

广东工业大学

2014 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 C 语言

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, 25 小题, 每题 2 分)

- 在长度为 n 的顺序表中删除第 i 个元素的算法的平均时间复杂度为 ()
A. $O(1)$ B. $O(i)$ C. $O(n)$ D. $O(\log_2 n)$
- 采用链式存储结构便于实现对线性表的 ()
A. 插入操作 B. 遍历操作 C. 查找操作 D. 定位操作
- 若循环队列用数组 $A[0, m-1]$ 存放元素, 其头尾指针分别为 $front$ 和 $rear$, 则当前队列长度为 ()
A. $(rear-front+m) \% m$ B. $rear-front+1$
C. $rear-front-1$ D. $(rear-front+1) \% m$
- 已知广义表 $A = ((a, b, c), (k, (h, i)))$, 从表 A 中取出原子 b 的运算是 ()
A. $tail(tail(head(A)))$ B. $head(tail(tail(A)))$
C. $head(tail(head(A)))$ D. $head(tail(head(tail(A))))$
- 稀疏矩阵的常用压缩方法有两种, 即 ()
A. 二维数组和三维数组 B. 三元组顺序表和散列表
C. 三元组顺序表和十字链表 D. 十字链表和散列表
- 在一棵二叉树中, 度为 2 的结点有 15 个, 度为 1 的结点有 2 个, 则度为 0 的结点数为
A. 13 B. 15 C. 16 D. 17
- 一个具有 n 个结点的有向完全图最多有 () 条弧?
A. $n*(n-1)$ B. $n*(n+1)$ C. $n*(n-1)/2$ D. $n*(n+1)/2$
- 要得到二叉排序树的结点的有序序列, 应对该树进行 ()
A. 先序遍历 B. 中序遍历 C. 后序遍历 D. 层次遍历
- 在对查找表的查找过程中, 若被查找的数据元素不存在, 则把该数据元素插入到查找表的方式主要适合于 ()。
A. 动态查找表 B. 静态查找表
C. 静态查找表与动态查找表 D. 静态查找表或动态查找表
- 时间复杂度为 $O(n \log n)$ 且占用额外空间最少的排序算法是 ()。
A. 堆排序 B. 归并排序 C. 快速排序 D. 希尔排序
- 一个 C 程序的执行是从 ()
A. 本程序文件的第一个函数开始, 到本程序文件的最后一个函数结束
B. 本程序的 $main$ 函数开始, 到本程序文件的最后一个函数结束
C. 本程序的 $main$ 函数开始, 在 $main$ 函数中结束
D. 本程序文件的第一个函数开始, 到本程序 $main$ 函数结束

12. C 语言的基本数据类型包括 ()

- A. 整型、实型、逻辑型 B. 整型、实型、字符型、逻辑型
C. 整型、字符型、逻辑型 D. 整型、实型、字符型

13. 能正确表达逻辑关系 "a<=50" 并且 "a>=0" 的 C 语言表达式是 ()

- A. a<=50 and a>=0 B. a<=50 & a>=0
C. a<=50 && a>=0 D. a>=50 && a<=0

14. 以下能定义能作为为变量名字的是 ()

- A. 3_a B. int C. break D. _int

15. 一个算法应该具有 "确定性" 等 5 个特性, 下面对另外 4 个特性的描述中错误的是 ()

- A. 有零个或多个输入 B. 有零个或多个输出 C. 有穷性 D. 可行性

16. 与 while(e) 控制效果等价的是 ()

- A. while(e==0) B. while(e!=0) C. while(e==1) D. while(e!=1)

17. 关于字符串和字符的说法, 正确的是 ()

- A. "A" 与 'A' 是相同的 B. 字符串是常量, 字符是变量
C. "A" 与 'A' 是不相同的 D. "A " 与 "A" 是相同的

18. 下列数组初始化不正确的是 ()

- A. int a[5] = {0}; B. int a[5] = 0; C. int a[] = {0,1,2,3}; D. int a[5][3] = {0};

19. 判断字符串是否相等, 下面语句中正确的是 ()

- A. if(str1=str2) B. if(str1==str2) C. if(strcmp(str1, str2)==0) D. if(strcmp(str1, str2))

20. 下面关于结构化算法的说法, 不正确的是 ()

- A. 采用自顶向下、逐步细化的设计方法 B. 至少有一个输出
C. 不存在死循环 D. 提倡使用灵活、简洁的 goto 语句

21. C 程序的最小单位是 ()

- A. 标识符 B. 函数 C. 表达式 D. 语句

22. 若函数调用形式为 f((x1,x2,x3),(y1,y2)), 则函数形参的个数是 ()

- A. 2 B. 3 C. 4 D. 5

23. 下列哪项不属于 C 程序三种基本结构是 ()

- A. 顺序结构 B. 循环结构 C. 跳转结构 D. 选择结构

24. 若有定义: int a[2][3]; 则对 a 数组的第 i 行 j 列元素的正确引用为 ()

- A. *(a[i]+j) B. *(a+i)+j C. *(a+i+j) D. a[i]+j

25. 若有如下定义:

```
struct person { char name[10]; int vote;}p;
```

则能输入字符串 "William" 到 p 的 name 成员的语句是 ()

- A. scanf("%s", &p.name); B. scanf("%s", p.name);
C. scanf("%s", p->name); D. scanf("%s", *p->name);

二. C 程序分析题 (共 26 分, 3 小题, 每题 8 或 9 分)

1. 阅读以下程序, 回答问题 (9 分)

```
#include <stdio.h>
```



```
#define N 20
void f211(char s[], char c)
{ int j=0,k=0;
  While(s[j]!='\0')
  { if(s[j]!=c) s[k++]=s[j];
    j++;
  }
  S[k]='\0';
}
void f212(char s[])
{ int c,j,k;
  for(j=0,k=strlen(s)-1;j<k;j++,k--)
    c=s[j],s[j]=s[k],s[k]=c;
}
void main()
{ int i,n;
  char a[N]="eraec";
  f211(a,'e');
  f212(a);
  printf("%s\n",a);
}
```

- (1) 程序输出的结果 (3分)
- (2) 函数 f211 的功能是什么 (3分)
- (3) 函数 f212 的功能是什么 (3分)

2. 阅读以下程序，回答问题 (8分)

```
#include <stdio.h>
#define N 20
int f221 ( char s[] , char t[])
{ int j=0;
  while(s[j]==t[j] && s[j]!='\0') j++;
  return s[j]-t[i];
}
void f222 ( char t[] , char s[])
{ int j=0;
  While(t[j]=s[j++]);
}
```

```

void main()
{
    char c[N]="fo3t",d[N]= "fo2t";
    int k,m;
    f221(c,N);
    f222(c,N);
}

```

- (1) 函数 f211 的功能是什么 (4 分)
- (2) 函数 f212 的功能是什么 (4 分)

3. 阅读以下程序，回答问题 (9 分)

```

#include <stdio.h>
#define N 3
void f231(int a[N])
{
    int i=0;
    while(i<=N-1)
    {
        while(a[i]<1 || a[i]>9)
            scanf("%d",&a[i]);
        i++;
    }
}

int f232(int a[N])
{
    int i,x,y;
    for(i=0;i<=N-2;i++)
    {
        x=a[i],y=a[i+1];
        if(x+1==y || x+1-9==y);
        else break;
    }
    if(i==N-1) return 1;
    else return 0;
}

void main()
{
    int a[N];
    a[0]=100;
    f231(a);
    if(f232(a)) printf("Yes\n");
    else printf("No\n");
}

```

(1) 当输入为 9 0 1 2 时, 程序输出的结果 (3 分)

(2) 函数 f231 的功能是什么 (3 分)

(3) 函数 f232 程序的主要功能是什么 (3 分)

三. C 程序填空题 (共 18 分, 2 小题, 每题 9 分, 每空 3 分)

1. 采用一重循环输出二维数组 **a** 的数据, 请在空缺处填写合适的内容。

```
#include <stdio.h>
#define M 3
#define N 4
void main()
{ int a[M][N]={1,2,3,4},{5,6,7,8},{9,10,11,12}};
  int i;
  for(i=0; (1) ;i++)
  { printf("%d ",a[(2)][(3)]);
    if(i%3==0) printf("\n");
  }
}
```

2. 外部输入一串字符, 字符中含有数字字符和非数字字符, 按从左到右的顺序, 把数字字符提取出来并组成一个整数, 比如输入的字符串为 "a10 b08c6", 则输出整数 10086。请在空缺处填写合适的内容(假设输入的字符串长度不超过 9, 且中间允许有空格字符)。

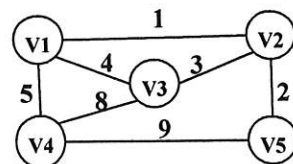
```
#include <stdio.h>
void main()
{ char c[10];
  int i=0;
  long k=0;
  (1) ;
  while( (2) )
  { if(c[i]>='0' && c[i]<='9')
    k= k*10+c[i]-'0';
    (3) ;
  }
  printf("%ld\n",k);
}
```


四. 数据结构解答题 (共 20 分, 3 小题, 每题 6 或 7 分)

1. (7 分) 已知某无向带权图如右图所示,

(1) (4 分) 请写出该图的邻接矩阵存储表示 (两顶点之间无边相连则用 ∞ 表示);

(2) (3 分) 画出该无向带权图在克鲁斯卡尔算法下构造的最小生成树。



2. (7 分) 请画出依次插入元素 29, 15, 48, 47, 23 后的二叉排序树, 并求出在等概率情况下查找成功时的平均查找长度。

3. (6 分) 对关键字序列 (50, 9, 51, 6, 91, 17, 89, 27, 65, 43) 进行快速排序, 将第一个关键字 50 作为枢轴, 使序列按关键字升序排列。请写出快速排序过程中得到的第 1 趟的序列状态。

五. 算法填空题 (共 18 分, 3 小题, 每题 6 分)

1. (6 分) 带头结点的单链表存储结构类型定义如下:

```
typedef struct LNode{
    ElemType    data;
    struct LNode *next;
} LNode, *LinkList;
```

假设某个带头结点的单链表 L 长度 n 大于 1, L 为指向链表的头指针。算法 f51 在 L 的第 i ($1 \leq i \leq n+1$) 个结点之前插入新结点, 新结点由指针 p 指示。请在空缺处填入合适内容, 使其成为完整的算法。

```
void f51(LinkList &L, int i, LinkList p)
{
    int j;
    LinkList q=L;
    for(j = 1; j < i; j++){
        (1) ;
    }
    (2) = q->next;
    (3) = p;
}
```

2. (6 分) 二叉树的存储结构的类型定义如下:

```
typedef struct BiTNode {
    char    data;
    BiTNode *lchild, *rchild;
} BiTNode, *BiTree;
```

已知两棵二叉树 B1 和 B2, 算法 f52 判断 B1 和 B2 是否相似, 相似则返回 1, 否则返

回 0。如果 B1 和 B2 均为空树，则两棵树相似，如果 B1 和 B2 非空且 B1 的左右子树与 B2 的左右子树分别相似，则两棵树相似。请在下面空缺处填入合适内容，使其成为完整的算法。

```
int f52(BiTree t1, BiTree t2){
    if (____(1)____) return 1;
    else if (t1 && t2 &&
             f52(t1->lchild, t2->lchild) &&
             f52(____(2)____))
        return 1;
    else ____ (3) ____;
}
```

3. (6 分) 图的邻接表存储结构的类型定义如下:

```
int visited[20]; //访问标志数组，已访问标记为 1，否则为 0
typedef struct ArcNode {
    int adjvex; //该弧所指向的顶点的位置
    struct ArcNode *nextArc; //指向下一条弧的指针
} ArcNode; //定义弧的结点

typedef struct Vnode{
    VertexType data; //顶点信息
    ArcNode *firstArc; //指向第一条依附该顶点的弧的指针
} VNode, AdjList[20]; //定义顶点数组，顶点数最大为 20

typedef struct {
    AdjList vertices;
    int vexnum, arcnum; //图的当前顶点数和弧数
    int kind;
} ALGraph; //邻接表类型
```

算法 f53 基于图的深度优先搜索策略，判断以邻接表方式存储的有向图 g 是否存在由顶点 i 到顶点 j 的路径 (i 与 j 不相等)，存在路径则返回 1，否则返回 0。请在空缺处填入合适的内容，使其成为完整的算法。

```
int f53(ALGraph g, int i, int j){
    ArcNode *t;
    if(visited[i] || !g.vexnum || !g.arcnum) return ____ (1) ____;
    visited[i] = 1;
    for(t = g.vertices[i].firstarc; t != NULL; t = t->nextarc){
        if(t->adjvex == ____ (2) ____) return 1;
        if(f53(g, ____ (3) ____, j)) return 1;
    }
    return 0;
}
```

六. 算法分析题 (共 18 分, 3 小题, 每题 6 分)

1. (6 分) 已知顺序表类型定义如下:

```
typedef struct {
    char    *elem;
    int     length;
    int     listsize;
} SqList;
```

阅读算法 f61, 并回答下列问题:

(1) 设顺序表指针 exp.elem 所指向的字符串为 "(((())))", 写出执行算法 f61(exp) 后的函数返回值;

(2) 简述算法 f61 的功能。

```
int f61(SqList exp) {
    int i, c=0;
    for (i=0; c>=0 && i<exp.length; i++) {
        if (exp.elem[i] == '(') c++;
        else if (exp.elem[i] == ')') c--;
    }
    if (c==0) return 1;
    else return 0;
}
```

2. (6 分) 已知串的堆存储表示如下所示:

```
typedef struct {
    char *ch;
    int  length;
} HString;
```

阅读下列算法, 并回答问题:

(1) 设串 S1='Sun', 串 S2='day', 写出执行算法 f62(S, S1, S2) 后的串 S 值;

(2) 简述算法 f62 的功能。

```
int f62(HString &S, HString S1, HString S2){
    int i, j;
    S.length = S2.length + S1.length;
    S.ch = (char*)malloc(sizeof(char)*S.length);
    if(!S.ch)
        return 0;
    for(i = 0; i < S1.length; i++)
        S.ch[i] = S1.ch[i];
    for(j = 0; j < S2.length; i++, j++)
        S.ch[i] = S2.ch[j];
    return 1;
}
```

3. (6 分) 已知哈希表的类型定义如下:

```
typedef struct {
```



```

StrKeyType key;
void      *any;
} HElemType;
int hashsize=11;
typedef struct {
    HElemType elem[MAXLEN];
    int      count;
    int      sizeindex;
} HashTable;

```

已知如下表所示的哈希表 HT 的装载因子小于 1，哈希函数 $\text{hash}(\text{key})$ 为关键字首字母在字母表中的序号，处理冲突的方法为线性探测开放定址法， $\text{print}(\text{key})$ 函数打印关键字的值。阅读下列算法，并回答问题：

- (1) 写出执行函数调用 $\text{f63}(\text{HT}, \text{print})$ 的输出结果；
- (2) 简述函数 f63 的功能。

0	1	2	3	4	5	6	7	8	9	10
	MV	NF	ZZ	XP	DC			IE		KK

```

void f63(HashTable ht, void(*print)(StrKeyType)){
    int i, j;
    for (i=0; i<26; i++) {
        j=i%hashsize;
        while (strlen(ht.elem[j].key)!=0) {
            if (hash(ht.elem[j].key)==i)
                print(ht.elem[j].key);
            j=(j+1)%hashsize;
        }
    }
}

```

计算机/软件工程专业

每个学校的

考研真题/复试资料/考研经验

考研资讯/报录比/分数线

免费分享



微信 扫一扫

关注微信公众号

计算机与软件考研