

## 2010 年硕士研究生入学考试

### 数据结构与 C 语言程序设计(993) 试题与答案

#### 一、单项选择题 (本题共 20 分, 每小题各 2 分)

1. 已知双向循环链表的结点构造为 

llink	data	rlink
-------	------	-------

, 在链表中由指针 q 所指结点的后面插入指针为 p 的结点的过程是依次执行\_\_\_\_\_。
- A.  $p \rightarrow \text{llink} = q; p \rightarrow \text{rlink} = q \rightarrow \text{rlink}; q \rightarrow \text{rlink} = p; q \rightarrow \text{llink} = p;$   
B.  $p \rightarrow \text{llink} = q; p \rightarrow \text{rlink} = q \rightarrow \text{rlink}; q \rightarrow \text{rlink} = p; q \rightarrow \text{rlink} \rightarrow \text{llink} = p;$   
C.  $p \rightarrow \text{llink} = q; p \rightarrow \text{rlink} = q \rightarrow \text{rlink}; q \rightarrow \text{rlink} = p; p \rightarrow \text{rlink} \rightarrow \text{llink} = p;$   
D.  $p \rightarrow \text{llink} = q; p \rightarrow \text{rlink} = q \rightarrow \text{rlink}; q \rightarrow \text{rlink} = p; p \rightarrow \text{llink} \rightarrow \text{llink} = p;$
2. 对于采用链式存储结构的队列, 在进行删除操作时\_\_\_\_\_。
- A. 只需修改队头指针  
B. 只需修改队尾指针  
C. 队头指针和队尾指针都需要修改  
D. 队头指针和队尾指针都可能需要修改
3. 将中缀表达式转换为等价的后缀表达式的过程中要利用堆栈保存运算符。对于中缀表达式  $A - (B + C/D) \times E$ , 当扫描读到操作数 E 时, 堆栈中保存的运算符依次是\_\_\_\_\_。
- A.  $- \times$   
B.  $-(\times$   
C.  $- +$   
D.  $-(+$
4. 若完全二叉树的第 7 层有 10 个叶结点, 那么, 该二叉树结点数目最大是\_\_\_\_\_。
- A. 73  
B. 74  
C. 234  
D. 235
5. 对于具有 k 条边的有向图, 其对应的邻接表中边结点的数目为\_\_\_\_\_。
- A.  $k-1$   
B. k  
C.  $2k$   
D.  $k^2$
6. 通过拓扑排序能够得到拓扑序列的图一定是\_\_\_\_\_。
- A. 连通图  
B. 带权连通图  
C. 无回路的图  
D. 无回路的有向图
7. 在具有 100 个元素、且元素按值有序排列的一维数组中进行折半查找, 最大比较次数为\_\_\_\_\_。
- A. 7  
B. 10  
C. 25  
D. 50
8. 评价一个散列函数的质量优劣的主要标准是\_\_\_\_\_。
- A. 函数的形式是否简单  
B. 函数的计算时间的多少  
C. 函数值的分布是否均匀  
D. 函数是否是解析式
9. 每一趟排序都从序列中未排好序的元素中挑出一个元素, 然后将其依次放入已经排好序序列的一端的排序方法是\_\_\_\_\_。
- A. 快速排序法  
B. 二路归并排序法  
C. 折半插入排序法  
D. 选择排序法
10. 对具有 n 个元素的序列采用堆积排序法排序, 排序的总趟数为\_\_\_\_\_。
- A.  $n-1$   
B. n  
C.  $n+1$   
D.  $2n$

#### 二、简答题 (本题共 20 分, 每小题各 5 分)

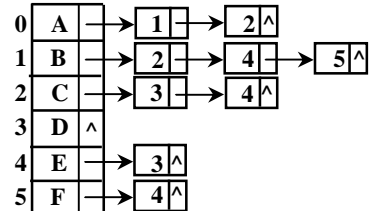
1. 若已知在长度为 n 的顺序表  $(a_1, a_2, \dots, a_n)$  的第 i 个位置  $(1 \leq i \leq n+1)$  插入一个新的数据元素的概率为  $p_i = \frac{2(n-i+1)}{n(n+1)}$ , 则平均插入一个元素时所需要移动元素次数的期望值(平均次数)是多少?
2. 什么是递归算法? 递归算法在执行时, 通常需要借助何种数据结构来完成?

3. 对一个图进行遍历可以得到不同的遍历序列,那么,导致得到的遍历序列不惟一的因素有哪些?

4. 若某地区有 10000 名学生参加数学竞赛,只录取成绩优异的前 10 名,并将他们按成绩从高分到低分依次输出,而对落选的其他参赛者不需排出名次。请问:在这种情况下,对于选择排序法、快速排序法和堆积排序法三种排序方法,应该采用其中哪一种?为什么?

### 三、综合题 (本题共 20 分, 每小题各 5 分)

1. 已知某有向图的邻接表如题三、1 图所示,请分别写出该有向图所有可能的拓扑序列。



题三、1 图

2. 已知二叉树的中序遍历序列为 C,A,D,F,B,E,按层次遍历序列为 A,C,B,D,E,F,请画出该二叉树。

3. 已知散列函数为  $H(k)=k \text{ MOD } 7$ ,并采用线性探测再散列法处理冲突,请画出在下列散列表中依次插入关键字 17, 27 以后的表的状态。

0	1	2	3	4	5	6
	15		10	45		20

4. 请写出下列递归算法的功能。

```
int ALGORITHM(int A[ ],int n)
{
    int m;
    if(n==1)
        m=A[0];
    else if(A[n-1] > ALGORITHM (A,n-1))
        m=A[n-1];
    else
        m=ALGORITHM (A,n-1);
    return m;
}
```

### 四、算法设计题 (本题 15 分)

已知非空二叉树采用二叉链表结构,链结点构造为 

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

,根结点指针为 T。请利用二叉树遍历的非递归算法写出求二叉树中由指针 q 所指结点(设 q 所指结点不是二叉树的根结点)的兄弟结点的算法。若二叉树中存在该兄弟结点,算法给出该兄弟结点的位置,否则,算法给出 NULL。

要求:写算法之前先用文字简要给出算法的核心思想。

### 五、单项选择题 (本题共 20 分, 每小题各 1 分)

1. 在 C 语言中,要求参加运算的操作数必须是整数的运算符是\_\_\_\_\_。

- A. !                      B. %                      C. /                      D. >

2. 若变量 x 为整型,变量 y 为实型,变量 i 为双精度型,则表达式  $20+'x'+i*y$  的值的类型为\_\_\_\_\_。

- A. int                      B. float                      C. double                      D. 不确定

3. 对于关系  $a \leq b \leq c$ ,对应的 C 语言表达式应该是\_\_\_\_\_。

- A.  $(a < b) \&\& (b <= c)$                       B.  $(a < b) \text{AND} (b <= c)$   
C.  $(a < b <= c)$                       D.  $(a < b) \& (b <= c)$

4. 以下关于输入的叙述中,正确的是\_\_\_\_\_。

- A. 只有格式控制, 没有输入项, 也能进行正确的输入, 如 `scanf("x=%d,y=%d");`  
B. 当输入实型数据时, 格式控制部分应规定小数点后的位数, 如 `scanf("%4.2f",&f);`  
C. 当输入数据时, 必须指明变量的地址, 如 `scanf("%f",&f);`  
D. 输入项可以是一个实型常量, 如 `scanf("%f",3.14);`

5. 对于以下程序段:

```
int k=0,s;  
do{  
    scanf("%d",&s);  
    k++;  
}while(s!=100 && k<3);
```

此处的 do-while 循环的结束条件是\_\_\_\_\_。

- A. s 的值不等于 100 并且 k 的值小于 3      B. s 的值等于 100 并且 k 的值大于等于 3  
C. s 的值不等于 100 或者 k 的值小于 3      D. s 的值等于 100 或者 k 的值大于等于 3

6. 执行语句 `for(j=1;j++<4;);` 后变量 j 的值是\_\_\_\_\_。

- A. 5      B. 4      C. 3      D. 不确定

7. 以下叙述中, 正确的是\_\_\_\_\_。

- A. `continue` 语句的作用是结束整个循环的执行  
B. 只能在循环体内和 `switch` 语句体内使用 `break` 语句  
C. 在循环体内使用 `break` 语句或 `continue` 语句的作用相同  
D. 从多层循环嵌套中退出时, 只能使用 `goto` 语句

8. 在 C 语言中, 数组名代表\_\_\_\_\_。

- A. 数组全部元素的值      B. 数组首地址  
C. 数组第 1 个元素的值      D. 数组元素的个数

9. 若有定义: `int a[10];`, 则对数组 a 元素的正确引用的是\_\_\_\_\_。

- A. `a[10]`      B. `a[3.5]`      C. `a(5)`      D. `a[10-10]`

10. 若已有定义 `char str1[8],str2[ ]={"123456"};`和 `int k;`, 要将字符串“123456”赋给 str1, 则下面的语句中, 错误的是\_\_\_\_\_。

- A. `strcpy(str1,str2);`      B. `strcpy(str1, "123456");`  
C. `str1="123456";`      D. `for(k=0;k<7;k++) str1[k]=str2[k];`

11. 判断字符串 str1 是否大于字符串 str2, 应该使用\_\_\_\_\_。

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

12. `main` 函数的正确说明形式是\_\_\_\_\_。

- A. `main(int argc,char *argv)`      B. `main(int abc,char **abv)`  
C. `main(int argc,char argv)`      D. `main(int c,char v[ ])`

13. C 语言规定, 简单变量做实参时, 它和对应的形参之间的数据传递方式是\_\_\_\_\_。

- A. 单向值传递      B. 地址传递  
C. 由实参传给形参, 再由形参传回给实参      D. 由用户指定传递方式

14. C 语言规定, 函数返回值的类型是由\_\_\_\_\_。

- A. `return` 语句中的表达式的类型所决定  
B. 调用该函数时的主函数的类型所决定

- C. 调用该函数时系统临时决定  
D. 在定义该函数时所指定的函数的类型所决定

15. 下面给出的 4 个定义语句中, 与 `int *p[5];` 等价的是\_\_\_\_\_。

- A. `int p[5];`                      B. `int *p;`                      C. `int *(p[5]);`                      D. `int (*p)[5];`

16. 若有以下定义和语句, 则值为 1002 的表达式是\_\_\_\_\_。

```
struct s{
    int age;
    int num;
};
static struct s a[3]={1001,20,1002,19,1003,21},*ptr;
ptr=a;
```

- A. `ptr++->num`    B. `(ptr++)->age`  
C. `(*ptr).num`    D. `(*++ptr).age`

17. 若要通过下面的程序段使得指针变量指向一个存储整型变量的动态存储单元, 则程序段中的空白处(横线上方)应该是\_\_\_\_\_。

```
int *ptr;
ptr=_____malloc(sizeof(int));
```

- A. `int`                      B. `int *`                      C. `(int *)`                      D. `(*int)`

18. 下面关于宏的叙述中, 错误的是\_\_\_\_\_。

- A. 宏名无类型, 其参数也无类型  
B. 宏定义不是 C 语句, 不必在行的末尾加分号  
C. 宏替换只是字符替换  
D. 宏定义命令必须写在文件的开头

19. 下列关于 C 语言文件操作的叙述中, 正确的是\_\_\_\_\_。

- A. 对文件的操作必须是先打开文件  
B. 对文件的操作必须是先关闭文件  
C. 对文件操作之前必须先测试文件是否已打开, 然后再打开文件  
D. 对文件的操作无顺序要求

20. 使用 `fopen()` 函数以文本方式打开或者建立可读写文件。要求: 若指定文件不存在, 则建立一个文件, 并使文件指针指向其开头; 若指定文件存在, 则打开该文件, 并将文件指针指向其结尾。下列“文件使用方式”中, 正确的是\_\_\_\_\_。

- A. `“r+”`                      B. `“a+”`                      C. `“w+”`                      D. `“a”`

## 六、填空题 (本题共 20 分, 每小题各 5 分)

1. 下列程序的运行结果是\_\_\_\_\_。

```
main()
{
    int a,b,c;
    a=1; b=2; c=3;
    a=b--<=a||a+b!=c;
    printf(“%d,%d”,a,b);
}
```

2. 下列程序的运行结果是\_\_\_\_\_。

```
#include <stdio.h>
f(char *s)
{
    char *p=s;
```

```

        while(*p)
            p++;
        return (p-s);
    }

    main( )
    {   char *a="abded";
        int k;
        k=f(a);
        printf("%d",k);
    }

```

3. 下列程序的运行结果是\_\_\_\_\_。

```

#include <stdio.h>
#define N 5
#define M N+1
#define f(x) (x*M)
main( )
{   int i,j;
    i=f(2);
    j=f(1+1);
    printf("%d %d\n",i,j);
}

```

4. 下列程序的运行结果是\_\_\_\_\_。

```

#include <stdio.h>
void main( )
{   FILE *fp;
    int d1,d2,a[6]={1,2,3,4,5,6};
    fp=fopen("file.dat","w");
    fprintf(fp,"%d %d %d\n",a[0],a[1],a[2]);
    fprintf(fp,"%d %d %d\n",a[3],a[4],a[5]);
    fclose(fp);
    fp=fopen("file.dat","r");
    fscanf(fp,"%d %d",&d1,&d2);
    printf("%d %d\n",d1,d2);
    fclose(fp);
}

```

## 七、程序设计题（本题 20 分）

请编写程序，该程序首先通过键盘输入获得整型数据  $a$  与  $n$ ，然后计算  $\text{sum}=a+aa+aaa+\cdots$  (共  $n$  项)，最后输出计算结果。例如：当  $a=5$ ， $n=4$  时，计算  $\text{sum}=5+55+555+5555$ 。

## 八、程序设计题（本题 15 分）

在 Unix 操作系统中有一条命令，命令的功能是打印文本文件的最后  $n$  行。命令格式为：

`tail [-n] filename`

其中，`tail` 为命令名；参数 `filename` 为文本文件名；参数 `[-n]` 表示要打印的行数，该参数是可选的，缺省值为 10，即无此参数时，表示打印文件的最后 10 行。例如，命令

`tail -20 example.txt`

表示打印文本文件 `example.txt` 的最后 20 行。如果被打印的文本文件中行数少于  $n$  行或者少于 10 行，该命令将打印文件中的所有行。

请用带参数的 `main` 函数实现该程序。该程序应该具有一定的错误处理能力，例如，能够处理非法

命令参数和非法文件名。

程序中使用以下 C 库函数：

- `int atoi(char *s)`——将数字串转换为相应整数；
- `fgets(char *s, int n, FILE *fp)`——从文件中读入一行；
- `void *malloc(unsigned size), free`——申请和释放内存；
- `strlen`——计算字符串的长度；
- `strcpy`——将一个字符串拷贝到另一个字符串中。

除此之外，不允许使用其它库函数。

提示：

1. 可以在命令行参数正确性分析过程中获取被打印的文本文件名称以及需要打印的行数等信息。
2. 如果命令行分析正确，可以建立一个不带头结点的单向循环链表存放从文件中读到的内容。

## 参考答案：

### 一、单项选择题

1. C
2. D
3. A
4. D
5. B
6. D
7. A
8. C
9. D
10. A

### 二、简答题

1. 答：
$$\sum_{i=1}^{n+1} pi(n-i+1) = \frac{2}{n(n+1)} \sum_{i=1}^{n+1} (n-i+1)^2 = \frac{2}{n(n+1)} \frac{n(n+1)(2n+1)}{6} = \frac{2n+1}{3}$$

2. 答：一个算法在结束本算法之前，直接或者间接地调用算法自身，这样的算法称为递归算法。递归算法在执行中通常需要借助于堆栈这种数据结构来完成。

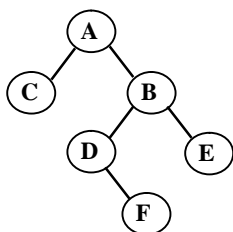
3. 答：导致得到的遍历序列不惟一的原因主要有：开始遍历的顶点不同、采用的遍历方法不同，图的存储结构不同(即邻接表中边结点的链接次序不同)。

4. 答：对于具有  $n$  个元素的序列，选择排序法虽然每一趟排序可以选出一个最大（或最小）元素，并加入到已有有序子序列中，但需要进行  $n-1$  次元素之间的比较；选出次最大（或次最小）元素需要再比较  $n-2$  次，可以推定，该排序方法的时间复杂度为  $O(n^2)$ ，故不能采用该方法。快速排序法虽然有更好的时间效率，但需要等到最后才能确定各元素的位置，故此方法也不适合采用。只有堆积排序法在未结束全部排序之前可以得到部分排序结果，根据堆积排序法的基本原理可知，建立初始堆积过程中元素之间的比较次数最多不超过  $4n$  次；若要在  $n$  个元素选出  $k$  个元素，则对于深度为  $k$  的堆积，堆积调整过程中进行的元素之间的比较次数最多为  $2(k-1)$  次，且辅助空间为  $O(1)$ 。综上所述，此题应该采用堆积排序法。

### 三、综合题

1. 拓扑序列： ABCFED      ABFCED

2.



3.

0	1	2	3	4	5	6
27	15		10	45	17	20

4. 求一个整型数组中最大值元素。

#### 四、算法设计题

**算法核心思想：**利用二叉树前序遍历的非递归算法解决该问题。在遍历过程中，当访问一个结点时，判断该结点的左孩子或者右孩子是否是 q 指结点，若是，返回该结点的左孩子或者右孩子的位置即可。

算法：

```
BTREE FINDBROTHER(BTREE T,BTREE q)
{
    /* T 为二叉树根结点所在链结点的地址 */
    BTREE STACK[M],p=T;
    int top=-1;
    do{
        while(p!=NULL){
            if(p->lchild==q);
                return p->rchild;
            if(p->rchild==q)
                return p->lchild;           /* 访问当前 p 指的结点 */
            STACK[++top]=p;                 /* 当前 p 指结点的地址进栈 */
            p=p->lchild;                     /* 将 p 移到其左孩子结点 */
        }
        p=STACK[top--];                     /* 退栈 */
        p=p->rchild;                         /* 将 p 移到其右孩子结点 */
    }while(p!=NULL||top!=-1);
}
```

#### 五、单项选择题

- |       |       |       |       |       |
|-------|-------|-------|-------|-------|
| 1. B  | 2. C  | 3. A  | 4. C  | 5. D  |
| 6. A  | 7. B  | 8. B  | 9. D  | 10. C |
| 11. C | 12. B | 13. A | 14. D | 15. C |
| 16. D | 17. C | 18. D | 19. A | 20. B |

#### 六、填空题

- |           |         |            |              |
|-----------|---------|------------|--------------|
| 1. 答案：1,1 | 2. 答案：5 | 3. 答案：11 7 | 4. 答案：123456 |
|-----------|---------|------------|--------------|

#### 七、程序设计题

```
#include <stdio.h>
void main( )
{
    int n,count=1;
    long a,sum=0,temp=0;
    printf("\nInput a and n:");
    scanf("%ld,%d",&a,&n);
    while(count<=n){
        temp=temp+a;
        sum=sum+temp;
        a=a*10;
        count++;
    }
    printf("%ld\n",sum);
}
```

#### 八、程序设计题

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

```

#include <string.h>
#include <alloc.h>

#define DEFLINES    10                                /* n 的缺省值为 10 */
#define MAXLEN      81                                /* 这里，假设一行长度为 80 个字符 */

struct Tail {
    char data[MAXLEN];
    struct Tail *link;
};                                                    /* 定义循环链表中一个链结点构造 */

main(int argc, char *argv[ ])
{
    char curline[MAXLEN], *filename;
    int n=DEFLINES,i;                                /* n 的缺省值为 10 */
    struct Tail *list,*ptr, *qtr;
    FILE *fp;

    if(argc==3 && argv[1][0]!='-'){                  /* 进行命令行的参数正确性检查 */
        n=atoi(argv[1]+1);                          /* 将字符类型的 n 转换为整类型的 n */
        filename=argv[2];
    }
    else if(argc==2){
        filename=argv[1];
        else{                                          /* 命令行本身有错 */
            fprintf(stderr, "Usage: tail [-n] filename\n");
            exit(1);
        }
    }
    if((fp=fopen(filename, "r" )) == NULL){           /* 以只“读”方式打开文本文件 */
        fprintf(stderr, "Cann't open file: %s!\n", filename);
        exit(-1);
    }                                                  /* 该文本文件不能打开 */

    list=qtr=(struct Tail *)malloc(sizeof(struct Tail));
    qtr->data[0]='\0';
    for(i=1; i<n; i++){
        ptr=(struct Tail *)malloc(sizeof(struct Tail));
        ptr->data[0]='\0';
        qtr->link=ptr;
        qtr=ptr;
    }
    ptr->link=list;                                    /* 建立一个长度为 n 且不带头结点的单循环链表 */

    ptr=list;
    while(fgets(curline, MAXLEN,fp) != NULL){         /* 从文本文件中读一行放 curline 中 */
        strcpy(ptr->data, curline);                   /* 将读到的一行送链结点的数据域 */
        ptr=ptr->link;                                 /* ptr 指向下一个链结点 */
    }

    for(i=0; i<n; i++){
        if(ptr->data[0]!='\0')
            printf( "%s", ptr->data);                 /* 打印文本文件的一行 */
        ptr=ptr->link;
    }                                                  /* 打印文本文件的最后 n 行 */
    fclose(fp);                                       /* 关闭文本文件 */
    return 0;
}

```