A，则需要修改指针的操作顺序为（C）
   1. q= p->next;p->data = q->data;p->next = q ->next;free(q);
   2. q = p->next;q->data = p->data;p->next = q->next;free(q);
   3. q = p->next;p->next = q->next;free(q);
   4. q = p->next;p->data = q->data;free(q);
3. 约瑟夫环问题中，有n个人，从第k个人数，数到第m个人出列，若n = 8,k= 3 m = 4,则正确的出队序列为：（A）
   1. 62743518
   2. 62751843
   3. 62734518
   4. 62734518
4. 顺序栈和链式栈的主要区别是（A）
   1. 逻辑结构不同
   2. 存储结构不同
   3. 数据类型不同
   4. 数据存取方式不同
5. 深度为k的完全二叉树中，最少有（C）个节点
   1. 2k-1-1
   2. 2k-1
   3. 2k-1+1
   4. 2k-1
6. **简答题(1-3题每题10分，4、5题每题15分，共60分)**
7. 简述以下算法的功能。

void BB(LNode \*s,LNode \*q)

{

p=s;

while(p->next!=q)p=p->next;

p->next=s;

}

void AA(LNode \*pa,LNode \*pb)

{

//pa和pb分别指向单循环链表的两个结点

BB(pa,pb);

BB(pb,pa);

}

将一个单循环链表拆分为两个单循环链表，pa和pb为断点

1. 写出程序删除单链表中的所有节点

void listdel(listlink H){

listlink p = H, q;

while(p){

q = p->next;

free(p);

p = q;

}

}

1. 写出一个循环队列的结构体，该队列元素最多是m个。
2. 写出如何判断该队列为空的函数
3. 写出如何判断该队列为满的函数
4. 写出计算当前队列中的元素个数的函数

结构体：

typedef int datatype;

typedef struct queue

{

int \*pBase;

int front; //指向队列第一个元素

int rear; //指向队列最后一个元素的下一个元素

int maxsize; //循环队列的最大存储空间

}loopque,\*ploopque;

元素入列时：rear = (rear+1)%m;

元素出列时：front = (front+1)%m;

1. 当rear = front 时，队列为空

bool EmptyQueue(ploopque Q)

{

if(Q->front==Q->rear) //判断是否为空

return true;

else

return false;

}

1. 当（rear + 1）% m = front时，队列为满

bool FullQueue(ploopque Q)

{

if(Q->front==(Q->rear+1)%Q->maxsize) //判断循环链表是否满，留一个预留空间不用

return true;

else

return false;

}

(3)

int CountQueue(ploopque Q)

{

if(Q->front > Q->rear)

return Q->rear + Q->maxsize – Q->front;

else

return Q->rear - Q->front ;

}