

# 北京航空航天大学 2006 年 硕士研究生入学考试试题

科目代码: 891

## 数据结构与 C 语言程序设计 (共 7 页)

考生注意: 所有答题务必书写在考场提供的答题纸上, 写在本试题单上的答题一律无效 (本题单不参与阅卷)。

### 一、填空题 (本题共 20 分, 每小题各 2 分)

1. 在长度为  $n$  的线性链表的已知位置插入一个新的链结点的操作的时间复杂度采用大  $O$  形式表示为\_\_\_\_\_。
2. 若频繁地对线性表进行插入或删除操作, 该线性表应采用\_\_\_\_\_存储结构。
3. 若某堆栈初始为空, PUSH 与 POP 分别表示对堆栈进行一次进栈与出栈操作, 那么, 对于输入序列  $a, b, c, d, e$ , 经过 PUSH, PUSH, POP, PUSH, POP, PUSH, PUSH 以后, 输出序列应该是\_\_\_\_\_。
4. 具有  $n$  个结点的非空满二叉树有\_\_\_\_\_个叶结点。
5. 若非空二叉树采用顺序存储结构依次将结点的数据信息存放于一个一维数组中 (假设数组的第一个元素的下标为 1), 下标分别为  $i$  和  $j$  的两个结点处在树中同一层的条件是\_\_\_\_\_。 ( $i \neq j \neq 1$ )
6. 若一棵度为 7 的树中有 8 个度为 1 的结点, 有 7 个度为 2 的结点, 有 6 个度为 3 的结点, 有 5 个度为 4 的结点, 有 4 个度为 5 的结点, 有 3 个度为 6 的结点, 有 2 个度为 7 的结点, 则该树一共有\_\_\_\_\_个结点。
7. 在一个按值有序排列的顺序表中进行折半查找, 其查找过程可以用一棵称之为“判定树”的二叉树来描述。若顺序表的长度为 19, 则对应的“判定树”的根结点的左孩子之值 (即元素在表中的位置) 是\_\_\_\_\_。
8. 若一个待散列的线性表为  $(18, 25, 63, 50, 42, 32, 9, 45)$ , 散列函数为  $H(k) = k \text{ MOD } 9$ , 则与 18 发生散列冲突的元素有\_\_\_\_\_个。



9. 插入排序法的时间花费主要取决于排序过程中元素之间的比较次数。若具有  $n$  个元素的序列初始时是一个按值递减的序列, 则一共要进行\_\_\_\_\_次这种比较。

10. 按照大顶堆积的定义, 对序列(26,5,77,1,61,11,59,15,48,19)进行堆积排序, 第二趟排序结束时序列的状态是\_\_\_\_\_。

## 二、简答题 (本题共 20 分, 每小题各 10 分)

1. 从源点到带权连通图中其他各顶点的最短路径可以构成一棵生成树, 该生成树是否是该图的最小生成树? 为什么?

2. 在散列函数与散列地址范围都分别相同的前提下, 为什么说采用链地址法处理冲突比采用开放地址法处理冲突的时间效率高?

## 三、问题求解题 (本题共 10 分, 每小题各 5 分)

对于长度为 12 的线性表 (Dec, Feb, Nov, Oct, Jul, Sept, Aug, Apr, May, Jun, Jan, Mar),

1. 请按线性表中元素在表中的先后顺序构造出一棵二叉排序树;

2. 若每个数据元素的查找概率均等, 请写出查找此二叉排序树中任意一个结点的平均查找长度 ASL。

## 四、算法设计题 (本题 15 分)

已知二叉树采用二叉链表存储结构, 链结点构造为

lchild	data	rchild
--------	------	--------

根结点指针为 T。请写出复制该二叉树的递归算法。

## 五、算法设计题 (本题 10 分)

已知一具有  $n$  个顶点、 $e$  条边的有向图采用邻接矩阵存储, 邻接矩阵以三元组表  $A[0..e][0..2]$  形式给出。请写一算法, 分别计算图中各顶点的度。

## 六、单项选择题 (本题共 20 分, 每小题各 1 分)

1. “||”、“!=”、“<=” 和 “+” 这四种运算符中, 优先级最低的运算符是\_\_\_\_\_。

A. ||

B. !=

C. <=

D. +



2. sizeof(double)是一个\_\_\_\_\_表达式。
- A. 双精度型      B. 整型      C. 不合法      D. 函数调用
3. 对于条件表达式(E)?(a++):(a--), 其中的表达式 E 等价于\_\_\_\_\_。
- A. E==1      B. E==0      C. E!=1      D. E!=0
4. 下面给出的定义中, 合法的数组定义是\_\_\_\_\_。
- A. int A[]="string";      B. char A[]={0,1,2,3,4,5};  
C. char A="string";      D. int A[5]={0,1,2,3,4,5};
5. 定义 "int (\*p)[n];" 的含义是\_\_\_\_\_。
- A. p 为指针类型  
B. p 为指针数组, 每个数组元素都是指针  
C. p 为指针数组, 每个数组元素都指向整型数据  
D. p 为数组指针, 它是指向含有 n 个元素的一维数组的指针变量
6. 若有说明 "double \*p,a;", 则能够通过 scanf 语句正确给输入项读入数据的程序段是\_\_\_\_\_。
- A. \*p=&a; scanf("%lf",p);      B. p=(double)malloc(8); scanf("%lf",p);  
C. p=&a; scanf("%lf",a);      D. p=&a; scanf("%le",p);
7. printf 的格式转换字符 "%o" 的含义是\_\_\_\_\_。
- A. 以八进制形式输出有符号的整数, 不带有前缀 0  
B. 以八进制形式输出有符号的整数, 带有前缀 0  
C. 以八进制形式输出无符号的整数, 不带有前缀 0  
D. 以八进制形式输出无符号的整数, 带有前缀 0
8. 语句 while(!e); 中的条件 !e 等价于\_\_\_\_\_。
- A. e!=0      B. e==0      C. e!=1      D. ~e
9. 判断两个字符串 s1 和 s2 是否相等应该使用\_\_\_\_\_。
- A. if(s1==s2)      B. if(s1=s2)      C. if(strcmp(s1,s2))      D. if(strcmp(s1,s2)==0)
10. 对于函数调用语句 "FUN(a+b,(x,y),FUN(n+k,d,(a,b)));", 语句中的实参个数是\_\_\_\_\_。



- A. 6                      B. 5                      C. 4                      D. 3

11. 函数 fopen() 的 mode 取值 “w+” 和 “a+” 时都可以写入数据，它们之间的差别是\_\_\_\_\_。

- A. 取 “w+” 时可以在中间插入数据，而取 “a+” 时只能在末尾追加数据  
B. 取 “w+” 和取 “a+” 时都只能在末尾追加数据  
C. 文件存在时，取 “w+” 时清除原文件数据，取 “a+” 时保留原文件数据  
D. 取 “w+” 时不能在中间插入数据，而取 “a+” 时只能在末尾追加数据

12. 在下列程序中，循环体一共被执行了\_\_\_\_\_次。

```
#include <stdio.h>
main()
{   int i=0,j=10,k=2,sum=0;
    for(;;){
        i+=k;
        if(i>j){
            printf("%d",sum);
            break;
        }
        sum+=j;
    }
}
```

- A. 5                      B. 6                      C. 7                      D. 8

13. 在下列程序段中，与其它程序段执行结果不同的是\_\_\_\_\_。

A. 

```
int k=1,s=0;
while(1){
    s+=k;k++;
    if(k>100)
        break;
}
printf("%d\n",s);
```

B. 

```
int k=1,s=0;
loop:
    s+=k;
    if(++k<=100)
        goto loop;
printf("%d\n",s);
```

C. 

```
int k,s=0;
for(k=1;k<=100;s+=++k);
printf("%d\n",s);
```

D. 

```
int k=1,s=0;
do s+=k;
while(++k<=100);
printf("%d\n",s);
```



14. 若要求函数的功能是交换  $x$  和  $y$  中的值，并且通过正确调用返回交换结果。下面给出的四个函数中，能够正确执行此功能的函数是\_\_\_\_\_。

A. FUN(int \*x,int \*y)  
{  
  int \*temp;  
  \*temp=\*x;  
  \*x=\*y;  
  \*y=\*temp;  
}

B. FUN(int x,int y)  
{  
  int temp;  
  temp=x;  
  x=y;  
  y=temp;  
}

C. FUN(int \*x,int \*y)  
{  
  \*x=\*y;  
  \*y=\*x;  
}

D. FUN(int \*x,int \*y)  
{  
  \*x=\*x-\*y;  
}

15. 若定义如下变量和数组：

int i;  
int A[3][3]={1,2,3,4,5,6,7,8,9};  
则下面语句的输出结果是\_\_\_\_\_。

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

  printf("%d ",A[i][2-i]);

A. 1 4 7

B. 1 5 9

C. 3 5 7

D. 3 6 9

16. 执行下列程序段的输出结果是\_\_\_\_\_。

#include <stdio.h>

main()

{  
  int a=02,b=3;

  printf("a=%02d,b=%02d",a,b);  
}

A. a=2,b=3

B. a=%02,b=%03

C. a=%02d,b=%02d

D. a=%d,b=%d

17. 以下程序的输出结果是\_\_\_\_\_。

#define MIN(x,y) (x)<(y)?(x):(y)

main()

{  
  int a,b,c;

  a=10;

  b=15;

  c=10\*MIN(a,b);

  printf("%d",c);  
}

A. 15

B. 100

C. 10

D. 150



18. 以下程序的输出结果是\_\_\_\_\_。

```
main()
{   int k=2,m=4,n=6;
    int *pk=&k,*pm=&m,*p;
    *(p=&n)=*pk*(*pm);
    printf("%d\n",n);
}
```

A. 10                      B. 8                      C. 6                      D. 4

19. 以下程序的输出结果是\_\_\_\_\_。

```
main()
{   int S[]={2,4,6},*p=&S[0],a=8,b,c;
    for(b=0;b<3;b++)
        c=*(p+b)<a)?*(p+b):a;
    printf("%d\n",c);
}
```

A. 2                      B. 4                      C. 6                      D. 8

20. 下列程序向文件输出的结果是\_\_\_\_\_。

```
#include <stdio.h>
main()
{   FILE *fp=fopen("test","wb");
    fprintf(fp, "%d%5.0f%c%d", 42, 75873.0, '-', 3353);
}
```

A. 42 75873-3353                      B. 4275873.000000-3353  
C. 4275873-3353                      D. 由于文件为二进制文件而不可读

### 七、简答题（本题共 20 分，每小题各 10 分）

1. 什么是递归程序？相对于非递归程序，递归程序有什么优点和不足？
2. 在 C 语言中，函数通常分为哪两类？这两类函数是如何被使用的？

### 八、（本题 15 分）

请编写一 C 程序，该程序求出(输出)1~1000 以内的所有同构数。所谓同构数是指对于一个 N 位的自然数，其数的平方值的末尾的 N 位数等于该数。例如，1 位数 5 就是一个同构数，因为 5 的平方为 25，而 25 的末尾的 1 位数为 5。2 位数

25 也是一个同构数，因为 625 的末尾的 2 位数为 25。

#### 九、(本题 20 分)

编写一个 C 程序 `fcac.c`，将命令行中指定的多个文本文件连接成为一个文本文件。例如：

```
fcac file1 file2 file3
```

其中，`fcac` 为命令名，`file1`、`file2` 和 `file3` 为参数。该命令的功能是将文本文件 `file1`、`file2` 和 `file3` 连接成为一个文本文件(文件名由第一个参数 `file1` 指定)。