广东工业大学

2011 年攻读硕士学位研究生入学考试试题

考试科目(代码)名称：（831）数据结构与程序设计基础　　满分：150

（考生注意：答卷封面需填写自己的准考证编号，答完后连同本试题一并交回！）

答题注意：请在答题纸上按题目顺序答题，在试题上答题无效！

一．单项选择题(共 50 分，每小题 2 分)

1. 设有说明“**int** i; **char** c; **float** f; ”，结果为整数的表达式是（　　）。

   [A] i*f　　[B] i+c　　[C] c+f　　[D] i+c-f

2. 使 i 的运算结果为 5 的程序段是（　　）。

   [A] **int** i=0,j=0; (i=3,(j++)+i);
   [B] **int** i=1,j=0; j=i=((i=3)*2);
   [C] **int** i=0,j=1; (j==1)?(i=4):(i=3);
   [D] **int** i=1,j=1; i+=j+=3;

3. 设有说明“**int** a=3,b=4;”， a+++b 的结果是（　　）。

   [A] 8　　[B] 7　　[C] 6　　[D] 5

4. 执行以下程序段的结果是（　　）。

```
int a=4,b=1,c=2;
float x=10.5,y=4.0,z;
z=(a+b)/c+sqrt((double)y)*1.2/c+x;
printf("%f\n",z);
```

   [A] 14.000000　　[B] 15.000000　　[C] 13.700000　　[D] 14.900000

5. 执行以下程序段的结果是（　　）。

```
int a=2,c=5;  printf("a=%%d,b=d\n",a,c);
```

   [A] a=%d,b=d　　[B] a=2,b=5　　[C] a=%2,b=%5　　[D] a=%%2,b=5

6. 计算以下函数的程序段是（　　）。

$$y=\begin{cases}-1 & x<0\\ 0 & x=0\\ 1 & x>0\end{cases}$$

```
[A]y=-1;                [B] if(x>=0)            [C] y=0;                [D] y=-1;
    if(x!=0)                if(x>0)y=1;             if(x>=0)                if(x>0)y=1;
    if(x>0)y=1;             else y=0;               if(x>0)y=1;             else y=0;
    else y=-1;              else y=-1;              else y=-1;
```

7. 条件“!r”等价于（　　）。

   [A] r==0　　[B] r==1　　[C] r!=0　　[D] r!=1

121

8. 以下 **for** 语句的循环次数是（　　）。

```
for(x=0, y=0; (x<8)&&(y!=123); ++x, y--);
```

[A] 7 次　　[B] 8 次　　[C] 无限次　　[D] 不确定

9. 执行“**int** a[][3]={6,5,4,3,2,1};”后，a[1][1]的值是（　　）。

[A] 4　　[B] 3　　[C] 2　　[D] 1

10. 执行以下程序的结果是（　　）。

```
int func(int a) { int b=0; static c=3; b++; c++; return a+b+c; }
void main() { for(int a=2,i=0; i<3; i++) printf("%d", func(a)); }
```

[A] 777　　[B] 7 7 7　　[C] 789　　[D] 7 8 9

11. 执行以下程序的结果是（　　）。

```
int func(int x) {
   if (!(x<0 || x>1)) return 3;
   int y=x-func(x-2);
   return y;
}
void main() { printf("%d\n", func(9)); }
```

[A] 0　　[B] 2　　[C] 3　　[D] 7

12. 在以下结构体定义中，不正确的是（　　）。

[A]
```
struct student {
   char name[10];
   int  no;
   float score;
};
```

[B]
```
struct stud[20] {
   char name[10];
   int  no;
   float score;
}
```

[C]
```
struct student {
   char name[10];
   int  no;
   float score;
} stud[20];
```

[D]
```
struct {
   char name[10];
   int  no;
   float score;
} stud[20];
```

13. 评价一个算法时间性能的主要标准是（　　）。

[A] 算法易于调试　　[B] 算法易于理解

[C] 算法的稳定性和正确性　　[D] 算法的时间复杂度

14. 对于只在表的首、尾两端进行插入操作的线性表，宜采用的存储结构为（　　）。

[A] 顺序表　　[B] 设头指针的单循环链表

[C] 单链表　　[D] 设尾指针的单循环链表

15. 链表不具有的特点是（　　）。

[A] 不必事先估计存储空间　　[B] 所需空间与表的长度成正比

[C] 可以随机访问任一结点　　[D] 插入和删除不需要移动元素

16. 一个栈的输入序列为 12345，其合法的输出序列是（　　）。

[A] 14253　　[B] 23415　　[C] 31245　　[D] 54132

122

17．上溢现象通常出现在（　　）。

[A] 顺序栈的入栈操作中　　[B] 顺序栈的出栈操作中

[C] 链栈的入栈操作中　　[D] 链栈的出栈操作中

18．在一棵二叉树中，含有度为 1 的结点 2 个和度为 2 的结点 15 个，则度为 0 的结点数为（　　）。

[A] 13　　[B] 15　　[C] 16　　[D] 17

19．下列陈述中正确的是（　　）。

[A] 二叉树是度为 2 的有序树　　[B] 二叉树中结点只有一个孩子时无左右之分

[C] 二叉树中必有度为 2 的结点　　[D] 二叉树中结点最多只有两个孩子，并且有左右之分

20．有向图的一个顶点的度是该顶点的（　　）。

[A] 入度　　[B] 出度　　[C] 入度与出度之和　　[D] 入度与出度的均值

21．假设一个含有 n 个顶点和 e 条弧的有向图采用邻接表表示，则删除与某个顶点 v 相关的所有弧的时间复杂度是（　　）。

[A] O(n)　　[B] O(e)　　[C] O(n2)　　[D] O(n*e)

22．从空树开始，依次插入元素 55，22，63，47，98，13，71，90，34 和 85 后构成了一棵二叉排序树。在该树查找 71 要进行比较次数为（　　）。

[A] 3　　[B] 4　　[C] 5　　[D] 6

23．设顺序存储的线性表共有 123 个元素，按分块查找的要求等分成 3 块。若对索引表采用顺序查找来确定块，并在确定的块中进行顺序查找，则在查找概率相等的情况下，分块查找成功时的平均查找长度为（　　）。

[A] 21　　[B] 23　　[C] 41　　[D] 62

24．在最好和最坏情况下的时间复杂度均为 O(nlogn) 且稳定的排序方法是（　　）。

[A] 归并排序　　[B] 堆排序　　[C] 快速排序　　[D] 基数排序

25．在下列排序方法中，平均时间性能为 O(nlogn) 且空间性能最好的是（　　）。

[A] 堆排序　　[B] 快速排序　　[C] 归并排序　　[D] 基数排序

二．解答题（共 20 分）

26．（6 分）已知某系统在通信中只使用 8 种字符，其频率分别为

A(0.05), B(0.29), C(0.07), D(0.09), E(0.13), F(0.23), G(0.03), H(0.11)

请为其构造哈夫曼树（权值小的结点在左）。

27．（6 分）已知有向图 G 定义如下：

```
G = (V, E)
V = { a, b, c, d, e, f }
E = { <a,b>,<a,c>,<a,d>,<c,b>,
      <c,e>,<d,e>,<f,d>,<f,e> }
```

写出 G 的全部拓扑序列。

123

28. （8 分）设哈希函数为 H(k)=k % 11，用二次探测法处理冲突。请画出依次插入元素 29，15，48，47，23，41，73，37 后，该哈希表的状态，在各元素下面标出其冲突次数，并求出查找成功的平均查找长度。

三. 分析题（共 40 分，每小题 8 分）

29. 阅读算法 f29，并回答下列问题：

（1）写出调用 f29("C","A","T")时算法的输出；

（2）简述算法 f29(x,y,z)的功能。

```
void f29(char x, char y, char z) {
   int t;
   if (x>y) { t=x;  x=y;  y=t; }
   if (x>z) { t=x;  x=z;  z=t; }
   if (y>z) { t=y;  y=z;  z=t; }
   printf("%c,%c,%c\n", x, y, z);
}
```

30. 阅读算法 f30，并回答下列问题：

（1）依次写出，调用 f30(a,5)后数组 a 的元素 a[0],a[1],a[2],a[3]和 a[4]的值；

（2）简述执行算法 f30(a,n)后，数组 a 的元素 a[i]的值的一般形式（0≤i≤n-1）。

```
void f30(int a[], int n) {
   int i,last=1;
   for (i=1; i<=n; i++) {
      a[i-1] = last*2*i;
      last = a[i-1];
   }
}
```

31. 阅读算法 f31，并回答下列问题：

（1）设栈 S=(3，4，5，6，7，8，9)，其中 9 为栈顶元素。写出调用 f31(S)后栈 S 的状态；

（2）简述算法 f31 的功能。

```
void  f31(Stack &S) {
   int i=0;
   Queue Q;  Stack T;
   InitQueue(Q);  InitStack(T);
   while(!StackEmpty(S)) {
      if ((i+1)%2==0) Push(T,Pop(S));
      else EnQueue(Q,Pop(S));
      i++;
   }
   while (!QueueEmpty(Q)) Push(S,DeQueue(Q));
   while (!StackEmpty(T)) Push(S,Pop(T));
}
```

32. 阅读算法 f32，并回答下列问题：

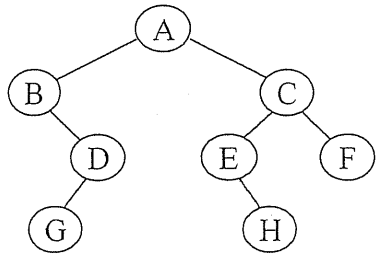
（1）设双向链表 L 形如：L → ∧ | a_1 | ⇄ | a_2 | ⇄ … ⇄ | a_n | ∧

画出调用 f32(L)后 L 的示意图；

（2）简要说明算法 f32(L)的功能。

```
void f32(DuLinkList L) {
   DuLinkList p,q,r,s;
   int even;
   if (L && L->next) {
      p=L;  q=p->next;  s=q->next;
      p->prior=NULL;  q->next=NULL;
      even=1;
   }
   while (s) {
      r=s;  s=s->next;
      p->next=r;   r->prior=p;
      q->prior=r;  r->next=q;
      if (even) p=r;
      else q=r;
      even=!even;
   }
}
```

33. 如图所示的二叉树 T 采用二叉链表存储结构，



（1）请画出调用算法 f33(T)后的树 T；

（2）简述算法 f33 对任意二叉树的操作意义。

```
void f33(BiTree &T) {
   if (T) {
      if (!T->lchild && !T->rchild) {
         free(T);
         T=NULL;
      } else {
         f33(T->lchild);
         f33(T->rchild);
      }
   }
}
```


125

四. 填空题（共 40 分，每小题 8 分）

34. 算法 f34 的功能是，输出由数字 1、2、3 和 4 能组成的各位互不相同的所有三位数。请在空缺处填入合适的内容，使其成为完整的算法。

```
void f34( ) {
  int i,j,k;
  for(i=1; i<5; i++)
    for(j=1; j<5; j++)
      for(_____①_____)
        if (i!=k && _____②_____ && _____③_____)
          printf("%1d,%1d,%1d\n", i, _____④_____);
}
```

35. 所谓“水仙花数”是指一个三位数，其各位数字的立方和等于该数本身。例如：153 是一个“水仙花数”，因为 $153=1^3+5^3+3^3$。算法 f35 打印出所有的“水仙花数”。请在空缺处填入合适的内容，使其成为完整的算法。

```
void f35( ) {
  int i,j,k,n;
  for (n=100; n<1000;_____①_____) {
    i = n/100;
    j = _____②_____;
    k = n%10;
    if (i*100+j*10+k == _____③_____)
      printf("%d\n", _____④_____);
  }
}
```

36. 已知 L 为一个带头结点的循环链表。算法 f36 的功能是删除 L 中数据域 data 的值小于 c 的所有结点，并由这些结点组建成一个新的带头结点的循环链表，其头指针作为函数的返回值。请在空缺处填入合适的内容，使其成为完整的算法。

```
LinkList f36(LinkList L, int c) {
  LinkList Lc,p,pre=L;
  p= _____①_____;
  Lc=(LinkList)malloc(sizeof(ListNode));
  Lc->next=Lc;
  while (p != _____②_____)
    if (p->data < c) {
      pre->next=p->next;
      _____③_____;
      Lc->next=p;  p=pre->next;
    } else {
```


126

```
            pre=p;
            _____④_____;
        }
      return Lc;
    }
```

37. 如果二叉树 T 不含度为 1 的结点，则称为正则二叉树。算法 f37(T)判定二叉树 T 是否为正则二叉树。请在空缺处填入合适内容，使其成为完整的算法。

```
int f37(BiTree T) {
  BiTree p;   Queue Q;
  if ( _____①_____ ) return 1;
  InitQueue(Q);  EnQueue(Q,T);
  do {
    DeQueue(Q, p);
    if (p->lchild || p->rchild)
      if ( _____②_____ ) {
        EnQueue(Q, p->lchild);
        _____③_____;
      } else return 0;
  } while (!QueueEmpty(Q));
  return _____④_____;
}
```

38. 图的邻接矩阵存储结构的类型定义如下：

```
typedef struct {
   VertexType vexs[MAX_VERTEX_NUM]; // 顶点向量
   int arcs[MAX_VERTEX_NUM][MAX_VERTEX_NUM]; // 邻接矩阵
   int vexNum,arcNum;  // 图的当前顶点数和弧数
   int kind;           // 图的种类标志
} MGraph;
```

算法 f38(G, v)在图 G 中查找顶点 v 的第一个邻接点，若存在，则返回其在顶点向量 vexs 中的下标，否则返回-1。请在空缺处填入合适的内容，使其成为完整的算法。

```
int f38(MGraph G, VertexType v) {
  int i=0, j=0;
  while (i<G.vexNum && _____①_____ )
    i++;
  while (j<G.vexNum && _____②_____ )
    j++;
  if ( _____③_____ ) return j;
  else return _____④_____;
}
```


127