

广东工业大学

2019 年硕士学位研究生招生考试试题

考试科目(代码)名称: (831)数据结构与 C 语言

满分 150 分

(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!)

一、选择题(共 50 分, 每小题 2 分)

1. 设某数据结构的二元组形式表示为 $A=(D,S)$, $D=\{a,b,c,d,e,f\}$, $S=\{\langle a,b\rangle,\langle a,c\rangle,\langle b,d\rangle,\langle b,e\rangle,\langle c,f\rangle\}$, 则数据结构 A 是 ()。
- A. 线性结构 B. 树型结构 C. 集合结构 D. 图型结构
2. 已知在一个单链表中, q 所指结点是 p 所指结点的前驱结点, 若在 q 和 p 之间插入 s 结点, 则执行语句 ()。
- A. $s \rightarrow next = p \rightarrow next; p \rightarrow next = s;$ B. $p \rightarrow next = s \rightarrow next; s \rightarrow next = p;$
C. $q \rightarrow next = s; s \rightarrow next = p;$ D. $p \rightarrow next = s; s \rightarrow next = q;$
3. 设数组 $data[m]$ 作为循环队列 SQ 的存储空间, front 为队头指针, rear 为队尾指针, 则执行出队操作后其头指针 front 值为 ()。
- A. $front = front + 1$ B. $front = (front + 1) \% (m - 1)$
C. $front = (front - 1) \% m$ D. $front = (front + 1) \% m$
4. 如下陈述中正确的是 ()。
- A. 串是一种特殊的线性表 B. 串的长度必须大于零
C. 串中元素只能是字母 D. 空串就是空白串
5. 一个非空广义表的表头 ()。
- A. 不可能是子表 B. 只能是子表
C. 只能是原子 D. 可以是子表或原子
6. 哈夫曼树的带权路径长度 WPL 等于 ()。
- A. 除根以外的所有结点的权值之和 B. 所有结点权值之和
C. 各叶子结点的带权路径长度之和 D. 根结点的值
7. 任何一个无向连通图的最小生成树 ()。
- A. 只有一棵 B. 有一棵或多棵 C. 一定有多棵 D. 可能不存在

8. 二分查找要求被查找的表是 ()。
- A. 键值有序的链表 B. 链表但键值不一定有序
C. 键值有序的顺序表 D. 顺序表但键值不一定有序
9. 若在排序中,从未排序序列中依次取出元素与已排序序列(初始为空)中的元素进行比较,将其放入已排序序列的正确位置,则该方法称为 ()。
- A. 希尔排序 B. 起泡排序
C. 插入排序 D. 选择排序
10. 稀疏矩阵的常用压缩方法有两种,即 ()。
- A. 十字链表和散列表 B. 三元组顺序表和散列表
C. 三元组顺序表和十字链表 D. 二维数组和三维数组
11. 下列数据中,不合法的 C 语言实型数据是 ()。
- A. 0.123 B. 123e3 C. 2.1e3.5 D. 789.0
12. 假定以下表达式中所有变量均已定义为 int 类型并已赋值,试说明其中哪个表达式没有语法错误 ()。
- A. 3.14%2 B. 2a+b; C. m=a<b<c D. n=(1-x)++
13. 已有定义 int a=6; , 则执行语句 a+=a-=a*a; 后, a 的值为 ()。
- A. 36 B. 0 C. -24 D. -60
14. 已有定义: int x=3, y=4, z=5; , 则表达式!(x+y)+z-1&& y+z/2 的值是 ()。
- A. 6 B. 0 C. 2 D. 1
15. 在下面的条件语句中(其中 s1 和 s2 是 C 语言的语句), () 语句在功能上与其他三个语句不等价。
- A. if(a) s1; else s2; B. if(a==0) s2; else s1;
C. if(a!=0) s1; else s2; D. if(a==0) s1; else s2;
16. 对以下循环: for(a=0, b=0; (b!=123)&&(a<=4); a++); 说法正确的是 ()。
- A. 无限循环 B. 循环次数不定 C. 执行 4 次 D. 执行 5 次
17. 下面程序段的输出结果是 ()。
- ```
for(x=9; x>0; x--)
if(x%3!=0)
{
 printf("%d", --x);
}
```

```
continue;
}
```

A. 741

B. 852

C. 963

D. 875421

18.若有数组定义: `char array[]="china";`, 则数组 `array` 所占的空间大小为 ( ) 个字节。

A. 4

B. 5

C. 6

D. 7

19.有定义语句: `char s[10];`若要从终端给 `s` 输入 5 个字符, 错误的输入语句是 ( )。

A. `gets(&s[0]);`

B. `scanf("%s",s+1);`

C. `gets(s);`

D. `scanf("%s",s[1]);`

20.C 语言中, 函数值类型的定义可以缺省, 此时函数值的隐含类型是 ( )。

A. `void`

B. `int`

C. `float`

D. `double`

21. 以下对 C 语言函数的有关描述中, 正确的是 ( )。

A. 在 C 语言中, 调用函数时, 只能把实参的值传给形参, 形参的值不能传给实参

B. C 函数既可以嵌套定义, 也可以嵌套调用

C. 函数必须有返回值, 否则不能使用函数

D. C 程序中有调用关系的所有函数必须放在同一个源程序文件中

22. 下面程序的输出结果是 ( )。

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

```
int f(int a)
```

```
{
```

```
 int b=0;
```

```
 static int c=3;
```

```
 b++;
```

```
 c++;
```

```
 return (a+b+c);
```

```
}
```

```
void main()
```

```
{
```

```
 int a=2, i;
```

```
 for(i=0;i<3;i++)
```

```
 printf("%4d",f(a));
```



}

A. 7 7 7

B. 7 10 13

C. 7 9 11

D. 7 8 9

23. 假定有以下变量定义语句, `float *p,q;`说法正确的是( )。

A. `p` 和 `q` 的类型均为 (`float *`)

B. `p` 的类型为 (`float *`), `q` 的类型为 `float`

C. `p` 的类型为 `float`, `q` 的类型是 (`float *`)

D. 此变量定义语句有语法错误

24. 有以下语句和定义, 对 `s` 数组元素引用形式正确的是 ( )。

`int s[4][5], (*ps)[5];`

`ps=s;`

A. `ps+1`

B. `*(ps+3)`

C. `ps[0][2]`

D. `*(ps+1)+3`

25. 假定有如下定义,

`struct st`

{

`char name[10];`

`int score;`

`} a[3], *p=a;`

以下哪个说法是正确的 ( )。

A. `p->name="Tom";`

B. `*p.name[0]='B';`

C. `(*p)->name[0]='C';`

D. `*( *p).name='D';`

## 二、程序分析题 (3 小题, 8+8+10=26 分)

1、阅读以下程序回答问题。(8 分)

`#define SIZE 20`

`#include <stdio.h>`

`void f11(int a[], int n)`

{ `int j,x;`

`for (j=1;j<=n;j++)`

`{ scanf("%d",&x);`

`a[x]++;`

```
}
}
void main()
{int n,j,sum,a[SIZE]={0};
scanf("%d",&n);
f11(a,n);
sum=j=0;
while (sum<n/2+1)
{ j++;
sum+=a[j];
}
printf("%d",j);
}
```

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

(1) 假设输入的数据在 1~16 之间, 函数 f11 的功能是什么? (4 分)

(2) 假设有如下输入, 则程序运行结果是什么? (4 分)

11 4 5 6 4 3 3 2 3 2 1 6

2、阅读以下程序回答问题。(8 分)

```
#include <stdio.h>
void f21(int a[], int n)
{ int temp,i,j;
for (i=n-1;i>=1;i--)
for (j=0;j<i;j++)
if (a[j]>a[j+1])
{ temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
```

```
void main()
{ int a[2019];
 int n,i;
 scanf("%d",&n);
 for (i=0;i<n;i++)
 scanf("%d",&a[i]);
 f21(a,n);
 for (i=0;i<n;i++)
 printf("%d",a[i]);
}
```

(1) 函数 f21 的功能是什么? (4 分)

(2) 假设有如下输入, 则程序运行结果是什么? (4 分)

7 9 4 6 5 2 0 1

3、阅读以下程序回答问题。(10 分)

```
#include <stdio.h>
#include <math.h>
int f31(int x)
{ int r=1,i;
 int k=sqrt(x);
 for (i=2;i<=sqrt(x);i++)
 if(x%i==0) break;
 if(i<=k) r=0;
 return r;
}
int f32(int y[][5])
{ int i,j,sum=0;
 for (i=0;i<2;i++)
 for (j=0;j<5;j++)
 if (f31(y[i][j])) sum+=y[i][j];
}
```

```
return sum;
}
main ()
{int a[][5]={12,9,4,25,36,3,7,15,26,81};

printf("%d",f32(a));
}
```

- (1)函数 f31 的功能是什么? (3 分) (2)函数 f32 的功能是什么? (3 分)  
(3)程序运行输出结果是什么? (4 分)

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

1. 在屏幕上输出一个矩阵形状的“12×12 乘法口诀表”, 除表头外有 12 行。

```
#include <stdio.h>
main ()
{ inti,j;
 for(i=1;i<=12;i++)
printf("%4d",i);
printf("\n-----\n");
 for(i=1;__(1)__;i++)
 {for(j=1;j<=12;j++)
printf("%4d",__(2)__);
printf("__(3)");
 }
}
```

2. 输入若干学生的学号和 3 门课程的成绩并输出。

```
#include <stdio.h>
#include <malloc.h>
struct student
{ int num;
 float score[3];
};
void main()
{int i=0,n;
 float a[3];
 __ (1) __ *ptr[200];
 printf ("请输入学生学号和 3 门课成绩，学号为 0 表示数据输入结束\n");
 do
 { ptr[i]=(struct student*)malloc(sizeof(struct student));
 scanf ("%d%f %f %f",&ptr[i]->num,&a[0],&a[1],&a[2]);
 ptr [i]->score[0]=a[0];
 ptr [i]->score[1]=a[1];
 ptr [i]->score[2]=a[2];
 if (__ (2) __) break;
 i++;
 }while(1);
 n=i-1;
 free (__ (3) __) ;
 for (i=0;i<=n;i++)
 printf ("%d%.1f %.1f %.1f\n",ptr[i]->num,ptr[i]->score[0],ptr[i]->score[1],ptr[i]->score[2]);
 }
```

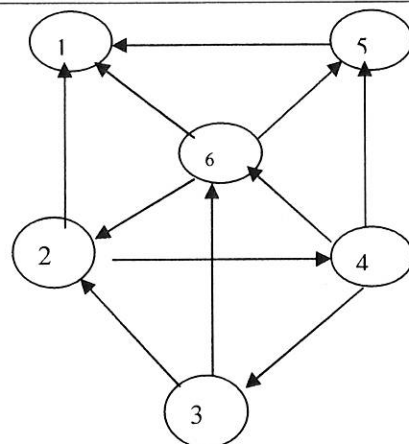


#### 四. 解答题 (共 20 分)

1. (共 10) 已知如图所示的有向图, 请给出该图:

(1) 每个顶点的入/出度; (共 5 分)

(2) 邻接矩阵; (共 5 分)



2. (10 分) 对序列 (3, 87, 12, 61, 70, 97, 26, 45) 执行升序排序。试根据堆排序原理, 填写完整下列各步骤结果。

建立大顶堆结构: \_\_\_\_\_

交换与调整:

(1) 87, 70, 26, 61, 45, 12, 3, 97; (2) \_\_\_\_\_;

(3) 61, 45, 26, 3, 12, 70, 87, 97; (4) 45, 12, 26, 3, 61, 70, 87, 97;

(5) 26, 12, 3, 45, 61, 70, 87, 97; (6) \_\_\_\_\_;

(7) 3, 12, 26, 45, 61, 70, 87, 97;

#### 五. 算法填空题 (共 18 分, 每空 3 分)

1. 完成下列算法, 实现顺序表的就地逆置, 即利用原表的存储空间将线性表(a1,a2,...,an)逆置为(an,an-1,...,a1)。

顺序表类型定义如下:

```
typedef struct {
```

```
 ElemType *elem;
```

```
 int length;
```

```
 int size;
```

```
 int increment;
```

```
} SqList;
```

```
void Inverse(SqList &L) {
```

```

int i,n,x;

ElemType ch;

n=L.length-1;

x = (n+1)/2 ;

for (i=0; _____; i++) {

 ch=L.elem[i];

 L.elem[i]= _____;

 L.elem[n-i]= _____;

}

}

```

2. 完成下列算法，实现链队列的求队列长度操作。

链队列的类型定义为：

```

typedef struct LQNode {

 ElemType data;

 struct LQNode *next;

} LQNode, *QueuePtr; // 结点和结点指针类型

typedef struct {

 QueuePtr front; // 队头指针

 QueuePtr rear; // 队尾指针

} LQueue; // 链队列类型

```

```

int QueueLength(LQueue Q)

{

 int i=0;

 LQNode *p;

 p = Q.front;

 while(_____) { _____; i++; }

 return _____;

}

```

## 六. 算法分析题 (共 18 分, 每题 9 分)

阅读下面的算法, 简述算法的功能, 并分析其时间性能

1. 以孩子兄弟链表表示的树。

孩子兄弟链表类型定义:

```
typedef struct CSTNode {
 TElemType data;
 struct CSTNode *firstChild, *nextSibling;
} CSTNode, *CSTree;
```

```
int f1(CSTree T)
{
 int d, ds, dt;
 CSTree p;
 if (!T) return 0;
 else {
 ds=0; dt=0;
 for (p=T->firstChild; p!=NULL; p=p->nextSibling) {
 dt++;
 d=f1(p);
 if (d>ds) ds=d;
 }
 return ds>dt ? ds : dt;
 }
}
```

2. 以邻接表方式存储的有向图  $G$ 。

图的邻接表存储结构的类型定义如下:

```
#define UNVISITED 0
#define VISITED 1
```

```

#define INFINITY MAXINT // 计算机允许的整数最大值，即∞
typedef char VexType;
typedef enum {DG,DN,UDG,UDN} GraphKind; // 有向图,有向网,无向图,无向网
typedef struct AdjVexNode {
 int adjvex; // 邻接顶点在顶点数组中的位序
 struct AdjVexNode *next; // 指向下一个邻接顶点（下一条边或弧）
 int info; // 存储边（弧）相关信息，对于非带权图可不用
} AdjVexNode, *AdjVexNodeP; // 邻接链表的结点类型
typedef struct VexNode {
 VexType data; // 顶点值，VexType 是顶点类型，由用户定义
 struct AdjVexNode *firstArc; // 邻接链表的头指针
} VexNode; // 顶点数组的元素类型
typedef struct {
 VexNode *vexs; // 顶点数组，用于存储顶点信息
 int n, e; // 顶点数和边（弧）数
 GraphKind kind; // 图的类型
 int *tags; // 标志数组
} ALGraph; // 邻接表类型

int f2(ALGraph G, int k)
{
 int odeg; odeg = 0;
 AdjVexNodeP p;
 if(k<0 || k>=G.n) return -1;
 p = G.vexs[k].firstArc;
 while(p!=NULL) {
 p = p->next; odeg++;
 }
 return odeg;
}

```