

## 2015 年“数据结构与 C 语言程序设计”(科目代码: 991)

### 一、单项选择题

1. 递归定义的问题可以利用递归过程求解, 也可以利用非递归过程求解, 若仅从运行的时间效率来分析, 则下面给出的四种说法中, 正确的是\_\_\_\_\_。  
A. 前者要比后者快; B. 前者要比后者慢; C. 前者与后者相同; D. 两者不能做比较。
2. 线性表采用顺序存储结构的优点之一是\_\_\_\_\_。  
A. 存储密度(即存储利用率)高; B. 适用于各种逻辑结构的存储表示;  
C. 在表中进行插入操作的时间效率高; D. 在表中进行删除操作的时间效率高。
3. 若用  $STACK[n]$  表示某堆栈采用顺序存储结构, 则下列关于堆栈及堆栈操作的叙述中, 正确的是\_\_\_\_\_。  
A. 堆栈的大小为  $n$ ; B. 堆栈为空时  $n=0$ ; C. 最多只能进行  $n$  次进栈和出栈操作;  
D.  $n$  个元素依次进栈后, 它们的出栈顺序一定与进栈顺序相反。
4. 仅从形态上看, 具有  $n$  个结点且深度也为  $n$  的二叉树一共有\_\_\_\_\_。  
A.  $2n$  种; (显示有问题) B.  $2n$  种; (显示有问题) C.  $2n-1$  种; (显示有问题)  
D.  $2n-1$  种。 (显示有问题)
5. 对于题一 5 图所示的二叉树, 若对其遍历后得到的遍历序列为  $4, 6, 7, 5, 2, 3, 1$ , 则由此可以断定所采用的遍历方法是\_\_\_\_\_。  
A. 前序遍历; B. 中序遍历; C. 后序遍历; D. 按层次遍历。
6. 已知某无回路的有向图  $G$  的邻接表如题一 6 图所示。下列四个顶点序列中, 不属于  $G$  的拓扑序列的是\_\_\_\_\_。  
A.  $v1, v2, v3, v5, v4$ ; B.  $v1, v2, v3, v4, v5$ ; C.  $v2, v1, v3, v5, v4$ ; D.  $v2, v1, v5, v3, v4$ 。  
(题一 5 图略) (题一 6 图略)
7. 若具有  $n$  个顶点的连通图采用邻接矩阵表示, 则该邻接矩阵中的非零元素的个数至少是\_\_\_\_\_。  
A.  $2(n-1)$ ; (显示有问题) B.  $n-1$ ; C.  $n+1$ ; D.  $n/2$ 。
8. 在长度为  $n$  且元素按值有序排列的顺序表中进行折半查找, 查找每个元素所进行的元素之间的比较次数\_\_\_\_\_对应的“判定树”的深度(设深度  $\geq 2$ )。  
A. 大于; B. 等于; C. 小于; D. 小于或等于。
9. 若散列表的长度为  $m$ , 待散列的元素的个数为  $n$ , 装填因子为\_\_\_\_\_, 则散列表的平均查找长度\_\_\_\_\_。  
A. 与  $m$  直接相关; B. 与  $n$  直接相关; C. 与\_\_\_\_\_直接相关; D. 与  $m, n$  和\_\_\_\_\_都直接相关。
10. 若对序列  $(2, 12, 16, 70, 5, 10)$  按值从小到大进行排序, 前三趟排序的结果分别为:  
第 1 趟排序的结果:  $(2, 12, 16, 5, 10, 70)$ ,  
第 2 趟排序的结果:  $(2, 12, 5, 10, 16, 70)$ ,  
第 3 趟排序的结果:  $(2, 5, 10, 12, 16, 70)$ ,  
则由此可以断定, 该排序过程采用的排序方法是\_\_\_\_\_。  
A. 插入排序法; B. 选择排序法; C. 泡排序法; D. 快速排序法。

### 二、填空题

1. 若需要频繁地对线性表进行插入或删除操作，则该线性表宜采用 存储结构。
2. 若要以复杂度为  $O(1)$  的时间代价将两个单链表链接成一个单链表，则这两个单链表分别应该为 。
3. 若堆栈采用单链表作为存储结构，链结点构造为 **data link**，且栈顶指针为 **top**，则在该堆栈中插入一个由 **p** 指的新结点所执行的操作(语句)依次为 。
4. 若某满二叉树的结点总数为 20 至 40 之间的一个素数，则该满二叉树中叶结点的数目为 。
5. 若某二叉树的中序遍历序列为 (d, b, a, e, c, f)，按层次遍历序列为 (a, b, c, d, e, f)，则该二叉树的后序遍历序列为 。
6. 一个具有 36 条边的非连通无向图中至少有 个顶点。
7. 除了拓扑排序方法外，判断一个有向图是否存在回路的方法还有 。
8. 在顺序表(6, 15, 30, 37, 65, 68, 70, 72, 89, 99)中采用折半查找法查找元素 37 的过程中与表中进行过比较的元素依次为 。
9. 由经验得知，为了降低发生散列冲突的可能性，在采用除留余数法构造的散列函数  $H(k) = k \bmod p$  中，**p** 的取值最好是 。（**k** 为关键字）
10. 有一种排序方法的基本原理是：每一趟排序都从一个未排序的序列中选择第一个元素作为分界元素，该分界元素将当前未排序的序列分成前后两个部分，前一部分中的所有元素均小于等于分界元素，后一部分中的所有元素均大于等于分界元素，而分界元素处在排序的最终位置；然后分别对被分成的两部分中元素个数超过 1 的部分重复上述过程，直至排序结束。这种排序方法是 。

### 三、综合题

1. 假设 **n** 为 2 的乘幂，即  $n=2, 4, 8, 16, \dots$ 。请用大 **O** 符号形式写出下列函数的时间复杂度。

```
main( )
{ int n, x=2, sum=0;
  scanf("%d", &n);
  while(x<n/2){
    x=2*x;
    sum++;
  }
  printf("sum=%d", sum);
}
```

2. 已知某二叉树的前序遍历序列为 A, B, E, F, G, C, H, D, I, J，中序遍历序列为 E, F, G, B, H, C, I, J, D, A，请画出该二叉树。若该二叉树是由某树转换而来，则请画出转换之前的树。
3. 证明：若无向图 **G** 中每个顶点的度至少为 2，则 **G** 必然存在回路。
4. 折半查找法适用于什么线性表？为什么不能在按值大小有序链接的线性链表(单链表)中采用折半查找法查找链结点？

### 四、算法设计题

选择排序法的基本原理是：每一趟排序从当前未排好序的那些元素中选择一个值最小的元素，将其与未排好序的那些元素的第一个元素交换位置。请根据该原理写出对一个带有头结点的单链表按数据域值从小到大进行选择排序的算法。

设链表的头结点指针为 **list**。链结点类型为：

```

typedef struct node{
    int data;                                /* 数据域 */
    struct node *link;                         /* 指针域 */
}*LinkList;

```

每一个链结点的数据域中存放一个数据，但头结点数据域中不存放任何信息。

要求：（1）算法中不得增加和使用新的链结点空间；

（2）不得改变链结点的数据域中原有的内容。

## 五、单项选择题

1. 在 C 语言中，对于下列存储类别，只有在使用时才为该类型的变量分配内存空间的是\_\_\_\_\_。

A. auto 和 static; B. register 和 static; C. auto 和 register; D. extern 和 register。

2. 在 if 语句中用作判断的条件表达式为\_\_\_\_\_。

A. 算术表达式; B. 关系表达式; C. 逻辑表达式; D. 任意表达式。

3. 下列四个叙述中，正确的是\_\_\_\_\_。

A. char a[ ]="china";等价于 char a[6]; strcpy(a, "china");;

B. char b[10]={“china”};等价于 char b[10]; b[ ]={“china”};;

C. char c[6]={“china”, d[6]={“china”};等价于 char c[6]=d[6]={“china”};;

D. char \*e={"china"};等价于 char \*e; \*e="china";。

4. 判断字符串“abcd”与字符串“ab cd”是否相等应该使用\_\_\_\_\_。

A. if("abcd"=="ab cd"); B. if(abcd==ab cd);

C. if(strcmp(abcd, ab cd)); D. if(strcmp("abcd", "ab cd"))。

5. 下列四个函数中，与库函数 strcmp(char \*s, char \*t)的功能相等的是\_\_\_\_\_。

A. strcmp1(char \*s, char \*t) B. strcmp2(char \*s, char \*t)

```

{for( ; *s++==*t++; ) {for( ; *s++==*t++; )
    if(*s=='\0') return 0; if(!*s) return 0;
    return(*s-*t); return(*s-*t);
}

```

C. strcmp3(char \*s, char \*t) D. strcmp4(char \*s, char \*t)

```

{for( ; *t==*s; ){ {for( ; *s==*t; s++, t++)
    if(!*t) return 0; if(!*s) return 0;
    t++; s++; return(*t-*s);
}

```

6. 若已有定义和说明：int a[2][3], (\*p)[3]; p=a;, 则对 a 数组元素地址正确引用的是\_\_\_\_\_。

A. \*(p+2); B. (p+1)+2; C. p[1]+1; D. p[2]。

7. 若已有定义：float x;, 则下列对指针变量 p 进行定义且赋初值的语句中，正确的是\_\_\_\_\_。

A. float \*p=1024; ; B. int \*p=(float)x;; C. float p=&x;; D. float \*p=&x;。

8. 函数调用语句 func(rec1, rec2+rec3, (rec4, rec5));中的实在参数的个数是\_\_\_\_\_。

A. 2; B. 3; C. 4; D. 5。

9. 执行下列程序段以后, 变量 c 的值是\_\_\_\_\_。

```
int a=1, b=2, c;  
c=a^(b<<2);  
A. 9;      B. 8;      C. 7;      D. 6.
```

10. 下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>  
#define FUDGE(y)                            2.84+y  
#define PR(x)                                printf("%d", (int)(x))  
#define PRINT1(x)                            PR(x); putchar('\n')  
main( )  
{int a=2;  
PRINT1(FUDGE(5)*a);  
}  
A. 15;      B. 14;      C. 13;      D. 12.
```

## 六、填空题

1. 有程序段如下(设其中所使用的变量已经正确定义和赋值)：

```
for(s=1.0,k=1; k<=n; k++)  
    s=s+1.0/(k*(k+1));  
printf("s=%f\n", s);
```

请在下面程序段的空白处(横线上方)填入合适的内容, 使之与上述程序段的功能完全相同。

```
s=1.0;  
k=1;  
while(        ){  
    s=s+1.0/(k*(k+1));  
    ;  
}  
printf("s=%f\n", s);
```

2. 若已经定义 `int a[10];`, 则下列函数 `func()` 的功能是在第一个循环中给数组的 10 个元素依次赋予 `1, 2, 3, 4, 5, 6, 7, 8, 9, 10`; 在第二个循环中使数组 `a` 的后 5 个元素的值为前 5 个元素的逆序, 即数组变成 `1, 2, 3, 4, 5, 5, 4, 3, 2, 1`。

请在该函数的空白处(横线上方)填入合适的内容, 使函数完整。

```
func(int a[ ])  
{int i;  
for(i=1; i<=10; i++)                                            /* 分别给数组元素赋值 */  
    =i;  
for(i=0; i<5; i++)                                            /* 重新给某些数组元素赋值 */  
    =a[i];  
}
```

3. 下列程序的功能是找出通过键盘输入的 5 个字符串中最大的字符串。

请在程序的空白处(横线上方)填入合适的内容, 使程序完整。

```
#include <stdio.h>  
#include <string.h>  
main( )
```

```
{char s[5][50], *p;  
int i;  
for(i=0; i<5; i++) /* 依次输入 5 个字符串 */  
    gets(s[i]);  
    ;  
for(i=1; i<5; i++) /* 寻找最大字符串 */  
    if(strcmp(p, s[i])<0)  
        ;  
puts(p); /* 输出最大字符串 */  
}
```

4. 下列程序的功能是先通过键盘接收一个字符串，然后按照字符顺序对串中字符从小到大进行排序，同时删除串中重复出现的字符。

请在程序的空白处(横线上方)填入合适的内容, 使程序完整。

```
#include <stdio.h>
#include <string.h>
main( )
{ char s[100], *p, *q, *r, ch;
  printf("Please input a string:");
  gets(s);                                     /* 输入一字符串 */
  for(p=s; *p; p++){
    for(q=r=p; *q; q++)
      if(*r>*q)
        r=q;
    if(      ){
      ch=*r; *r=*p; *p=ch;
    }
  }
  for(p=s; *p; p++){
    for(q=p; *p==*q; q++);
      strcpy(      , q);
  }
  printf("result: %s\n", s);
  /* 输出结果字符串 */
}
```

5. 下列函数 move 的功能是将长度为 n 的序列 a 中各元素依次顺序循环右移 m 个位置。例如, 对于序列  $a=(1, 3, 5, 7, 9, 11)$ , 若  $m=2$ , 则循环右移后的结果为  $a=(9, 11, 1, 3, 5, 7)$ 。请在函数的空白处(横线上方)填入合适的内容, 使函数完整。

```
move(int *a, int n, int m)
{int *p, k, temp;
 for(k=1; k<=m; k++){
    temp=      ; /* 保存序列的最后那个元素 */
    for(p=a+n-1;      ; p--)
        *p=*(p-1); /* 将元素右移一个位置 */
    *a=temp;
}
```

}

6. 下列递归函数 sum(int n)的功能是计算  $\sum i$  。 (i=1,2,3, ..., n)

请在函数的空白处(横线上方)填入合适的内容, 使函数完整。

```
sum(int n)
{ int result;
i f(n<=0)
    printf("\n 数据有问题!");
if(n==1)
    ;
else
    ;
return result;
}
```

7. 下列函数的功能是计算并返回所使用的计算机中 int 类型数据的字长(即二进制位)的位数。

请在函数的空白处(横线上方)填入合适的内容, 使函数完整。

```
wordlength( )
{int i;
unsigned int v=      ;           /* 将各二进制位置初值 */
for(i=1; (v=v>>1)>0; i++);    /* 统计二进制位数 */
return(      );
}
```

8. 下列程序的功能是实现以命令行形式给出的文件复制。设命令行的形式为:

命令名 文件 1 文件 2

即把文件 2 的内容复制到文件 1 中。

请在程序的空白处(横线上方)填入合适的内容, 使程序完整。

```
#include <stdio.h>
main(int argc, char *argv[ ])
{ FILE *out, *in;
if(argc!=3)
    return;
if((in=fopen(argv[2], "rb"))==NULL)           /* 打开输入文件 */
    return;
out=      ;           /* 建立输出文件 */
fcopy(out, in);
fclose(in);
fclose(out);
}
void fcopy( FILE *fout, FILE *fin)           /* 复制文件内容 */
{ char ch;
do{
    ch=fgetc(fin);
    if(feof(fin))
        break;
}
```

```

        fputc(      );
    }while(1);
}

9. 下列程序的功能是显示指定文件的内容，并且在显示内容的同时加上相应的行号。
请在程序的空白处(横线上方)填入合适的内容，使程序完整。
#include <stdio.h>
main( )
{ char s[80], filename[20];
    int i=0, flag=1;                      /* 行号和处理标志置初值 */
    FILE *fp;
    printf("Enter filename:");
    gets(filename);                      /* 输入指定文件名 */
    if((fp=fopen(filename, "r"))==NULL){
        printf("File cannot open!");
        exit(0);
    }
    while(fgets(s, 80, fp)!=NULL){          /* 从文件中读取一行 */
        if(      )
            printf("%3d: %s", ++i, s);      /* 显示当前行号和一行的内容*/
        else
            printf("%s", s);
        if(      )
            flag=1;
        else
            flag=0;
    }
    fclose(fp);
}

```

10. 下列程序的功能是把通过键盘输入的 10 个 **float** 类型的数据以二进制方式写到名为 **number.dat** 的文件中。

请在程序的空白处(横线上方)填入合适的内容，使程序完整。

```

#include <stdio.h>
main( )
{ int i;
    float n;
    FILE *fp;
    if((fp=fopen("number.dat", "wb"))==NULL) /* 打开输出文件 */
        exit(0);
    for(i=0; i<10; i++){
        ;                                /* 从键盘读入一个数据 */
        ;                                /* 把数据写到文件中 */
    }
    fclose(fp);
}

```

## 七、程序设计题

请编写一程序，该程序的功能是首先将用户通过键盘输入的若干字符(用 **EOF** 结束输入)存入一维数组 **s** 中，然后找出数组中具有最大 **ASCII** 码值的字符，并且输出该字符以及该字符对应的 **ASCII** 码。

要求： 程序中有关输入、输出以及查找等操作必须通过指针完成。

## 八、程序设计题

请编写一程序，该程序的功能是确定字符串中首次出现的某个字符在串中的位置(即该字符是字符串中的第几个字符)，然后从字符串中删除该字符。

要求： (1) 将确定字符位置以及删除该字符的过程编写为一个独立的函数。(注：函数中不考虑非首次出现的该字符的删除)

(2) 在主函数中通过键盘输入字符串和被确定的字符。若字符串中没有被确定的字符，程序给出相应信息，否则，输出该字符在字符串中首次出现的位置，并且显示删除前、后的字符串。

北京航空航天大学 2014 年  
硕士研究生入学考试试题 科目代码: 991

数据结构与 C 语言程序设计 (共 8 页)

考生注意: 所有答案务必写在考场提供的答题纸上, 写在本试题单上的答案一律无效 (本题单不参与阅卷)。

一、填空 (本题共 20 分, 每小题各 2 分)

1. 设带有头结点的非空单向循环链表的链结点构造为 

data	link
------	------

, rear 为指向链尾结点的指针。删除链表头结点后面那个结点应该依次执行的语句是\_\_\_\_\_。
2. 为了增加空间的利用率和减少溢出的可能性, 在两个堆栈共享一片连续空间时通常将两个堆栈的栈底位置分别设置在这片空间的两端, 当\_\_\_\_\_时才会产生上溢。
3. 如果一棵二叉树有 1024 个结点, 其中 465 个是叶结点, 那么, 该二叉树中度为 1 的结点的个数是\_\_\_\_\_。
4. 已知在一棵二叉树中, a 是 b 的祖先结点, 若要通过遍历操作找到从 a 到 b 的路径, 则在前序遍历、中序遍历、后序遍历和按层次遍历这 4 种遍历方法中, 应该选择\_\_\_\_\_。
5. 如果从无向图中的任意一个顶点出发进行 1 次深度优先搜索便可以访问到图中的所有顶点, 这样的图一定是\_\_\_\_\_。
6. 在顺序表(b, c, d, e, f, g, q, r, s, t)中采用折半查找法查找元素 b 的过程中, 被比较过的元素依次为\_\_\_\_\_。
7. 所谓 m 阶 B-树中的 m 是指\_\_\_\_\_。
8. 若 n 个关键字互为同义词, 并且采用线性探测再散列法处理冲突, 则将这组关键字散列到一个散列空间中, 需要进行的探测次数为\_\_\_\_\_。
9. 在插入排序法、选择排序法、泡排序法和快速排序法这 4 种排序方法中有一种方法, 若初始时最小值元素位于待排序序列的最后, 则在最后一趟排序开始之前, 所有元

素都不在其最终位置上。这种排序方法是\_\_\_\_\_。

19. 若采用直接排序法对序列(tang, deng, an, wang, shi, bai, fang, liu)中元素按值从小到大进行排序，则第2趟排序结束时的结果是\_\_\_\_\_。

### 二、简答(本题共 20 分, 每小题各 5 分)

1. 一个完整的算法通常应该具备哪 5 个最基本的特性? (可以不对各特性的具体意义进行解释)

2. 对于长度为  $n$  的线性表, 下面给出的 4 种操作中, 在顺序表上实现比在链表上实现效率更高的是哪一种? 请分别用大 O 符号表示的时间复杂度形式说明你的理由。

- (1) 输出线性表中第  $i$  个数据元素的值 ( $1 \leq i \leq n$ );
- (2) 交换线性表中第 1 个数据元素与第 2 个数据元素的值;
- (3) 依次输出线性表的  $n$  个数据元素;
- (4) 输出线性表中与给定值  $x$  相匹配的数据元素在线性表中的序号。

3. 具有  $n$  个结点且深度也为  $n$  的二叉树一共有多少种? 请具体说明你的结论。

4. 为什么在建立散列表时若采用线性探测再散列法处理散列冲突容易产生聚集 (clustering)? 采用其他什么方法可以减少这种聚集?

### 三、问题求解 (本题共 20 分, 每小题各 5 分)

1. 设有编号分别为 1, 2, 3 的三辆列车顺序进入一个栈式结构的车站站台, 请分别写出这三辆列车开出车站所有可能的顺序。

2. 已知对某二叉排序树进行前序遍历得到的前序遍历序列为 (60, 45, 35, 40, 50, 65, 75, 70), 请画出该二叉排序树。

3. 若某无向图一共有 16 条边, 并且有 3 个度为 4 的顶点, 4 个度为 3 的顶点, 其余顶点的度均小于 3, 则该无向图至少有多少个顶点? (请写出结论的求解过程)

4. 在设计快速排序法的非递归算法时, 通常利用了一个堆栈来记录待排序区间的首、尾两个端点的位置, 而实际上也可以利用其他数据结构(如队列)来代替这个堆栈。请说明其中的理由。

#### 四、算法设计 (本题 15 分)

已知某具有  $n$  个顶点的有向图采用邻接表方法存储，其中，用以存储有向边信息的边结点类型为

```
typedef struct edge{  
    int adjvex; /* 存放某有向边的终止顶点在顶点结点中的位置 */  
    struct edge *next; /* 指向下一个边结点 */  
}ELink;
```

用以存储顶点信息的顶点结点类型为

```
typedef struct ver{  
    int degree; /* 存放某顶点的度 */  
    vertype vertex; /* 存放一个顶点的数据信息 */  
    ELink *link; /* 指向以该顶点为出发点的第一个边结点 */  
}VLink;
```

并且  $n$  个顶点结点构成一个数组  $G[0..n-1]$ 。请写一个算法，该算法依次求出图中各顶点的度，并分别存放在相应的顶点结点的  $degree$  域中。

#### 五、单项选择 (本题共 20 分，每小题各 2 分)

1. 下列关于 C 语言的自增和自减运算符的使用中，正确的是\_\_\_\_\_。

- A.  $15++$ ;      B.  $(x+y)++$ ;      C.  $++(a-b)$ ;      D.  $s+++t+++u++$ 。

2. 若有  $int x=10;$ ，则执行  $y=(2*8, x+=5)$  以后， $x$  和  $y$  的值分别是\_\_\_\_\_。

- A. 15 和 16;      B. 15 和 15;      C. 16 和 15;      D. 16 和 16。

3. 对于下列 4 种输入函数，当用户要输入的字符串中含有空格字符时，应该使用\_\_\_\_\_。

- A. `scanf()`;      B. `getchar()`;      C. `gets()`;      D. `getc()`。

4. 对于  $char s1[] = {"abcd"}; s2[] = {'a', 'b', 'c', 'd'};$ ，下列叙述中，正确的是\_\_\_\_\_。

- A. 数组  $s1$  与数组  $s2$  等价;  
B. 数组  $s1$  的长度与数组  $s2$  的长度相等;  
C. 数组  $s1$  的长度大于数组  $s2$  的长度;  
D. 数组  $s1$  的长度小于数组  $s2$  的长度。

5. 设已有如下定义和语句:

```
int a=4, b=3, *ptr, *ptr, *t;  
ptr=&a; qtr=&b; t=qtr; qtr=NULL;
```

下列4条赋值语句中, 错误的是\_\_\_\_\_。

A. \*qtr=0;

B. t=ptr;

C. \*ptr=a;

D. \*ptr=\*t;

6. 若从键盘上输入 abc def, 则下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>  
#include <stdlib.h>  
void main()  
{  
    char *ptr, *qtr;  
    ptr=(char *)malloc(sizeof(char)*20);  
    qtr=ptr;  
    scanf("%s %s", ptr, qtr);  
    printf("%s%s", ptr, qtr);  
}
```

A. defabc;

B. defdef;

C. abcdef;

D. abcabc.

7. 已知变量定义如下:

```
struct{  
    float x, y;  
    char s[10];  
}point; *ptr=&point;
```

下列4个表达式中, 错误的是\_\_\_\_\_。

A. ptr->x=2.0;    B. (\*ptr).y=3.0;    C. point.x=2.0;    D. ptr->s="str"。

8. 若有宏定义 "#define AREA(a, b) a\*b", 则下列"宏调用"中, 正确的是\_\_\_\_\_。

A. s=AREA(r\*t);

B. s=AREA(x\*y);

C. s=c\*AREA((x=3.5), (y+4.1));

D. s=AREA。

9. 设下列程序经编译链接后生成的可执行文件是 test.exe。若运行该程序时输入带参数的命令行为 test abcd efg h3 k44<回车>, 则程序执行后的输出结果是\_\_\_\_\_。

```
#include <stdio.h>  
#include <string.h>  
void main(int argc, char *argv[ ])
```

```
int i, len=0;
for(i=1; i<argc; i+=2)
    len+=strlen(argv[i]);
printf("%d\n", len);
```

A. 6; B. 8;

C. 12;

D. 14.

10. 下列程序的功能是\_\_\_\_\_。

```
#include <stdio.h>
void main()
{
    FILE *fp;
    fp=fopen("file.dat", "r+");
    while(!feof(fp))
        if(fgetc(fp)=='*')
            fseek(fp, -1L, SEEK_CUR);
            fputc('#', fp);
            fseek(fp, ftell(fp), SEEK_SET);
    }
    fclose(fp);
}
```

- A. 将文件 file.dat 中的所有字符都换成#;
- B. 将文件 file.dat 中所有的\*都换成#;
- C. 查找文件 file.dat 中所有的\*;
- D. 查找文件 file.dat 中所有的#。

六、简答 (本题共 20 分, 每小题各 4 分)

1. 在 C 语言中, `int a[10]; a++;` 是否正确? 为什么?
2. 在 C 语言中, `strlen` 与 `sizeof` 的区别是什么?
3. 使用函数 `strcpy(char *s1, char *s2)` 应该注意哪些问题?
4. 在 C 语言中, 什么是函数指针? 什么是指针函数?
5. 递归程序在进行递归调用时通常用到递归工作栈, 请问: 通常情况下, 该递归栈中需要保存哪些信息?

七、程序阅读填空（本题共 20 分，每小题各 4 分）

1. 若已有定义 int a[10]，则下列函数 FUNC( )的功能是在第一个循环中给数组的 10 个元素依次赋予 1, 2, 3, 4, 5, 6, 7, 8, 9, 10；在第二个循环中使数组 a 的后 5 个元素的值为前 5 个元素的逆序，即数组变成 1, 2, 3, 4, 5, 5, 4, 3, 2, 1。

为了使该函数正确、完整，请写出函数的空白处(方框内)应该填入的内容。

```
FUNC(int a[ ])  
{    int k;  
    for(k=1; k<=10; k++)  
        [ ]=k;  
    for(k=0; k<5; k++)  
        [ ]=a[k];  
}
```

2. 下列函数 int MINPRIME(int num)的功能是返回大于 num 的最小素数。例如：若 num=13，则函数返回素数 17；若 num=29，则函数返回素数 31。

为了使该函数正确、完整，请写出函数的空白处(方框内)应该填入的内容。

```
#include <stdio.h>  
int MINPRIME(int num)  
{    int k, j;  
    k=num+1;  
    while(1){  
        for(j=2; j<=k-1; j++)  
            if([ ])                /* 如果 k 不是素数 */  
                [ ];  
        if(j>k-1)  
            return k;  
        else k++;  
    }  
}
```

3. 下列函数 int INDEX(char str[ ], char substr[ ])的功能是返回子串 substr 在主串 str 中的位置；若主串 str 中不存在子串 substr，则函数返回 0。

提示：所谓一个子串在包含它的主串中的位置是指主串中首次出现的这个子串的第 1

个字符在主串中的位置。例如：若 str=“Beijing and Nanjing”，substr=“jing”，则 substr 在 str 中的位置为 4。

为了使该函数正确、完整，请写出函数的空白处(方框内)应该填入的内容。

```
#include <stdio.h>
int INDEX(char str[], char substr[])
{
    int i, j, k;
    for(i=0; str[i]; i++)
        for(j=i, k=0; str[j]==substr[k]; j++, k++)
            if([ ] == substr[k])
                return [ ];
    return 0;
}
```

4. 下列函数的功能是用字符串 str2 替换字符串 olds 中所有出现的子串 str1，olds 串中的其他字符不变，并且将形成的新字符串存放在 news 中。

例如：若 olds=“123ab123x”，str1=“123”，str2=“45”，则 news=“45ab45x”。

为了使该函数正确、完整，请写出函数的空白处(方框内)应该填入的内容。

```
void REPLACE(char *olds, char *str1, char *str2, char *news)
{
    char *p, *q;
    while(*olds != '\0'){
        for(p=olds, q=str1; *p=='\0' && *q=='\0' && [ ] == str1; p++, q++);
        if(*q != '\0') /* 在 olds 中查找与 str1 相匹配的子串 */
            *news++ = [ ];
        else /* 若在 olds 中未找到与 str1 相匹配的子串 */
            for(q=str2; *q=='\0'; q++)
                *news++ = *q;
        olds = p;
    }
}
```

5. 下列程序的功能是找出由数字字符组成的文本文件 file.dat 中的所有不同整数，并且分别将这些整数存放在一维数组 a 中。

约定：该文本文件中的各整数之间以空格字符或者 Tab(制表符)或者回车符分隔，并且假定文件中的不同整数不超过 1000 个。

为了使该程序正确、完整，请写出程序的空白处(方框内)应该填入的内容。

```
#include <stdio.h>
#define N 1000
void main()
{
    FILE *fp;
    int a[N], i, num, count;
    if((fp=_____)==NULL){           /* 打开文本文件 */
        printf("Can not open file!\n");
        exit(0);
    }
    count=0;
    while(fscanf(_____)==1){          /* 从文件中读一个整数 */
        a[count]=num;
        for(i=0; a[i]!=num; i++);
        if(_____)
            count++;                  /* 累计不同整数的个数 */
    }
    fclose(fp);
}
```

#### 八、程序设计 (本题 15 分)

请编写一程序，该程序的功能是计算并输出某子串 substr 在主串 str 中出现的次数。

要求：

(1) 把求子串在主串中出现次数的过程编写为一个独立的函数：

```
int STRCOUNT(char *str, char *substr)
```

(2) 若主串 str 中未出现子串 substr，则函数 STRCOUNT(char \*str, char \*substr)返回 0。

(3) 在主函数中通过键盘输入方式分别给 str 与 substr 赋值，并且所有涉及到字符串的操作均通过指针完成。

- 一、 1. `P=rear->link->link;rear->link->link=p->link;free(p);`  
 2. 两个栈顶相遇 3. 95 4. 后序 5. 连通图 6. f,c,b  
 7. 每个分支结点最多有  $m$  棵子树 8.  $n(n-1)/2$  9. 插入排序法  
 10. an,deng,shi,bai,fang,liu,tang,wang

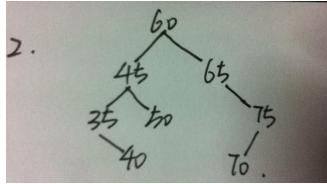
- 二、 1. 有穷性、确定性、可行性、输入、输出

2.	(1)	(2)	(3)	(4)
顺	$O(1)$	$O(1)$	$O(n)$	$O(n)$
链	$O(n)$	$O(1)$	$O(n)$	$O(n)$

3.  $2^{(n-1)}$

4. 链地址

- 三、 1. 321 231 213 123 132



3. 11 4. 可以 (08 年 - 4)

- 四 void DEGREE(VLink G[], int n)

```

{  int num,i;  VLink v;  ELink *p;
  for(i=0; i<n; i++){
    v=G[i];
    num=0;
    p=v.link;
    while(p!=NULL){
      num++;
      p=p->next;
    }
    G[i].degree=num;
  }
}
  
```

- 五、 1-5 DACCA 6-10 BDCAB

六、 1. 不正确，因为数组名 `a` 代表数组首元素的地址，它是一个指针常量，它的值在程序运行期间是不变的

2. `sizeof` 是算符，`strlen` 是函数；`strlen` 是返回字符串的长度，而 `sizeof` 却是用来取得变量所占用的内存；返回的单位不一样

3. `s1` 必须定义得足够大，以便容纳被复制的字符串 `s2`，`s1` 的长度不应小于 `s2` 的长度

4. 函数指针是一个指向某函数起始地址的指针；指针函数是带指针的函数，返回类型是某一类型的指针

5. 返回地址信息、实参信息

- 七、 1. `a[k-1] a[9-k]` 2. `k%j==0` continue  
 3. `Substr[k+1]=='\0'` `i+1` 4. `*p==*q` `*olds++` `*new='0'`  
 5. `fopen("file.dat", "r")` `fp, "%d", &num` `i==count`

- 八、 `#include<stdio.h>`

`#include<string.h>`

```
main()
{
    int STRCOUNT(char *str, char *substr);
    char str[80], substr[80];  int num;
    printf("Input string:");
    gets(str);
    printf("Input substring:");
    gets(substr);
    num=STRCOUNT(str, substr);
    printf("num = %d", num);
}

int STRCOUNT(char *str, char *substr) //根据第七大题 4 改写的
{
    char *p, *q;
    int num=0;
    while(*str!='\0'){
        for(p=str, q=substr; *p=='\0'&&*q=='\0'&&*p==*q; p++, q++);
        if(*q=='\0')
            *str++;
        else{
            num++;
            str=p;
        }
    }
    return num;
}
```

By Jason Lee  
Roc°  
卡卡

## 2013 年“数据结构与 C 程序设计”(代码 991)试题

### 一、单项选择题(本题共 20 分, 每小题各 2 分)

1. 对于长度为  $n$  的线性表, 建立其对应的单链表的时间复杂度为( )。  
A.  $O(1)$ ; B.  $O(\log_2 n)$ ; C.  $O(n)$ ; D.  $O(n^2)$ 。
2. 一般情况下, 在一个双向链表中插入一个新的链结点, ( )。  
A. 需要修改 4 个指针域内的指针; B. 需要修改 3 个指针域内的指针;  
C. 需要修改 2 个指针域内的指针; D. 只需要修改 1 个指针域内的指针。
3. 假设用单个字母表示中缀表达式中的一个运算数(或称运算对象), 并利用堆栈产生中缀表达式对应的后缀表达式。对于中缀表达式  $A+B*(C/D-E)$ , 当从左至右扫描到运算数  $E$  时, 堆栈中的运算符依次是( )。  
(注: 不包含表达式的分界符)  
A.  $+*/-$ ; B.  $+*(/-$ ; C.  $+*-$ ; D.  $+*(-$ 。
4. 若某二叉排序树的前序遍历序列为 50, 20, 40, 30, 80, 60, 70, 则后序遍历序列为( )。  
A. 30, 40, 20, 50, 70, 60, 80; B. 30, 40, 20, 70, 60, 80, 50;  
C. 70, 60, 80, 50, 30, 40, 20; D. 70, 60, 80, 30, 40, 20, 50。
5. 分别以 6, 3, 8, 12, 5, 7 对应叶结点的权值构造的哈夫曼 (Huffman) 树的深度为( )。  
A. 6; B. 5; C. 4; D. 3。
6. 下列关于图的叙述中, 错误的是( )。  
A. 根据图的定义, 图中至少有一个顶点;  
B. 根据图的定义, 图中至少有一个顶点和一条边(弧);  
C. 具有  $n$  个顶点的无向图最多有  $n(n-1)/2$  条边;  
D. 具有  $n$  个顶点的有向图最多有  $n(n-1)$  条边(弧)。
7. 若在有向图  $G$  的拓扑序列中, 顶点  $v_i$  在顶点  $v_j$  之前, 则下列 4 种情形中不可能出现的是( )。  
A.  $G$  中有弧  $\langle v_i, v_j \rangle$ ;  
B.  $G$  中没有弧  $\langle v_i, v_j \rangle$ ;  
C.  $G$  中有一条从顶点  $v_i$  到顶点  $v_j$  的路径;  
D.  $G$  中有一条从顶点  $v_j$  到顶点  $v_i$  的路径。
8. 下列关于查找操作的叙述中, 错误的是( )。  
A. 在顺序表中查找元素可以采用顺序查找法, 也可以采用折半查找法;  
B. 在链表中查找结点只能采用顺序查找法, 不能采用折半查找法;  
C. 一般情况下, 顺序查找法不如折半查找法的时间效率高;  
D. 折半查找的过程可以用一棵称之为“判定树”的二叉树来描述。
9. 在一棵  $m$  阶 B-树中, 除根结点之外的任何分支结点包含关键字的个数至少是( )。  
A.  $m/2-1$ ; B.  $m/2$ ; C.  $m/2-1$ ; D.  $m/2$ 。
10. 若对序列(49, 38, 65, 97, 76, 13, 27, 49')进行快速排序, 则第一趟排序结束(即确定了第 1 个分界元素的最终位置)时, 序列的状态是( )。  
A. (13, 27, 49', 38, 49, 76, 97, 65); B. (13, 38, 27, 49', 49, 76, 97, 65);  
C. (13, 38, 49', 27, 49, 97, 76, 65); D. (13, 38, 49', 27, 49, 76, 97, 65)。

### 二、填空题(本题共 20 分, 每小题各 2 分)

1. 非空线性表在采( )存储结构的情况下, 删除表的一个数据元素平均需要移动表中近一半元素的位置。
2. 将一个长度为  $n$  的单链表链接到一个长度为  $m$  的单链表后面, 该算法的时间复杂度用大  $O$  符号表示为( )。
3. 若完全二叉树的叶结点的数目为  $k$ , 且最下面一层的结点数大于 1, 则该完全二叉树的深度为( )。

4. 若深度为 8 的完全二叉树的第 7 层有 10 个叶结点，则该二叉树的结点总数为( )。
5. 在具有  $n$  个顶点的有向图中，每个顶点的度最大可以达到( )。
6. 若对有向图进行拓扑排序，则能够得到拓扑序列的条件是( )。
7. 已知长度为 10 的顺序表中数据元素按值从小到大排列。若在该表中进行折半查找，则平均查找长度(ASL)是( )。
8. 若在一棵  $m$  阶 B-树的某个结点中插入一个新的关键字值而引起结点产生分裂，则该结点中原有的关键字值的数目是( )。
9. 有一种排序方法可能会出现这种情况：最后一趟排序开始之前，序列中所有的元素都不在其最终应该在的位置上，这种排序方法是( )。
10. 若按照泡排序法的思想将序列(2, 12, 16, 5, 10)中元素按值从小到大进行排序，整个排序过程中所进行的元素之间的比较次数为( )。

### 三、综合题(本题共 20 分，每小题各 5 分)

1. 一般情况下，当一个算法中需要建立多个堆栈时可以选用下列三种处理方案之一。问：这三种方案之间相比较各有什么优点和缺点？

- (1) 多个堆栈共享一个连续的存储空间；
- (2) 分别建立多个采用顺序存储结构的堆栈；
- (3) 分别建立多个采用链式存储结构的堆栈。

2. 已知二叉树采用二叉链表存储结构，根结点指针为 T，链结点类型定义为：

```
typedef struct node{
```

```
    char data;           /* 数据域 */
    struct node *lchild, *rchild; /* 指向左、右子树的指针域 */
} *BTREE;
```

下面的算法的功能是输出二叉树中所有叶结点的数据信息。

请在算法的空白处(符号-----处)填入合适内容，使算法完整。

```
void FUNC(BTREE T)
{
    if(T!=NULL){
        if((-----)
            printf("%c", T->data);
        FUNC(-----);
        FUNC(-----);
    }
}
```

3. 对给定 AOE 网(如题三 3 图所示)，请完成

- (1) 分别求出各活动  $ai(i=1, 2, \dots, 14)$  的最早开始时间与最晚开始时间；(以表格形式给出结果)

- (2) 求出所有关键路径。(请以图形方式画出各关键路径)

(说明：由于题三 3 图在本网站内无法显示，可参见指定教材 p280 页 8-16 题)

4. 已知将给定的关键字值序列(42, 51, 16, 26, 50, 25, 37, 68, 64, 33, 18)进行散列存储，并且要求装填因子(也称负载因子) $\alpha \approx 0.61$ ，

- (1) 请利用除留余数法构造出合适的散列函数；

- (2) 请画出利用该散列函数依次将序列中各关键字值插入到散列表以后表的状态。设散列表初始为空，并且采用线性探测再散列法处理散列冲突。

### 四、算法设计题(本题 15 分)

假设长度为  $n$  的顺序表  $A[1..n]$  中每个数据元素为一整数, 请写出按照下列思想将表中数据元素按值从小到大进行排序的算法: 第 1 趟排序将最小值元素放在  $A[1]$  中, 最大值元素放在  $A[n]$  中; 第 2 趟排序将次小值元素放在  $A[2]$  中, 次大值元素放在  $A[n-1]$  中; ……, 依此下去, 直至排序结束。

### 五、填空题(本题共 20 分, 每小题各 2 分)

1. 已知某等比数列的第一项  $a_1$  为 1, 公比为 3, 下列程序的功能是输出该数列中小于 1000 的最大项  $a_n$  及其对应的  $n$ 。

请在程序的空白处(符号-----处)填入合适内容, 使程序完整。

```
main( )
{
    int n=1, a=1, q=3;
    while(1){
        a=a*q;
        n++;
        if(a>=1000)
            ----;
    }
    printf("n=%d,a=%d\n", n-1, -----);
}
```

2. 下列递归函数 FUNC2 的功能是判断整型数组  $a[n]$  是否为递增数组, 即判断数组的元素是否按值从小到大排列。若是一个递增数组, 则函数返回 `true`, 否则, 函数返回 `false`。

请在函数的空白处(符号-----处)填入合适内容, 使函数完整。

```
bool FUNC2(int a[ ], int n)
{
    if(n==1)
        return true;
    if(n==2)
        return ----;
    return ---- && (a[n-1]>=a[n-2]);
}
```

3. 下列程序的功能是主函数调用 FUNC3 函数求方阵  $a$  中两条对角线上元素之和。

请在程序的空白处(符号-----处)填入合适内容, 使程序完整。

```
#define N 10
void FUNC3(int a[N][N], int *p, int *q)
{
    int i;
    *p=0;
    *q=0;
    for(i=0; i<N; i++){
        *p=*p+(*-----);
        *q=*q+(*-----);
    }
}
main( )
{
    int a[N][N], i, j, x, y;
    for(i=0; i<N; i++)
        for(j=0; j<N; j++)
            a[i][j]=i+j;
    x=a[0][0];
    y=a[N-1][N-1];
    FUNC3(a, &x, &y);
    printf("x=%d, y=%d\n", x, y);
}
```

```

for(j=0; j<N; j++)
    scanf("%d", *(a+i)+j);
FUNC3(a, &x, &y); /* x, y 中分别存放主对角线与副对角线上的元素之和 */
printf("%d, %d\n", x, y);
}

```

4. 下列程序的功能是先通过键盘输入一正整数，然后调用一递归函数 **FUNC4**，该函数将正整数转换为对应的数字字符组成的字符串显示在屏幕上。例如：若输入的正整数为 583，则屏幕上显示的是字符串 583。请在程序的空白处(符号-----处)填入合适内容，使程序完整。

```

#include <stdio.h>
void FUNC4(int n)
{ int i;
    i=n/10;
    if(-----)
        FUNC4(i);
    putchar(-----);
}
main( )
{ int n;
    printf("请输入一正整数 n: ");
    scanf("%d", &n);
    printf("转换后的字符串是: ");
    FUNC4(n);
}

```

5. 下列程序的功能是将小写字母转换成对应的大写字母后的第 2 个字母，例如，将 a 转换成 C，将 b 转换成 D，其中，y 转换成 A，z 转换成 B。

请在程序的空白处(符号-----处)填入合适内容，使程序完整。

```

#include <stdio.h>
main( )
{ char ch;
    while((ch=getchar( ))!='\n')
        if(ch>='a' && ch<='z'){
            -----;
            if(ch>'Z' && ch<='Z'+2)
                -----;
        }
}

```

6. 下列函数 **FUNC6** 的功能是删除字符串 s 中的所有空白字符，包括 **Tab** 字符、回车符以及换行符。

请在函数的空白处(符号-----处)填入合适内容，使函数完整。

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>
FUNC6(char *s)
{ int i, t;
    char c[80];

```

```

for(i=0,t=0; s[i]; i++)
    if(!isspace(-----))
        c[-----]=s[i];
    c[t]='\0';
    strcpy(s, c);
}

```

7. 下列程序的功能是判断输入的字符串是否是“回文”。(注:按顺序读与按逆序读都一样的字符串被称为“回文”，例如: `abcdcba`)。

请在程序的空白处(符号-----处)填入合适内容，使程序完整。

```

#include <stdio.h>
#include <string.h>
main( )
{
    char ch[81], *p=ch, *q;
    gets(p);
    q=p+-----;
    while(-----){
        if(*p==*q){
            p++; q--;
        }
        else
            break;
    }
    if(p<q)
        printf("该字符串不是回文! \n");
    else
        printf("该字符串是回文! \n");
}

```

8. 下列程序的功能是: 对于字符类型变量 `ch=108`，保留中间两位，而将高、低 3 位清零。

请在程序的空白处(符号-----处)填入合适内容，使程序完整。

```

main( )
{
    char ch;
    ch=108;
    ch=-----;
    printf("%d", ch);
}

```

9. 设 `file` 为存放了整型数据的二进制文件。下列程序的功能是从该文件中读入第 3 个数据输出到屏幕上。

请在程序的空白处(符号-----处)填入合适内容，使程序完整。

```

#include <stdio.h>
main( )
{
    FILE *fp;
    int number;
    fp=fopen("file","rb");
    fseek(fp, -----, SEEK_SET);
    fread(-----, 1, fp);
}

```

```

    printf("%d", number);
    fclose(fp);
}

10. 下列程序的功能是将一个磁盘中的二进制文件复制到另一个磁盘中。两个文件的文件名随命令行一起
输入，输入时原有文件的文件名在前，新复制文件的文件名在后。
请在程序的空白处(符号-----处)填入合适内容，使程序完整。
#include <stdio.h>
main(int argc, char *argv[ ])
{ FILE *old, *new;
if(argc!=3){
    printf("You forgot to enter a filename!\n");
    exit(0);
}
if((old=fopen(-----)==NULL){
    printf("Cannot open infile!\n");
    exit(0);
}
if((new=fopen(-----)==NULL){
    printf("Cannot open outfile!\n");
    exit(0);
}
while(!feof(old))
    fputc(fgetc(old), new);
fclose(old);
fclose(new);
}

```

#### 六、简答题（本题共 20 分，每小题各 5 分）

1. 在 C 语言中，函数调用时数据的传递通常有哪几种方式？
2. 在 C 语言中，指针可以做哪些运算？
3. 共用体(union)具有哪些基本特征？
4. 使用文件的基本操作步骤是怎样的？

#### 七、程序设计题（本题 15 分）

请编写一程序，该程序的功能是找出并且删除一维整型数组 a[100]中的最小值元素。

- 要求：
1. 数组各元素通过键盘输入获得初值；
  2. 所有对数组元素的引用必须通过指针完成。

#### 八、程序设计题（本题 20 分）

请仅编写出一 C 语言函数 `char *maxword(char *s, char *t)`，该函数的功能是求出字符串 `s` 与字符串 `t` 的最长公共单词(这里，假设两个字符串均由英文字母和空格字符组成)；若找到这样的公共单词，函数返回该单词，否则，函数返回 `NULL`。

例如：若 `s="This is C programming text", t="This is a text for C programming"`，则函数返回 `"programming"`。

- 要求： 1. 函数中不得设置保存单词的存储空间；  
2. 给出函数之前请用文字简要叙述函数的基本思想。

## 2013 年“数据结构与 C 程序设计”(代码 991) 试题参考答案

### 一、单项选择题

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

### 二、填空题

1. 顺序 2.  $O(m)$  3.  $\log_2 k + 1$  4. 235 5.  $2^{n-1}$  6. 该有向图中不存在回路 7. 2.9 8.  $m-1$  9. 插入排序法 10. 9

### 三、综合题

1. 答：(1) 多个堆栈共享一个连续的存储空间，可以充分利用存储空间，只有在整个存储空间都用完时才能产生溢出，其缺点是当一个堆栈溢出时需要向左、右栈查询有无空闲单元。若有，则需要移动相应元素和修改相关的栈底和栈顶指针的位置。当各个堆栈接近溢出时，查询空闲单元、移动元素和修改栈底栈顶指针位置的操作频繁，计算复杂，并且耗费时间。

(2) 每个堆栈仅用一个顺序存储空间时，操作简便。但难以确定初始分配存储空间的大小，空间分配少了，容易产生溢出，空间分配多了，容易造成空间浪费；并且各个堆栈不能共享空间。

(3) 一般情况下，分别建立多个链接堆栈不考虑堆栈的溢出（仅受用户内存空间限制），缺点是堆栈中各元素要通过指针链接，比顺序存储结构多占用存储空间。

2. ( $T \rightarrow lchild == \text{NULL} \& \& T \rightarrow rchild == \text{NULL}$ )

$T \rightarrow lchild$

$T \rightarrow rchild$

3. (由于图表显示限制，此题答案见指定教材(《数据结构教程 第二版》(2012 年 4 月第 7 次印刷)) 第 418 页 8-16 题)

4.

(1). 根据  $a = \text{散列表中存入的元素数}/\text{散列表的长度}$ ，得到表的长度为 18，因此，合适的散列函数应该为  $H(k) = k \bmod 17$ 。

(2). (由于图表显示限制，此题答案见指定教材(《数据结构教程 第二版》(2012 年 4 月第 7 次印刷)) 第 428 页 9-15 题)

### 四、算法设计题

```
SORT(int A[ ], int n)
{
    int i, j, min, max, temp;
    i=1;
    while(i<=n/2){
        min=i;
        max=i;
        for(j=i+1;j<n-i+1;j++){
            if(A[j]<A[min])
                min=j;
            if(A[j]>A[max])
                max=j;
        }
        if(min!=i)
            temp=A[min];
            A[min]=A[i];
            A[i]=temp;
        if(max!=i)
            temp=A[max];
            A[max]=A[i];
            A[i]=temp;
    }
}
```

```

        min=j;
        if(A[j]>A[max])
            max=j;
    } /* 确定某趟排序的最小值元素和最大值元素 */
    if(min!=i){
        temp=A[min]; A[min]=A[i]; A[i]=temp;
    } /* 交换 A[min]与 A[i]的位置 */
    if(max!=n-i+1)
        if(max==i){
            temp=A[min]; A[min]=A[n-i+1]; A[n-i+1]=temp;
        } /* 交换 A[min]与 A[n-i+1]的位置 */
        else{
            temp=A[max]; A[max]=A[n-i+1]; A[n-i+1]=temp;
        } /* 交换 A[max]与 A[n-i+1]的位置 */
    }
    i++;
}
}

```

## 五、填空题

1. break a/q    2. a[n-1]>=a[n-2] FUNC2(a, n-1)    3. (\*(&a+i)+i) (\*(&a+i)+N-i-1)    4. i!=0  
n%10+'0'    5. ch-=30 ch-=26
6. \*(s+i) t++    7. strlen(p)-1 p<q    8. ch & 24    9. 4 &number    10. argv[1],"rb" argv[2],  
"wb"

## 六、简答题

1. 答：通常有下列三种方式：

(1)参数传递方式：函数调用时根据实参传递给形参内容的不同又分为值传递与地址传递两种。

(2) 通过 return 语句传递数据：被调用函数可以通过 return 语句将函数值传递给调用函数。

(3)利用全局变量传递数据。

2. 答：指针可以进行下列三种运算：

(1) 指针加/减一个整数。表示以当前指针所指单元的地址为起点的后或前整数个数据的地址。

(2) 指针减指针。表示两个地址之间的数据个数。(指针加指针为非法运算)

(3) 比较。表示同类型的两个指针所指对象在地址位置上的关系。

3. 答：共用体具有以下三个特征：

(1) 共用体变量的成员共用一块存储空间，共用体变量所占用的字节数等于最长成员所占用的字节数；

(2) 共用体不能在定义时进行初始化；

(3) 共用体中的成员每次只能有一个起作用，当存入新成员时，原来的成员失效，其值被覆盖。

4. 答：使用文件的基本操作一般有下列五个步骤：

(1) 在程序中包含头文件 stdio.h

(2) 定义文件指针。例如：FILE \*fp;

(3) 打开文件，使文件指针与磁盘中的实际存储的数据文件建立关联。例如：

```
fp=fopen("test.txt", "r");
```

(4) 对文件进行读写操作。例如：fread(f, 4, 2, fp);

(5)文件使用完毕后，关闭文件。例如： `fclose(fp);`

## 七、程序设计题

```
#include <stdio.h>
main( )
{ int a[100], i, *p, k=0;
  p=a;
  for(i=0; i<100; i++)
    scanf("%d", p+i); /* 对数组进行数据输入 */
  for(i=1; i<100; i++) /* 找出最小值元素，并记录其位置 */
    if(*(p+k)>*(p+i))
      k=i;
  for(i=k; i<99; i++) /* 删除最小值元素 */
    *(p+i)=*(p+i+1);
  for(i=0; i<99; i++) /* 输出处理后数组各元素 */
    printf("%d", *(p+i));
  printf("\n");
}
```

## 八、程序设计题

函数的基本思想：

从左至右顺序扫描字符串 `s`，逐个找出单词，并记录单词的开始位置与单词的长度；若该单词的长度比已找到的单词更长，则从左至右顺序扫描字符串 `t`；当在字符串 `t` 中找到与在 `s` 中找到的当前最长单词相匹配的单词时，记录单词的开始位置与单词的长度，并回到字符串 `s`，在其中找出下一个更长的单词。如此下去，只至字符串 `s` 扫描结束，最后返回相应结果。

```
#include <stdio.h>
#include <string.h>
char *maxword(char *s, char *t)
{ char res, *temp, chs, cht;
  int i, j, found, maxlen=0;
  while(*s!='\0'){
    while(*s==' ')
      s++; /* 过滤 s 中的空格 */
    for(i=0; s[i]!='&&s[i]!='\0'; i++) /* 确定 s 中单词 */
      if(i>maxlen){
        chs=s[i];
        s[i]='\0';
        temp=t;
        found=0;
        while(*temp!='\0'&&!found){
          while(*temp==' ')
            temp++; /* 过滤 t 中的空格 */
          for(j=0;temp[j]!='&&temp[j]!='\0';j++)
            if(j==i){
```

```
        cht=temp[j];
        temp[j]='\0';
        if(strcmp(s, temp)==0){
            maxlen=i;
            res=s;
            found=1
        }
        temp=cht;
    }
    temp=&temp[j]; /* 回到字符串 t 的某一位置*/
}
s[i]=chs;
}
s=&s[i]; /* 回到字符串 s 的某一位置*/
}
if(maxlen==0)
    return NULL; /* 未找到最长公共单词, 返回 NULL */
else{
    res[maxlen+1]='\0';
    return res; /* 找到最长公共单词, 返回该单词 */
}
}
```

北京航空航天大学 2012 年硕士研究生入学考试试题  
“数据结构与 C 语言程序设计”(科目代码: 991)

一、填空题(本题共 20 分, 每小题各 2 分)

1. 从总体上说, “数据结构”课程主要研究\_\_\_\_\_三个方面的内容。
2. 若对某线性表最常用的操作是在表中插入元素或者删除表中元素, 则对于顺序存储结构\_\_\_\_\_和链式存储结构这两种存储结构而言, 线性表应该采用\_\_\_\_\_。
3. 在长度为  $n$  的非空队列中进行插入或者删除操作的时间复杂度用大  $O$  符号表示为\_\_\_\_\_。
4. 若一棵度为 4 的树中度为 1、2、3 和 4 的结点个数分别为 4、2、1 和 1, 则该树中叶结点的个数为\_\_\_\_\_。
5. 若某二叉树的中序遍历序列为 B,A,F,D,G,C,E, 按层次遍历序列为 A,B,C,D,E,F,G, 则该二叉树的后序遍历序列为\_\_\_\_\_。
6. 将一棵结点总数为  $n$ 、且具有  $m$  个叶结点的树转换为一棵二叉树以后, 该二叉树中右子树为空的结点有\_\_\_\_\_个。
7. 对于图  $G=(V,E)$  与  $G'=(V',E')$ , 若有  $V' \subseteq V$ ,  $E' \subseteq E$ , 则称  $G'$  是  $G$  的\_\_\_\_\_。
8. 在顺序表(6,15,30,37,65,68,70,72,89,99)中采用折半查找法查找元素 37, 与表中进行过比较的元素依次是\_\_\_\_\_。
9. 若已知  $n$  个关键字值具有相同的散列函数值, 并且采用线性探测再散列法处理冲突, 那么, 将这  $n$  个关键字值全部散列到初始为空的地址空间中, 发生散列冲突的次数是\_\_\_\_\_。
10. 若长度为  $n$  的序列  $K=(k_1, k_2, \dots, k_n)$  当且仅当满足  $k_i \leq k_{2i}$  并且  $k_i \leq k_{2i+1}$  ( $1 \leq i \leq n/2$ ) 时, 则称该序列为一个小顶堆积(Heap)。根据该定义, 序列(26,5,77,1,61,11,59,48,15,19)对应的小顶堆积是\_\_\_\_\_。

二、简答题(本题共 20 分, 每小题各 5 分)

1. 如果一个具有 100 个顶点、200 条边的有向图采用邻接矩阵存储, 该邻接矩阵是否是稀疏矩阵? 为什么? (这里我们假设: 当矩阵中非零元素的数目小于整个矩阵总元素的数目的 5% 时认为该矩阵为稀疏矩阵)
2. 一般情况下, 建立散列表时难以避免出现散列冲突, 常用处理散列冲突的方法之一是开放定址法, 该方法的基本思想是什么?
3. 若对序列(2,12,16,88,5,10)按值从小到大进行排序, 前三趟排序的结果分别为:

第 1 趟排序的结果: (2,12,16,5,10,88)  
第 2 趟排序的结果: (2,12,5,10,16,88)  
第 3 趟排序的结果: (2,5,10,12,16,88)

请问: 该结果是采用了选择排序法还是采用了(起)泡排序法得到的? 为什么?

4. 快速排序法的排序过程是递归的。若待排序序列的长度为  $n$ , 则快速排序的最小递归深度与最大递归深度分别是多少?

三、综合题(本题共 20 分, 每小题各 5 分)

1. 若非空双向循环链表中链结点结构为 `struct data { link llink; link rlink; };`, 则依次执行下列 4 条语句的目的是在该链表中由  $q$  指的结点后面插入一个由  $p$  指的结点, 其中 1 条语句有错误, 请找出该语句, 并写出正确的语句。

`p->llink=q; /* 第 1 条语句 */`      `p->rlink=q->rlink; /* 第 2 条语句 */`

- q->rlink=p; /\* 第 3 条语句 \*/      q->rlink->llink=p; /\* 第 4 条语句 \*/
2. 已知某完全二叉树的第 7 层有 10 个叶结点, 请求出该完全二叉树的结点总数的最大值。  
(要求写出结论的求解过程)
3. 证明: 具有  $n$  个顶点的无向图最多有  $n(n-1)/2$  条边。
4. 请分别写出对数据元素序列(80,30,50,10,90,20) 按值从大到小进行选择排序时每一趟的排序结果。

#### 四、算法设计题 (本题 15 分)

已知某具有  $n$  个顶点的有向图采用邻接表方法存储, 其中, 用以存储有向边信息的边结点类型为

```
typedef struct edge{
    int adjvex;           /* 某有向边的终止顶点在顶点结点中的位置 */
    struct edge *next;    /* 指向下一个边结点 */
}ELink;
```

用以存储顶点信息的顶点结点类型为

```
typedef struct ver{
    int indegree;         /* 某顶点的入度 */
    vertype vertex;       /* 某顶点的数据信息 */
    ELink *link;          /* 指向以该顶点为出发点的第一个边结点 */
}VLink;
```

并且  $n$  个顶点结点构成一个数组结构  $G[0..n-1]$ 。请写一个算法, 该算法判断给定的顶点序列  $V[0..n-1]=\{v_1, v_2, v_3, \dots, v_n\}$  是否是该有向图的一个拓扑序列, 若是该有向图的一个拓扑序列, 算法返回 1, 否则, 算法返回 0。

#### 五、单项选择题 (本题共 20 分, 每小题各 2 分)

- 在 C 语言中, 标识符只能由字母、数字和下划线三种字符组成, 并且第一个字符 。  
A. 必须是字母 B. 必须是下划线  
C. 必须是字母或者下划线 D. 可以是字母、数字和下划线之一
- 若整型变量  $x$  的初值为 6, 则计算表达式“ $x+=x-=x*x$ ”之后,  $x$  的值是 。  
A. 50 B. 60 C. -50 D. -60
- 下列 4 个程序段中, 不是无限循环的是 。  
A. `for(b=0,a=1; a>++b; a=k++) k=a;` B. `for( ; ; a++=k) ;`  
C. `while(1) { a++; }` D. `for(k=10; ; k--) total+=k;`
- 说明“`double (*ptr)[N];`”中的标识符 `ptr` 是 。  
A.  $N$  个指向 `double` 类型变量的指针  
B. 指向  $N$  个 `double` 类型变量的函数指针  
C. 一个指向由  $N$  个 `double` 类型元素组成的一维数组的指针  
D. 具有  $N$  个指针元素的一维指针数组, 其每一个元素都只能指向 `double` 类型变量
- 下列 4 个叙述中, 正确的是 。  
A. `char *r="china";` 等价于 `char *r; *r="china";`  
B. `char *ptr="china";` 等价于 `char *ptr; ptr="china";`  
C. `char string[10]={“china”};` 等价于 `char string[10]; string[ ]={“china”};`  
D. `char str[4]={“abc”,temp[4]=“abc”};` 等价于 `char str[4]=temp[4]={“abc”};`
- 在 C 程序中, 语句“`char *func(int x,int y);`”表示 。

- A. 对函数 func 的定义 B. 对函数 func 的调用  
C. 对函数 func 返回值类型的说明 D. 对函数 func 的原型说明  
7. 对于下列程序, 若从键盘上输入: abc def<回车>, 则输出结果是 \_\_\_\_\_。

```
#include <stdio.h>
#include <malloc.h>
main()
{ char *p,*q;
  p=(char *)malloc(sizeof(char)*20);
  q=p;
  scanf("%s%s",p,q);
  printf("%s%s\n",p,q);
}
```

- A. defdef B. abcdef C. abc d D. d d  
8. 当说明一个结构体变量时系统分配给它的内存是 \_\_\_\_\_。

- A. 结构中最后一个成员所需的内存量  
B. 结构中第一个成员所需的内存量  
C. 成员中占内存量最大者所需的容量  
D. 各成员所需内存量的总和

9. 下列程序的输出结果为 \_\_\_\_\_。

```
#define ABC(x) x*x
main()
{ int a, k=3;
  a=++ABC(k+1);
  printf("%d",a);
}
```

- A. 8 B. 9 C. 14 D. 17

10. 若要以 a+方式打开一个已经存在的文件, 则下列叙述中, 正确的是 \_\_\_\_\_。  
A. 文件被打开时, 原有的文件内容不被删除, 位置指针移动到文件的末尾, 可进行添加和读操作  
B. 文件被打开时, 原有的文件内容不被删除, 位置指针移动到文件的开头, 可进行重写和读操作  
C. 文件被打开时, 原有的文件内容被删除, 只能进行写操作  
D. 以上三种说法都不正确

## 六、简答题 (本题共 20 分, 每小题各 5 分)

1. 在 C 语言中, 头文件的作用是什么?
2. 在 C 语言中, #include "filename.h" 和 #include <filename.h> 的区别是什么?
3. 在 C 语言中, 全局变量和局部变量的主要区别是什么?
4. 字符指针、浮点数指针、以及函数指针这三种类型的变量哪个占用的内存最大? 为什么?

## 七、填空题 (本题共 20 分, 每小题各 2 分)

(说明: 由于一些符号无法在本网站显示, 本大题中的填空处用(i)表示第 i 个空 ---- “答疑”)

1. 下列代码的功能包括：定义一个 x 数组，说明一个结构体，同时对变量 t 进行初始化，使得 t 的 a 成员的值为 50， b 成员的值为 x 数组的首地址。

请在空白处(方框内)填入合适的内容，以完成上述功能。

```
int x[5]={1,2,3,4,5};  
struct{ int a,  
        int *b  
    }t{ (1), (2) };
```

2. 下列函数的功能是根据公式  $s=1-1/3+1/5-1/7+ \dots + 1/(2n+1)$  计算 s 的值，其中，n 通过形参传入( $n \geq 0$ )，计算结果通过形参指针传回。请在函数的空白处填入合适的内容，使函数完整。

```
void fun(float *sn,int n)  
{ float s=0,w,f=-1;  
    int i;  
    for(i=0;i<=n;i++){  
        f= (1);  
        w=f/(2);  
        s+=w;  
    }  
    *sn=s;  
}
```

3. 下列程序实现将输入的一个小写字母循环后移 5 个位置后输出。例如，若输入字母‘a’，则输出字母‘f’，若输入字母‘w’，则输出字母‘b’。请在程序的空白处填入合适的内容，使程序完整。

```
#include <stdio.h>  
main()  
{ char c;  
    c=getchar();  
    if(c>='a'&& c<='u')  
        (1);  
    else if(c>='v'&& c<='z')  
        (2);  
    putchar(c);  
}
```

4. 下列自定义函数的功能是实现两个字符串的比较。请在函数的空白处填入合适的内容，使函数完整。

```
int strcmp(char *s,char *t)  
{  
    while(*s && *t && *s== (1 )){  
        s++; t++; }  
    return ( (2) );  
}
```

5. 下列程序的功能是将已经按升序排好序的两个字符串 str1 和 str2 中的字符再按升序归并到字符串 str3 中。请在程序的空白处填入合适的内容，使程序完整。

```

#include <stdio.h>
main( )
{ char str1[ ]=“acegikm”;
  char str2[ ]=“bdfhjlnpq”;
  char str3[ ],*p;
  int i=0,j=0,k=0;
  while(str1[i]!='\0'&& str2[j]!='\0'){
  if(str1[i]<str2[j]) str3[k]=str1[i++];
  else (1);
  k++; }
  str3[k]='\0';
  if( (2) ) p=str2+j;
  else p=str1+i;
  strcat(str3,p);
  puts(str3);
}

```

6. 对于下列 main 函数，经过编译、连接后得到的可执行文件名为 file.exe，并且已知在系统的命令状态下输入命令行“file Beijing Shanghai<回车>”后得到的输出结果是  
Beijing

Shanghai

请在函数的空白处填入合适的内容，使函数完整。

```

main(int argc,char *argv[ ])
{ while( (1) ){
  ++argv;
  printf(“%s\n”, (2) );
  --argc; }
}

```

7. 下列程序的功能是打开两个已存在的文件 file1 和 file2，并将 file2 拼接到 file1 的后面。  
请在程序的空白处填入合适的内容，使程序完整。

```

#include <stdio.h>
int main( )
{ FILE *fp1,*fp2;
  if((fp1=fopen(“file1”,“ (1) ”))==NULL){
    printf(“Cannot open file1!\n”);
    return 0; }
  if((fp2=fopen(“file2”,“ (2) ”))==NULL){
    printf(“Cannot open file2!\n”);
    return 0; }
  while(!feof( (3) ))
    fputc( (4) ,fp1);
  fclose(fp1);
  fclose(fp2);
}

```

8. 设 n>0。下列函数的功能是\_\_\_\_\_。

```
int fun(int n)
{ int count=0;
  while(n){ count++; n=n/10; }
  return count;
}
```

9. 下列程序的功能是\_\_\_\_\_。

```
#include <stdio.h>
#include <string.h>
main()
{ char str[81],*ptr1,*ptr2;
  int n;
  gets(str);
  n=strlen(str);
  ptr1=str;
  ptr2=str+n-1;
  while(ptr1<ptr2){
    if(*ptr1!=*ptr2)
      break;
    else{ ptr1++;
           ptr2--; }
  }
  if(ptr1<ptr2) printf("No!\n");
  else printf("Yes!\n");
}
```

10. 下列程序的功能是\_\_\_\_\_。(提示: ftell(\*FILE)返回 long 类型的文件指针位置)

```
#include <stdio.h>
void main()
{ FILE *fp;
  long position;
  fp=fopen("file.tex","a");
  fprintf(fp,"data");
  position=ftell(fp);
  printf("position=%ld\n",position);
  fclose(fp);
}
```

#### 八、程序设计题 (本题 15 分)

请编写一 C 语言程序, 该程序的功能是确定字符串中首次出现的某字符在串中的位置(即该字符是字符串中的第几个字符), 然后从字符串中删除该字符。要求:

(1) 如果未找到该字符, 程序给出相应信息, 否则, 输出该字符在字符串中首次出现的位置, 删除该字符(注: 不考虑非首次出现的该字符的删除), 并且显示删除前后的字符串。

(2) 通过键盘输入字符串以及被确定的字符

## “数据结构与 C 语言程序设计”（科目代码：991）参考答案

### 一、填空(本题共 20 分, 每小题各 2 分)

1. 从总体上说, “数据结构”课程主要研究数据的 逻辑结构、存储结构和算法 三个方面的内容。
2. 若对某线性表最常用的操作是在表中插入元素或者删除表中元素, 则对于顺序存储结构和链式存储结构这两种存储结构而言, 线性表应该采用 链式存储结构 。
3. 在长度为  $n$  的非空队列中进行插入或者删除操作的时间复杂度用大  $O$  符号表示为  $O(1)$  。
4. 若一棵度为 4 的树中度为 1、2、3 和 4 的结点个数分别为 4、2、1 和 1, 则该树中叶结点的个数为 8 。
5. 若某二叉树的中序遍历序列为 B,A,F,D,G,C,E, 按层次遍历序列为 A,B,C,D,E,F,G, 则该二叉树的后序遍历序列为 B,F,G,D,E,C,A 。
6. 将一棵结点总数为  $n$ 、且具有  $m$  个叶结点的树转换为一棵二叉树以后, 该二叉树中右子树为空的结点有  $n-m+1$  个。
7. 对于图  $G=(V,E)$  与  $G'=(V',E')$ , 若  $V' \subseteq V$ ,  $E' \subseteq E$ , 则称  $G'$  是  $G$  的一个子图 。
8. 在顺序表(6,15,30,37,65,68,70,72,89,99)中采用折半查找法查找元素 37, 与表中进行过比较的元素依次是 65,15,30,37 。
9. 若已知  $n$  个关键字值具有相同的散列函数值, 并且采用线性探测再散列法处理冲突, 那么, 将这  $n$  个关键字值全部散列到初始为空的地址空间中, 发生散列冲突的次数是  $n(n-1)/2$  。
10. 若长度为  $n$  的序列  $K=(k_1, k_2, \dots, k_n)$  当且仅当满足  $k_i \leq k_{2i}$  并且  $k_i \leq k_{2i+1}$  ( $1 \leq i \leq n/2$ ) 时, 则称该序列为一个小顶堆积(Heap)。根据该定义, 序列(26,5,77,1,61,11,59,48,15,19)对应的小顶堆积是 1,5,11,15,19,77,59,48,26,61 。

### 二、简答题(本题共 20 分, 每小题各 5 分)

1. 答: 该邻接矩阵是稀疏矩阵。因为邻接矩阵中一共有 10000 个元素, 只有 200 个非零元素, 非零元素的数目只占整个稀疏矩阵元素总个数的 2%, 按照题目假设, 可以认为该邻接矩阵是稀疏矩阵。

2. 答: 该方法的基本思想是当发生散列冲突时按照下列方法求得后继散列地址:  
$$D_i = H(k) + d_i \bmod m \quad i=1,2,\dots,n \quad (n \leq m-1)$$
其中,  $H(k)$  为散列函数,  $k$  为关键字,  $m$  为散列表长,  $d_i$  为地址增量序列, 可以有以下三种取法: (1)  $d_i = 1, 2, 3, \dots, m-1$  称为线性探测再散列;  
(2)  $d_i = 1^2, -1^2, 2^2, -2^2, \dots, \pm n^2$  ( $n \leq m/2$ ) 称为二次探测再散列;  
(3)  $d_i = \text{伪随机数序列}$ , 称为伪随机探测再散列。

3. 答: 该结果是采用了起泡排序法排序得到的。若选择排序法每一趟排序选择一个最大值元素, 该最大值元素只需要与当前参加排序的最后那个元素交换位置, 而不需要改变其他元素的位置, 显然, 从上述结果可以看出不是如此。上述结果符合起泡排序的规律。

4. 答: 最小递归深度为  $\log_2 n + 1$ , 最大递归深度  $n$ 。

### 三、综合题(本题共 20 分, 每小题各 5 分)

1. 第④条语句有错, 正确的语句是  $p->rlink->llink=p;$
  2. 解: 若完全二叉树的第 7 层有 10 个叶结点, 则有两种情况:
    - ① 10 个叶结点集中在第 7 层的最左边, 此时可求出该二叉树的结点总数为  $(2^6-1)+10=73$ 。
    - ② 该完全二叉树的深度为 8, 10 个叶结点集中在第 7 层的最右边, 此时, 可求出该二叉树的结点总数为  $(2^6-1)+2^7-1+108=235$ 。
- 因此, 根据题意, 该完全二叉树最多有 235 个结点。

3. 证明: 设无向图的边数为  $e$ , 顶点  $v_i$  的度为  $TD(v_i)$ , 根据图的性质, 有关系  
$$2e = \sum TD(v_i) \quad (1 \leq i \leq n)$$

由于每一个顶点最多与图中其他  $n-1$  个顶点直接有关, 即图中每一个顶点的度的最大值为  $n-1$ , 因此, 图中  $n$  个顶点的度的最大值之和为  $n(n-1)$ , 即

$$2e = \sum TD(v_i) = n(n-1) \quad (1 \leq i \leq n)$$

于是, 有  $e=n(n-1)/2$

证毕

#### 4. 解:

(注:每一趟选择最大者放前面)

原 始: 80,30,50,10,90,20

第 1 趟: 90,30,50,10,80,20

第 2 趟: 90,80,50,10,30,20

第 3 趟: 90,80,50,10,30,20

第 4 趟: 90,80,50,30,10,20

第 5 趟: 90,80,50,30,20,10

(注:每一趟选择最小者放后面)

原 始: 80,30,50,10,90,20

第 1 趟: 80,30,50,20,90,10

第 2 趟: 80,30,50,90,20,10

第 3 趟: 80,90,50,30,20,10

第 4 趟: 80,90,50,30,20,10

第 5 趟: 90,80,50,30,20,10

### 四、算法设计题 (本题 15 分)

```
int TOPOTEST(TOPOVLINK G[],verttype V[],int n)
{
    Elink *p;
    int i,k;
    for(i=0;i<n;i++){
        for(k=0;k<n;k++){
            if(G[k].vertex==V[i]) { /* 若顶点 V[i]是 G 中的顶点 */
                if(G[k].indegree!=0) { /* 若顶点 V[i]的入度不为 0 */
                    return 0; /* 序列不是该有向图的拓扑序列 */
                }
                p=G[k].link; /* 若顶点 V[i]的入度为 0 */
                while(p!=NULL){
                    G[p->adjvex].indegree--; /* 相关顶点的入度减 1 */
                    p=p->next; /* p 指向下一个边结点 */
                }
                break; /* 测试序列的下一个顶点 */
            }
        }
    }
}
```

```
    }
    return 1; /* 序列是该有向图的拓扑序列 */
}
```

#### 五、单项选择题（本题共 20 分，每小题各 2 分）

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

#### 六、简答题（本题共 20 分，每小题各 5 分）

1. 答：① 通过头文件来调用库功能。在很多场合，源代码不便(或不准)向用户公布，只向用户提供头文件和二进制的库即可。用户只需要按照头文件中的接口声明来调用库功能，而不必关心接口怎么实现的。编译器会从库中提取相应的代码。

② 头文件能加强类型安全检查。如果某个接口被实现或被使用时，其方式与头文件中的声明不一致，编译器就会指出错误，这一简单的规则能大大减轻程序员调试、改错的负担。

2. 答：#include “filename.h”表明该文件是用户提供的头文件，查找该文件时从当前文件目录开始；#include <filename.h>表明这个文件是一个工程或者标准头文件，查找过程会检查预定义的目录。

3. 答：① 生命周期不同：

  &#8226; 全局变量随主程序创建和创建，随主程序销毁而销毁；

  &#8226; 局部变量在局部函数内部，甚至局部循环体等内部存在，退出就不存在；内存中分配在全局数据区。

② 使用方式不同：

  &#8226; 通过声明后全局变量程序的各个部分都可以用到；

  &#8226; 局部变量只能在局部使用。

4. 答：指针变量也占用内存单元，而且所有指针变量占用内存单元的数量都是相同的，就是说，不管是指向何种对象的指针变量，它们占用内存的字节数都是一样的，并且要足够把程序中所能用到的最大地址表示出来（通常是一个机器字长）。

#### 七、填空题（本题共 20 分，每小题各 2 分）

1. ① 50      ② x
2. ① (-1)\*f ② 2\*i+1
3. ① c=c+5    ② c=c-21
4. ① \*t      ② \*s-\*t
5. ① str3[k]=str2[j++] ② str1[i]=='\0'
6. ① argc>1    ② \*argv
7. ① a+      ② r    ③ fp2      ④ fgetc(fp2)
8. 统计正整数 n 的位数
9. 判断输入的字符串是否为“回文”
10. 以追加方式打开文件“file.txt”后向文件写入“data”，然后查看(输出)文件指针位置。

#### 八、程序设计题（本题 15 分）

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

```

#include <string.h>
int func(char *s,int n,char ch)
{
    int j,k=0;
    s[n]=ch;
    s[n+1]='\0';
    while(s[k]!=ch)
        k++;
    if(k==n)
        return 0; /* 字符串中不存在该字符 */
    else{ /* 字符串中存在该字符 */
        for(j=k;j<n;j++)
            s[j]=s[j+1]; /* 删除首次出现的该字符 */
        s[j-1]='\0';
        return k+1; /* 该字符在字符串中位置 */
    }
}
main()
{
    char s[80],ch;
    int len,position;
    gets(s);
    puts(s); /* 输出删除前的字符串 */
    printf("Input a char"); /* 输入字符串 */
    scanf("%c",&ch); /* 输入字符 */
    len=strlen(s); /* 计算字符串的长度 */
    position=fun(s,len,ch);
    if(position==0)
        printf("Not exit!\n");
    else{
        puts(s); /* 输出删除后的字符串 */
        printf("\nPosition=%d\n",position); /* 输出位置 */
    }
}

```

# 2011 年硕士研究生入学考试

## “数据结构与 C 语言程序设计”(科目代码:991) 试题与答案

一、单项选择题(本题共 20 分, 每小题各 2 分)

10. 下列 4 种排序方法中, 当待排序的序列中元素初始时已经按值有序, 排序所花费的时间反而有可能最多的是\_\_\_\_\_。

  - A. 泡排序
  - B. 谢尔(Shell)排序
  - C. 快速排序
  - D. 堆积(Heap)排序

### 三、简答题(本题共 20 分, 每小题各 5 分)

1. 等概率情况下, 在长度为  $n$  的顺序表中插入和删除一个数据元素分别需要平均移动多少个元素? 移动的元素个数主要取决于哪几个因素?
  2. 在采用循环单链表作为某队列的存储结构时, 可以只设置一个队头指针, 也可以只设置一个队尾指针。请问: 从操作的时间效率考虑, 采用哪种方案更合适? 为什么?
  3. 对于具有  $n$  个顶点、 $e$  条边的稀疏图和稠密图, 就空间性能而言, 采用邻接矩阵存储方法和邻接表存储方法哪一种更合适? 为什么?
  4. 什么是小顶堆积(Heap)? 在小顶堆积中, 值最大的元素可能处在什么位置? (可以借助一棵二叉树描述)

### 三、综合题(本题共 20 分, 每小题各 5 分)

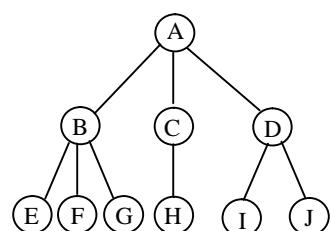
1. 下列算法的功能是删除长度为  $n$  的顺序表  $A$  中重复出现的多余元素，即对于重复出现的元素，表中只保留一个。请在算法的空白处填上必要的内容，使算法完整。

```

void PURGE(ElemType A[ ], int &n)
{
    int i=0,j,k;
    while(i<n){
        j=i+1;                                /* 从第 i+1 个元素开始逐个与第 i 个元素比较 */
        while(j<n)
            if(A[j]==A[i]){
                for(①)
                    A[k-1]=A[k];
                n--;
                /* 修改表的长度 */
            }else
                ②
    }
}

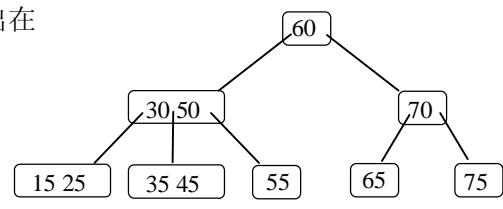
```

2. 请将由题三 2 图给定的树转换为一棵二叉树。  
(只须画出转换后的二叉树)



### 题三 2 图

3. 已知某 3 阶 B-树如题三 3 图所示, 请画出在该 B-树中插入关键字 20 以后得到的 B-树。



### 题三 3 图

4. 请分别写出对数据元素序列(49,38,65,97,76,13,27,49')按从小到大进行谢尔(Shell)排序时每一趟的结果。设排序的间隔数(也称为增量)依次为 4,2,1。

#### 四、算法设计题(本题 15 分)

已知某哈夫曼树采用二叉链表存储, 结点构造为 

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

, 其中, 叶结点的 data 域中已经存放了该叶结点对应的权值。请写一**非递归算法**, 该算法的功能是计算根结点指针为 T 的哈夫曼树的带权路径长度(WPL)。

要求:

1. 用文字简要给出算法的基本思想; (5 分)
2. 根据算法的基本思想写出相应算法。(10 分)

#### 五、程序阅读题(本题共 20 分, 每小题各 2 分)

1. 下列程序的输出结果是\_\_\_\_\_。

```
main( )
{
    char ch='A';
    printf("ch(1)=%d, ch(2)=%c\n", ch, ch+1);
}
```

2. 下列程序段的输出结果是\_\_\_\_\_。

```
k=1; t=3;
do{
    t+=k++;
    if(t%7==0)
        continue;
    else
        ++k;
}while(t<15);
printf("%d", k);
```

3. 下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
main( )
{
    int s[12]={1,2,3,4,4,3,2,1,1,1,2,3}, a[5]={0}, i;
    for(i=0; i<12; i++)
        a[s[i]]++;
    for(i=1; i<5; i++)
        printf("%d", a[i]);
    printf("\n");
}
```

4. 下列程序的输出结果是\_\_\_\_\_。

```
#include <string.h>
main( )
{
    char str1[15]= "good", str2[10]= "morning";
    printf("%d\n", strlen(strcat(str1, str2)));
}
```

5. 下列程序的输出结果是\_\_\_\_\_。

```
main( )
```

```
{    int a[5]={1,2,3,4,5},*p;
    p=a;
    printf("%d\n",*(++p));
}
```

6. 下列程序的输出结果是\_\_\_\_\_。

```
main( )
{    char *s="13579";
    s++;
    printf("%c%c%c%c",*s,*(s+1),*s+1);
}
```

7. 下列程序的输出结果是\_\_\_\_\_。

```
#define MAX(A,B)  (A)>(B)?(A):(B)
#define PRINT(Y)    printf("Y=%d\t",Y)
main( )
{    int a=1,b=2,c=3,d=4,temp;
    temp=MAX(a+b,c+d);
    PRINT(temp);
}
```

8. 下列程序的输出结果是\_\_\_\_\_。

```
int fun(int x,int y)
{    return(x+y);    }

main( )
{
    int a=2,b=5,c=8;
    printf("%d\n",fun(fun(a+c,b),a-c));
}
```

9. 下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
main( )
{
    struct date{
        int year,month,day;
    }today;
    printf("%d\n",sizeof(struct date));
}
```

10. 执行下列程序后，文件 file2.txt 中的内容是\_\_\_\_\_。

```
#include <stdio.h>
main( )
{    FILE *in,*out;
    char *str1="YOU PLAN TO FAIL.";
    char *str2="IF YOU FAIL TO PLAN.";
    if((in=fopen("file1.txt","w"))!=NULL)
        while(*str1!='.')
            fputc(*str1++,in);
    fclose(in);
    if((in=fopen("file1.txt","r"))!=NULL)&&
        ((out=fopen("file2.txt","w"))!=NULL){
        while(!eof(in)){
            fgetc(in);
```

```

        fputc(*str2++,out);
    }
}
fclose(in);
fclose(out);
}

```

## 六、填空题(本题共 20 分, 每小题各 4 分)

1. 对于下列程序, 为了使输出结果为  $t=4$ , 输入量  $x$  和  $y$  应该满足的条件是\_\_\_\_\_。

```

main()
{
    int x,y,s=1,t=1;
    scanf("%d,%d",&x,&y);
    if(x>0)
        s=s+1;
    if(x>y)
        t=s+t;
    else if(x==y)
        t=5;
    else
        t=2*s;
    printf("t=%d",t);
}

```

2. 若已有下列定义, 则表达式  $p->b/n.a$  的值是 ①, 表达式  $p->b/n.a*++p->b$  的值是 ②, 表达式  $(*p).a+p->c$  的值是 ③。

```

struct num{
    int a;
    int b;
    float c;
}n={1,3,5.0};
struct num *p=&n;

```

3. 下列程序段的功能是计算  $1000!$  的末尾含有多少个零。请在程序段的空白处填上必要的内容, 使程序段完整。(提示: 只要计算出  $1000!$  中含有因数 5 的个数即可)

```

for(k=0,i=5;i<=1000;i+=5){
    m=i;
    while(){
        k++;
        m=m/5;
    }
}

```

4. 下列程序的功能是通过指针操作, 找出并输出三个整数中的最小者。请在程序的空白处填上必要的内容, 使程序完整。

```

#include <stdlib.h>
main( )
{
    int *a,*b,*c,num,x,y,z;
    a=&x;
    b=&y;
    c=&z;
    printf("Input a,b,c:");
}

```

```

scanf("%d%d%d",a,b,c);
printf("%d,%d,%d\n",*a,*b,*c);
num=*a;
if(*a>*b)
    (1);
if(num>*c)
    (2);
printf("The minimum=%d\n",num);
}

```

5. 下列程序的功能是先由用户通过键盘输入一个文件名，然后向此文件输入一串字符(假设输入以字符“#”结束)，最后再将当前日期写到文件的尾部。请在程序的空白处填上必要的内容，使程序完整。

```

#include <stdio.h>
main()
{
    char ch,date[20],fname[30];
    FILE *fp;
    printf("Input the file name:");
    scanf("%s",fname);
    if((fp=fopen(1))==NULL){
        printf("Can not open file %s !\n",fname);
        exit(0);
    }
    printf("Input a string:\n");
    while((ch=getchar())!='#')
        fputc(2);
    printf("Enter date:\n");
    scanf("%s",date);
    fprintf(3);
    fclose(fp);
}

```

## 七、程序设计题(本题 15 分)

请编写一 C 语言程序，该程序的功能是先通过键盘输入一个整数 n，然后调用一个**递归函数** fun(int n)计算  $1+2+3+\cdots+n$ ，最后输出计算结果。

## 八、程序设计题(本题 20 分)

请设计一 C 语言函数(注：只须写出函数，不必写出完整程序)，该函数的功能是用尽可能高的时间效率与空间效率将一个 int 类型的数组 A[0..n-1]的所有元素依次循环右移 k 个位置。

例如，对于某数组，当 k=3 时(即把数组所有元素循环右移 3 个位置)，是将

10	20	30	40	50	60	70	转换为	50	60	70	10	20	30	40
----	----	----	----	----	----	----	-----	----	----	----	----	----	----	----

## 参考答案:

### 一、单项选择题

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

### 二、简答题

1. 答: 在等概率情况下, 在长度为  $n$  的顺序表中插入一个数据元素需要平均移动  $n/2$  个元素, 删一个数据元素需要平均移动  $(n-1)/2$  个元素。具体移动的元素个数主要取决于表的长度  $n$  以及插入或删除的位置, 位置越接近  $n$ , 做需要移动的元素就越少。

2. 答: 只设置一个队尾指针更合适。因为对于采用循环单链表作为存储结构的队列而言, 可以通过队尾指针在  $O(1)$  的时间内找到队头指针, 而只设置队头指针要在  $O(n)$  的时间内才能找到队尾指针。因此, 只设置队尾指针, 进队和出队操作的时间复杂度均为  $O(1)$ ; 而只设置队头指针, 出队操作的时间复杂度为  $O(1)$ , 但进队操作的时间复杂度为  $O(n)$ 。

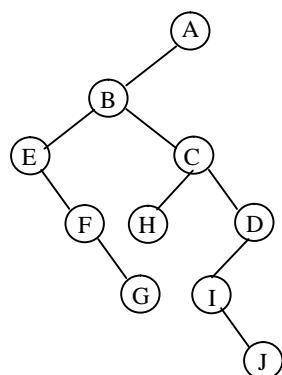
3. 答: 一般情况下, 采用邻接矩阵存储图需要一个一维数组存储顶点的数据信息和一个二维数组(称之为邻接矩阵)存储边或弧的信息, 因此, 空间复杂度为  $O(n^2)$ , 与图中边或弧的数量无关, 可见邻接矩阵适合存储稠密图; 而采用邻接表需要分别将以某顶点为出发点的所有边对应的边结点链接为一个线性链表, 同时用一个一维数组存储图中顶点的数据信息以及指向以该顶点为出发点的第一条边对应的边结点的指针, 因此, 空间复杂度为  $O(n+e)$ , 可见图中边(或弧)数越少需要的存储空间就越少, 因此, 邻接表适合存储稀疏图。

4. 答: 如果借助二叉树来描述, 小顶堆积是一棵完全二叉树, 二叉树中任意分支结点的值均小于或等于其左孩子和右孩子(若右孩子存在)的值。堆积中值最大的元素对应的结点一定是叶结点, 否则, 该结点必定有大于它的孩子结点, 这与小顶堆积的定义相矛盾; 因此, 值最大的元素对应的结点只能作为叶结点出现在二叉树的最下面两层中的一层中。

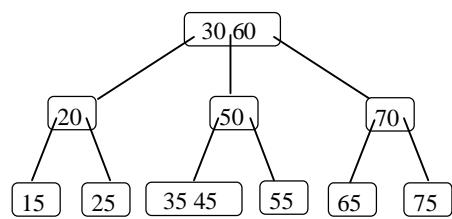
### 三、综合题

1. ①  $k=j; k < n; k++$     ②  $j++;$     ③  $i++;$

2.



3.



4.

趟序	间隔数	各趟排序结果							
		49	38	65	97	76	13	27	49'
初始									
第1趟	4	49	13	27	49'	76	38	65	97
第2趟	2	27	13	49	38	65	49'	76	97
第3趟	1	13	27	38	49	49'	65	76	97

#### 四、算法设计题

(1) 算法的基本思想:

本题宜采用二叉树的后序遍历的非递归算法完成。在遍历过程中, 访问一个叶结点时, 将该叶结点的数据域值(该叶结点的权值)与该叶结点的路径长度(即当前栈顶指针值加 1)相乘, 并进行 WPL 值的累加。遍历结束时便求的该哈夫曼树的 WPL。

(2) 算法:

```
#define MaxNum 50           /* 定义二叉树中结点最大数目 */
int POSTORDER_WPL(BTREE T)
{
    /* T 为二叉树根结点所在链结点的地址 */
    BTREE STACK1[MaxNum],p=T;
    int STACK2[MaxNum],flag,top=-1;
    WPL=0;
    if(T!=NULL)
        do{
            while(p!=NULL){
                STACK1[++top]=p;           /* 当前 p 指结点的地址进栈 */
                STACK2[top]=0;           /* 标志 0 进栈 */
                p=p->lchild;           /* 将 p 移到其左孩子结点 */
            }
            p=STACK1[top];
            flag=STACK2[top--];       /* 退栈 */
            if(flag==0){
                STACK1[++top]=p;           /* 当前 p 指结点的地址再次进栈 */
                STACK2[top]=1;           /* 标志 1 进栈 */
                p=p->rchild;           /* 将 p 移到其右孩子结点 */
            }
            else{
                if(p->lchild==NULL && p->rchild==NULL) /* p 指结点为叶结点 */
                    WPL=WPL+p->data*(top+1);
                p=NULL;
            }
        }while(!(p==NULL && top===-1));
    return WPL;
}
```

#### 五、程序阅读题

- |                     |                       |         |       |
|---------------------|-----------------------|---------|-------|
| 1. ch(1)=65,ch(2)=B | 2. 8                  | 3. 4332 | 4. 11 |
| 5. 2                | 6. 354                | 7. Y=7  | 8. 9  |
| 9. 6                | 10. IF YOU FAIL TO PL |         |       |

#### 六、填空题

- |  |                     |
|--|---------------------|
| 1. 0<x<y                               | 2. ①3 ②12 ③6.0      |
| 3. m%5==0                              | 4. ① num=*b ②num=*c |
| 5. ① fname, “w” ② ch,fp ③ fp,“%s”,date |                     |

#### 七、程序设计题

```
double fun(int n)
{
    double s;
    if(n==1)
        return 1;
    s=n+fun(n-1);
    return s;
}
```

```

main()
{
    double sum;
    int n;
    printf("Input n:");
    scanf("&d",&n);
    sum=fun(n);
    printf("The result is %lf\n",sum);
}

```

## 八、程序设计题

### (1) 算法的基本思想:

根据  $k$  值将数组  $A[0..n-1]$  分成前后两个部分，其中，前一部分为数组的前  $n-k$  个元素，后一部分为数组的后  $k$  个元素；然后先将后一部分中的  $k$  个元素进行逆置(即前后对应位置的元素依次颠倒位置)，接着将前一部分中的  $n-k$  个元素进行逆置，最后将整个数组的所有元素再进行一次逆置，即得到所需要的结果。

### (2) 程序:

```

void MOVE2(int A[ ],int n,int k)
{
    REVERSE(A,n-k,n-1);           /* 逆置数组的后 k 个元素 */
    REVERSE(A,0,n-k-1);           /* 逆置数组的前 n-k 个元素 */
    REVERSE(A,0,n-1);             /* 逆置数组的所有元素 */
}

void REVERSE(int A[ ],int from,int to)
{
    int i, temp;
    for(i=0;i<(to-from+1)/2;i++){ /* 逆置下标为 from 到 to 之间的所有元素 */
        temp=A[from+i];
        A[from+i]=A[to-i];
        A[to-i]=temp;             /* 交换元素 A[from+i] 与 A[to-i] 的位置 */
    }
}

```

效率分析：第 1 次调用 REVERSE 函数的时间复杂度为  $O(k)$ ，第 2 次调用 REVERSE 函数的时间复杂度为  $O(n-k)$ ，第 3 次调用 REVERSE 函数的时间复杂度为  $O(n)$ ，因此，整个程序的时间复杂度为  $O(n)$ 。只用了 1 个数组元素的辅助空间 temp。

建议选用 2010 年 7 月  
第 6 次印刷的书

指定参考书：

1. 《数据结构教程 第二版》唐发根 编著 北京航空航天大学出版社 2005
2. 《C 程序设计 第三版》 谭浩强 编著 清华大学出版社 2005

## 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.  $p->llink=q; p->rlink=q->rlink; q->rlink=p; q->llink=p;$
  - B.  $p->llink=q; p->rlink=q->rlink; q->rlink=p; q->rlink->llink=p;$
  - C.  $p->llink=q; p->rlink=q->rlink; q->rlink=p; p->rlink->llink=p;$
  - D.  $p->llink=q; p->rlink=q->rlink; q->rlink=p; p->llink->llink=p;$
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)

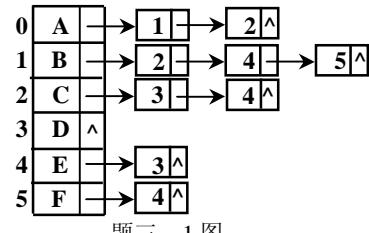
1. 若已知在长度为 n 的顺序表( $a_1, a_2, \dots, a_n$ )的第 i 个位置( $1 \leq i \leq n+1$ )插入一个新的数据元素的概率为  $p_i = \frac{2(n-i+1)}{n(n+1)}$ , 则平均插入一个元素时所需要移动元素次数的期望值(平均次数)是多少?
2. 什么是递归算法? 递归算法在执行 时, 通常需要借助何种数据结构来完成?

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。



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  $H(k)=k \bmod 7$ ，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  $20 + 'x' + i * y$  的值的数据类型为\_\_\_\_\_。  
A. int B. float C. double D. 不确定
  3. 对于关系  $a < b \leq c$ , 对应的 C 语言表达式应该是\_\_\_\_\_。  
A.  $(a < b) \&\& (b \leq c)$  B.  $(a < b) \text{AND} (b \leq c)$   
C.  $(a < b) \leq (c = c)$  D.  $(a < b) \& (b \leq c)$
  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  
C. (*ptr).num
```

- B. `(ptr++)->age`
- 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  $sum=a+aa+aaa+\cdots$ （共 n 项），最后输出计算结果。例如：当 a=5，n=4 时，计算  $sum=5+55+555+5555$ 。

## 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。

程序中可以使用以下 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：
$$\sum_{i=1}^{n+1} pi(n-i+1) = \frac{2}{n(n+1)} \sum_{i=1}^{n+1} (n-i+1)^2 = \frac{2}{n(n+1)} \frac{n(n+1)(2n+1)}{6} = \frac{2n+1}{3}$$

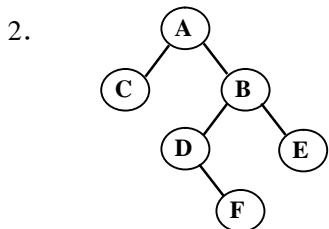
2. 答：一个算法在结束本算法之前，直接或者间接地调用算法自身，这样的算法称为递归算法。递归算法在执行中通常需要借助于堆栈这种数据结构来完成。

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。快速排序法虽然有较好的时间效率，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



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]=='-'){
        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, "Can'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;
    }
    fclose(fp);                  /* 打印文本文件的最后 n 行 */
    /* 关闭文本文件 */
    return 0;
}

```

# 2008 年硕士研究生入学考试 数据结构与 C 语言程序设计(991) 试题与答案

## 一、简答题 (本题共 20 分, 每小题各 5 分)

- 一般情况下, 线性表可以采用哪几种存储结构? 请分别叙述每一种存储结构的构造原理。
- 若 5 个元素 A,B,C,D,E 按此先后次序进入一个初始为空的堆栈, 则在所有可能的出栈序列中, 第一个元素为 C、且第二个元素为 D 的出栈序列是哪几个?
- 设散列函数为  $H(key)=key \bmod 7$ , 散列地址范围为  $[0..7]$ , 采用线性探测再散列法处理冲突。依次将序列(20,11,13,21,34)中各关键字值插入初始为空的散列表以后, 查找关键字值 34, 需要依次与散列表中哪些关键字值进行比较? (符号 MOD 表示求模运算)
- 在设计快速排序法的非递归算法时, 能否不用堆栈保存当前尚未参加排序的子序列的首、尾位置, 而改用其他机制? 如队列。请说明理由。

## 二、综合题 (本题共 20 分, 每小题各 5 分)

- 已知在非空双向循环链表中由指针  $q$  指的那个链结点的前面插入一个由指针  $p$  指的链结点的过程的前 3 个动作对应的赋值语句依次为 “ $p->rlink=q; p->llink=q->llink; q->llink=p;$ ”, 请写出第 4 个动作对应的语句。
- 请证明具有  $n_0$  个叶结点的满二叉树的分支数为  $2(n_0-1)$ 。
- 已知有向图  $G=(V,E)$ , 其中, 顶点的集合为  $V=\{v_1,v_2,v_3,v_4,v_5\}$ , 弧的集合为  $E=\{<v_1,v_2>,<v_1,v_3>,<v_1,v_4>,<v_2,v_5>,<v_3,v_4>,<v_4,v_5>\}$ , 请写出  $G$  的所有拓扑序列。
- 请画出依次插入关键字序列(6,7,10,14,9,2,13,5,4,11)中各关键字值以后的 4 阶 B-树。

## 三、算法设计题 (本题 15 分)

请写一算法, 依次输出通过键盘输入的一组整型数据中的最后  $k$  个元素。

约定: 以  $Ctrl+z$  作为键盘输入的结束, 并假设  $k \leqslant$  输入的数据元素的个数。

限制: 算法中不允许使用数组, 也不允许有计算输入数据个数的过程。

## 四、算法设计题 (本题 20 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  $T$ , 请写一非递归算法, 判断该二叉树是否为二叉排序树。若是二叉排序树, 算法返回 1, 否则, 返回 0。

## 五、单项选择题 (本题共 20 分, 每小题各 2 分)

- 在下面给出的四个选项中, 值为 1 的表达式是\_\_\_\_\_。  
A.  $1-\text{'0'}$       B.  $1\text{'0'}$       C.  $\text{'1'}-0$       D.  $\text{'0'}-\text{'0'}$
- 对于条件表达式  $(E)?(x++):(x--)$ , 其中的表达式  $E$  等价于\_\_\_\_\_。  
A.  $E==0$       B.  $E==1$       C.  $E!=0$       D.  $E!=1$
- 以下语句中存在语法错误的是\_\_\_\_\_。  
A.  $\text{char } *S[]=\{\text{"right?"}\};$       B.  $\text{char } S[20]=\{\text{"right?"}\};$

C. `char *S[6]; S[1]=“right?”;`      D. `char S[6][20]; S[1]=“right?”;`

4. 若变量 `a, b` 已经正确定义并赋值, 以下符合 C 语言语法规则的表达式是\_\_\_\_\_。

A. `a+1=y`      B. `++a,b=a--`      C. `a=a+10=a+b`      D. `double(a)/10`

5. 设已有定义 “`int x; float y;`”, 执行语句 “`scanf(“%2d%f”,&x,&y);`” 时, 若从键盘输入数据 `876 543.0<CR>`, `x` 和 `y` 的值分别是\_\_\_\_\_。(<CR>表示回车)

A. 876 和 543.000000      B. 87 和 543.000000  
C. 87 和 6.000000      D. 76 和 543.000000

6. 下面关于宏的四个描述中, 错误的是\_\_\_\_\_。

A. 进行宏替换时是先求出实参表达式的值, 然后代入形参数, 运算求值  
B. 宏不存在类型问题, 宏名无类型, 它的参数也无类型  
C. 宏替换不占用运行时间  
D. 实际上, 宏替换只不过是字符替代而已

7. 若有以下程序

```
int ADD(int x,int y) { return x+y; }
main()
{
    int k, (*f)(),x=5,y=10;
    f=ADD;
    ...
}
```

则下面给出的四个函数调用语句中, 有错误的是\_\_\_\_\_。

A. `k=(*f)(x,y);`      B. `k=ADD(x,y);`      C. `k=*f(x,y);`      D. `k=f(x,y);`

8. 若变量已经正确定义, 则下面能够正确计算 `f=n!` 的程序段是\_\_\_\_\_。

A. `f=0;`      B. `f=1;`  
`for(i=1;i<=n;i++)`      `for(i=1;i<n;i++)`  
`f*=i;`      `f*=i;`

C. `f=1;`      D. `f=1;`  
`for(i=n;i>1;i++)`      `for(i=n;i>=2;i--)`  
`f*=i;`      `f*=i;`

9. 对于以下程序,

```
#include <string.h>
main()
{
    char s1[]={‘x’,‘y’,‘z’}, s2[10]={‘x’,‘y’,‘z’};
    printf(“%d %d\n”,strlen(s1), strlen(s2));
}
```

下面给出的四个叙述中, 正确的是\_\_\_\_\_。

A. 在给 `s1` 和 `s2` 数组置初值时, 系统会自动添加字符串结束符, 故输出的长度都为 3。  
B. 由于 `s1` 数组中无字符串结束符, 长度不能确定; 但 `s2` 数组中字符串长度为 3。  
C. 由于 `s2` 数组中无字符串结束符, 长度不能确定; 但 `s1` 数组中字符串长度为 3。  
D. 由于 `s1` 和 `s2` 数组中都没有字符串结束符, 长度都不能确定。

10. 若需要打开一个已经存在的非空文件“FILE”，并对其进行修改操作，则正确的打开语句是\_\_\_\_\_。

- A. fp=fopen("FILE", "r");
- B. fp=fopen("FILE", "ab+");
- C. fp=fopen("FILE", "w+");
- D. fp=fopen("FILE", "r+");

## 六、填空题（本题共 20 分，每小题各 2 分）

1. 在 C 语言中，表示逻辑“真”值用\_\_\_\_\_。

2. 若有  $a=1, b=2, c=3$ ，则表达式  $(a < b ? a : b) == c++$  的值是\_\_\_\_\_。

3. 如果已经有如下定义“float a=123.4567;”，那么，执行输出语句“`printf("%f\n", (int)(a*100+0.5)/100.0);`”的输出结果是\_\_\_\_\_。

4. 若有定义“`double x[4][6];`”，则数组 x 中行下标的下限与列下标的上限分别是\_\_\_\_\_。

5. 若有以下宏定义：

```
#define WIDTH 80
#define LENGTH WIDTH+40
```

则执行赋值语句“`x=LENGTH*20;`（设 x 为 int 类型变量）”后，x 的值是\_\_\_\_\_。

6. 若 for 循环语句用以下形式表示：

```
for(表达式 1, 表达式 2, 表达式 3)
    循环体语句
```

则执行语句“`for(i=0;i<4;i++) printf("*");`”时，表达式 3 执行了\_\_\_\_\_次。

7. 执行以下程序段的输出结果是\_\_\_\_\_。

```
int a=3;
do{
    printf("%3d", a-2);
}while(!(--a));
```

8. 若已经定义：

```
struct num{
    int x;
    int y;
} n={1,3};
struct num *p=&n;
```

则表达式 `p->y/n.x*++p->y` 的值是\_\_\_\_\_。

9. 以下程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
int A[3][3]={1,2,3,4,5,6,7,8,9},*p;
main()
{
    p=(int*)malloc(sizeof(int));
    FUN(p,A);
    printf("%d\n", *p);
}
FUN(int *r,int p[][3])
{
    *r=p[1][1];
}
```

10. 若下面给出的程序是输出如右下所示形式的一个方阵，则在程序中的空白处(横线

上)需要填入内容\_\_\_\_\_。

```
main()
{
    int i,j,x;
    for(j=4;j_____;j--){
        for(i=1;i<=4;i++){
            x=(j-1)*4+i;
            printf("%4d",x);
        }
        printf("\n");
    }
}
```

13	14	15	16
9	10	11	12
5	6	7	8
1	2	3	4

## 七、程序设计题（本题 15 分）

请编写一程序，将一字符串的第 k 个字符开始的全部字符复制成为另外一个字符串。

- 要求：1. 将复制过程单独编写为一个函数，并且采用指针完成；  
2. 在主函数中输入字符串和 k 的值，并且在主函数中输出复制结果。

## 八、程序设计题（本题 20 分）

请使用命令行参数形式编写程序，该程序将指定文本文件中所有某个单词的出现均替换为另一个单词，经过替换后的文件信息存放于另一个文本文件中。

设命令行格式为

```
replace oldfile newfile oldword newword
```

其中，replace 为命令名，oldfile、newfile、oldword 和 newword 为命令行参数。

例如，执行命令

```
c>replace file1 file2 old new<CR>
```

的结果是将文本文件 file1 中的所有单词“old”替换为“new”，替换后的文件信息存放于文本文件 file2 中。（<CR>表示回车）

**要求：**程序中必须有命令行的正确性检查。

## 答 案

### 一、简答题

1. 答：一般情况下，线性表可以采用顺序存储结构与链式存储结构。

顺序存储结构是按照表中元素之间的逻辑结构依次将元素存放在一组地址连续的存储空间里，数据元素之间的逻辑结构通过元素的地址直接反映出来。而链式存储结构则是将表中元素依次存放与一组地址任意的存储空间中，数据元素之间的逻辑结构通过指针间接反映出来。因此，在链式存储结构中，除了存储每一个数据元素的信息之外，还要存储指针信息，以反映元素之间的逻辑结构。

2. 答：这样的序列一共有三个，它们分别是 C,D,B,A,E，C,D,E,B,A 和 C,D,B,E,A。

3. 答：20,13,21,34

4. 答：可以不用设置堆栈，而采用其它机制。因为快速排序法经过一次划分(一趟排序)以后，基准元素将当前参加排序的那些元素分成前后两个部分，如果这两个部分的长度都超过 1，下一次划分(排序)先处理其中哪一个部分，后处理哪一个部分，与先后次序无关紧要，而设置何种机制只是决定处理的先后次序而已。

## 二、综合题

1.  $p->llink->rlink=p;$
2. 证明：假设满二叉树的结点总数为  $n$ , 分支数为  $B$ , 有

$$n=n_0+n_2 \quad B=n-1$$

由二叉树的性质(四)得到  $n_2=n_0-1$ , 于是,

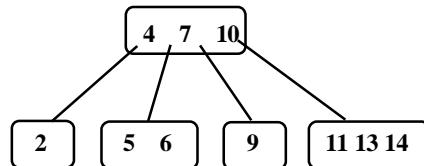
$$n=2n_0-1$$

由此得到结论  $B=2n_0-1-1=2(n_0-1)$

证毕

3.  $v_1, v_2, v_3, v_4, v_5, \dots, v_1, v_3, v_2, v_4, v_5, \dots, v_1, v_3, v_4, v_2, v_5$

4.



## 三、算法设计题

```
void PRINTELE(int k)
{
    LinkkList list, p,r;
    int i,a;
    list=(LinkList)malloc(sizeof(LNode)); /* 建立第一个结点 */
    r=list;
    for(i=1;i<k;i++){
        p=(LinkList)malloc(sizeof(LNode));
        r->link=p;
        r=p;
    }
    r->link=list; /* 建立循环链表 */
    p=list;
    while(scanf("%d",&a)>0){
        p->data=a;
        p=p->link;
    } /* 将数据依次读入链表 */
    r=p;
    do{
        printf("%d",p->data);
        p=p->link;
    }while(p!=r); /* 依次打印所有元素 */
}
```

## 四、算法设计题

```
#define NodeNum 100
int TESTSORTTREE(BTREE T)
{
    BTREE STACK[NodeNum],p=T;
    int top=-1;
    datatype last=MinValue; /* 最小值 */
    if(T!=NULL){
        do{
            while(p!=NULL){
                STACK[++top]=p;
                p=p->lchild;
            }
        }
```

```

p=STACK[top--];
if(p->data<last)
    return 0;           /* 不是二叉排序树 */
last=p->data;
p=p->rchild;
}while(p!=NULL||top!=-1);
}
return 1;           /* 是二叉排序树 */
}

```

## 五、单项选择题

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

## 六、填空题

- |                        |                                |
|------------------------|--------------------------------|
| 1. 非 0 的数字             | 2. 0                           |
| 3. 123.46 或 123.460000 | 4. 0 和 5                       |
| 5. 880                 | 6. 4                           |
| 7. 1 -2                | 8. 12                          |
| 9. 5                   | 10. $\geq 1$ 或 $>0$ 或 $\neq 0$ |

## 七、程序设计题

```

#include <stdio.h>
#include <string.h>
main()
{
    int k;
    char str1[80],str2[80];
    printf("\nInput a string:\n");
    gets(str1);
    printf("Input k:\n");
    scanf("%d",&k);
    if(k<=0||strlen(str1)<k)
        printf("Error input! \n"); /* 若 str1 的长度小于 k */
    else{
        COPYSTR(str1,k,str2); /* 将 str1 的第 k 个字符开始的全部字符复制成 str2 */
        printf("Result is :%s\n",str2);
    }
}

COPYSTR(char *p1,int k,char *p2)
{
    int n;
    while(n<k-1){
        p1++;
        n++;
    }           /* 找到 str1 的第 k 个字符 */
    while(*p1!='\0'){
        *p2=*p1;
        p2++;
        p1++;
    }           /* 实现将 str1 的第 k 个字符开始的全部字符复制成 str2 */
    *p2='\0';
}

```

## 八、程序设计题

```
#include <stdio.h>
#include <string.h>
main(int argc,char *argv[])
{
    char buff[256];
    FILE *fp1,*fp2;
    if(argc<5){
        printf("\nUsage:replace oldfile newfile oldword newword\n");
        exit(0);
    }
    if((fp1=fopen(argv[1], "r"))==NULL){
        printf("\nCan not open file %s!\n",argv[1]);
        exit(1);
    }
    if((fp2=fopen(argv[2], "w"))==NULL){
        printf("\nCan not build file %s!\n",argv[2]);
        exit(1);
    }
    while(fgets(buff,256,fp1)!=NULL){
        while(word_replace(argv[3],argv[4],buff)!=-1);
        fputs(buff,fp2);
    }
    fclose(fp1);
    fclose(fp2);
}

int word_replace(char oldstr[],char newstr[],char str[])
{
    int i,j,k,location=-1;
    char temp1[256],temp2[256];
    for(i=0;str[i] && (location== -1);i++)
        for(j=i,k=0;str[j]==oldstr[k];j++,k++)
            if(!oldstr[k+1])
                location=i;
    if(location!= -1){
        for(i=0;i<location;i++)
            temp1[i]=str[i];
        temp1[i]='\0';
        strcat(temp1,newstr);
        for(i=0,j=location+k;str[j];i++,j++)
            temp2[i]=str[j];
        temp2[i]='\0';
        strcat(temp1,temp2);
        strcpy(str,temp1);
        return location;
    }
    else
        return -1;
}
```

北京航空航天大学 2007 年  
硕士研究生入学考试试题 科目代码: 921  
数据结构与 C 语言程序设计 (共 8 页)

考生注意: 所有答题务必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、单项选择题 (本题共 20 分, 每小题各 2 分)

1. 在非空双向循环链表中由  $q$  所指的链结点前面插入一个由  $p$  指的链结点的过程是依次执行语句  $p->rlink=q$ ;  $p->llink=q->llink$ ;  $q->llink=p$ ; \_\_\_\_。

(横线上方为一条语句)

- A.  $q->rlink->llink=p$ ;      B.  $q->llink->rlink=p$ ;  
C.  $p->rlink->llink=p$ ;      D.  $p->llink->rlink=p$ ;

2. 将长度为  $n$  的单链表链接在长度为  $m$  的单链表后面的算法的时间复杂度采用大O形式表示应该是 \_\_\_\_。

- A.  $O(1)$       B.  $O(n)$       C.  $O(m)$       D.  $O(n+m)$

3. 若将一个  $n$  阶三对角矩阵的所有非零元素按照行序为主序方式依次存放在一个一维数组中, 则该三对角矩阵在一维数组中占用了 \_\_\_\_ 个数组元素。

- A.  $n^2$       B.  $3n-2$       C.  $3n$       D.  $3n+2$

4. 中缀表达式  $A-(B+C/D)\times E$  的后缀形式是 \_\_\_\_。

- A.  $ABCD/+Ex-$       B.  $ABC+D/\times E-$   
C.  $AB-C+D/E\times$       D.  $ABC-+D/E\times$

5. 在一棵度为 4 的树中, 若度分别为 1,2,3,4 的结点的数目依次是 4,3,2,1, 该树中叶结点数目是 \_\_\_\_。

- A. 8      B. 9      C. 10      D. 11

6. 已知某有向图  $G=(V,E)$ , 其中  $V=\{v_1, v_2, v_3, v_4, v_5, v_6\}$ ,  $E=\{<v_1, v_2>, <v_1, v_4>, <v_2, v_6>, <v_3, v_1>, <v_3, v_4>, <v_4, v_5>, <v_5, v_2>, <v_5, v_6>\}$ ,  $G$  的拓扑序列是\_\_\_\_\_。

- A.  $v_1, v_4, v_3, v_5, v_2, v_6$       B.  $v_3, v_4, v_1, v_5, v_2, v_6$   
C.  $v_1, v_3, v_4, v_5, v_2, v_6$       D.  $v_3, v_1, v_4, v_5, v_2, v_6$

7.  $m$  阶 B-树的每个分支结点中最多包含\_\_\_\_\_个关键字值。

- A.  $\lceil m/2 \rceil$       B.  $m-1$       C.  $m$       D.  $m+1$

8. 设初始为空的散列表的地址空间为  $[0..10]$ , 散列函数为  $H(k)=k \bmod 11$ , 采用线性探测再散列法处理冲突, 若依次插入关键字值  $37, 95, 27, 14, 48$ , 则最后一个关键字值  $48$  的插入位置是\_\_\_\_\_。

- A. 4      B. 5      C. 6      D. 7

9. 对具有  $n$  个元素的序列采用二路归并排序法排序, 算法的空间复杂度是\_\_\_\_\_。

- A.  $O(n)$       B.  $O(2n)$       C.  $O(n^2)$       D.  $O(\log_2 n)$

10. 下列名词术语中, 与数据的存储结构无关的是\_\_\_\_\_。

- A. 顺序表      B. 二叉链表      C. 堆栈      D. 线索二叉树

## 二、简答题 (本题共 15 分, 每小题各 5 分)

1. 当线性表的长度基本稳定, 且很少进行插入和删除操作, 但要求以最快的速度存取表中元素时, 该线性表采用何种存储结构比较合适? 为什么?

2. 从构造上说, B-树与 B+树的区别是什么?

3. 若某序列的原始状态为  $(1, 2, 3, 4, 5, 10, 6, 7, 8, 9)$ , 要使得整个排序过程中元素之间的比较次数较少, 在插入排序法与泡排序法这两种排序法中应该选择哪一种方法? 对于该序列, 这两种方法元素之间的比较次数分别是多少?

## 三、(本题共 15 分, 每小题各 5 分)

1. 请写出下列递归算法的功能。

```
typedef struct node{
    datatype data;
    struct node *link;
} *LinkList;

int ALGORISM(LinkList list)
{
    if(list==NULL)
        return 0;
    else
        return 1+ALGORISM(list->link);
}
```

2. 设 a,b,c 三个元素的进栈次序是 a,b,c, 符号 PUSH 与 POP 分别表示对堆栈进行 1 次进栈操作与 1 次出栈操作。

- (1) 请分别写出所有可能的出栈序列以及获得每个出栈序列的操作序列;
- (2) 指出不可能出现的出栈序列。

3. 已知某带权无向连通图采用邻接矩阵存储方法, 邻接矩阵以三元组表形式给出, 不包括主对角线元素在内的下三角形部分元素对应的各三元组分别为 (2,1,7), (3,1,6), (3,2,8), (4,1,9), (4,2,4), (4,3,6), (5,1,∞), (5,2,4), (5,3,∞), (5,4,2)。请画出该带权连通图所有可能的最小生成树。

#### 四、(本题 10 分)

请写一非递归算法, 该算法在按值严格递增排列的顺序表 A[1..n]中采用折半查找方法查找值不大于 item 的最大元素。若表中存在这样的元素, 则算法给出该最大元素在表中的位置, 否则, 给出信息 0。

#### 五、(本题 15 分)

已知具有 n 个结点的非空完全二叉树采用顺序存储结构, 结点的数据信息依次存放于数组 BT[0..n-1]中。请写一算法, 生成该二叉树的二叉链表结构。

#### 六、单项选择题 (本题共 20 分, 每小题各 1 分)

1. 若有语句: char \*s[5];, 以下给出的四个叙述中, 正确的是\_\_\_\_\_。

- A. 定义 `s` 是一个数组，每个数组元素是一个基类型为 `char` 的指针变量  
B. 定义 `s` 是一个指针变量，该变量可以指向一个长度为 5 的字符型数组  
C. 定义 `s` 是一个指针数组，语句中的\*号称为间址运算符  
D. 定义 `s` 是一个指向字符型函数的指针
2. 以下给出的叙述中，错误的是\_\_\_\_\_。  
A. C 语言中的文本文件以 ASCII 码形式存储数据  
B. 语句 `FILE fp;` 定义了一个名为 `fp` 的文件指针  
C. 函数 `fopen()` 的 mode 取值为“w”时，表示可以由文件输出  
D. 函数 `fgets(s,n,fp)` 将从文件中最多读入 `n-1` 个字符
3. 若有 `char a;int b;float c;double d;`，则表达式 `a*b+d-c` 的结果为\_\_\_\_\_类  
型。  
A. `char`      B. `int`      C. `float`      D. `double`
4. 设有定义： `int x=2,y=3,z=4;`，下列表达式中，值为 0 的是\_\_\_\_\_。  
A. `(!x==1)&&(!y==0)`      B. `(x<y)&&!z||1`  
C. `x&&y`      D. `x||(y+y)&&(z-x)`
5. 设有定义： `int a=1,b=2; float c=7;`，下列表达式中，错误的是\_\_\_\_\_。  
A. `a=a>=a`      B. `-a++`      C. `a%int(c)`      D. `a>=c>=b`
6. 以下能够正确定义且赋初值的语句是\_\_\_\_\_。  
A. `int a=b=50;`      B. `float d=d+5.5;`  
C. `double e=12.3E3.5;`      D. `char c=32;`
7. 当把以下 4 个表达式用作 `if` 语句的控制表达式时，其中，\_\_\_\_\_与其他  
3 个表达式的含义不同。  
A. `n%10`      B. `n%10==1`      C. `(n%10)!=0`      D. `!n%10==1`
8. 设有定义： `int a=0,b,*p=&b,*q=&a;`，以下赋值语句中，与 `b=a;` 语句等

价的是\_\_\_\_\_。

- A.  $*p = *q;$       B.  $p = q;$       C.  $*p = \&a;$       D.  $p = *q;$

9. 以下不能正确计算代数式  $\frac{1}{3} \sin^2(\frac{1}{2})$  值的 C 语言表达式是\_\_\_\_\_。

- A.  $1/3\sin(1/2)*\sin(1/2)$       B.  $\sin(0.5)*\sin(0.5)/3$   
C.  $\text{pow}(\sin(0.5),2)/3$       D.  $1/3.0*\text{pow}(\sin(1.0/2),2)$

10. 对于以下程序段, 为了使它不陷入死循环, 从键盘输入的数据应该是\_\_\_\_\_。

```
int m,t=1,s=0;
scanf("%d",&m);
do{
    s=s+t; t=t-2;
}while(t!=m);
```

- A. 任意正偶数      B. 任意负偶数      C. 任意正奇数      D. 任意负奇数

11. 下面给出的定义中, 合法的数组定义是\_\_\_\_\_。

- A.  $\text{int } A[] = \text{"string"};$       B.  $\text{char } A[] = \{0,1,2,3,4,5\};$   
C.  $\text{char } A = \text{"string"};$       D.  $\text{int } A[5] = \{0,1,2,3,4,5\};$

12. 对于以下说明, 给出的 4 种叙述中, 错误的是\_\_\_\_\_。

```
union data{
    int i;
    char c;
    float f;
}
```

- A. a 所占用的内存长度等于成员 f 的长度  
B. a 的地址和它的各成员地址都是同一地址  
C. a 可以作为函数参数  
D. 虽然不能对 a 进行赋值, 但可以在定义 a 时对它进行初始化

13. 执行下列程序段以后, 变量 s 的值是\_\_\_\_\_。

```
int s=0,x=1,y=2,z=3;
s=x<y ? y:x;
s=s>z ? z:s;
```

- A. 0      B. 1      C. 2      D. 3

14. 下列函数 FUN 的类型是\_\_\_\_\_。

```
FUN(float y)
{
    printf("%d\n", y*y);
}
```

- A. void 类型   B. int 类型   C. 与参数 y 的类型相同   D. 无法确定

15. 下列函数 STRCAT 的功能是将 str2 所指的字符串连接到 str1 所指的字符串之后，要实现该功能，程序的空白处应该填上\_\_\_\_\_。

```
char *STRCAT(*str1,*str2)
{
    char *t=str1;
    while(*str1) str1++;
    while( _____ );
    return t;
}
```

- A. str1=str2      B. \*str1++=\*str2++
C. str1++=str2++      D. \*str1++=\*str2

16. 已知字符 A 的 ASCII 码为 65，下列程序运行后输出的结果是\_\_\_\_\_。

```
main()
{
    char x,y;
    x='A'+'5'-'3'; y=x+'6'-'2';
    printf("%d %c",x,y);
}
```

- A. 67 G      B. 67 F      C. 67 E      D. 67 H

17. 下列程序运行后的输出结果是\_\_\_\_\_。

```
main()
{
    int s[10]={1,2,3,4,5,6,7,8,9,10},*p=&s[3],*q=p+2;
    printf("%d\n",*p+*q);
}
```

- A. 6      B. 8      C. 10      D. 16

18. 下列程序运行后的输出结果是\_\_\_\_\_。

```
main()
{
    char r[]={‘1’,‘2’,‘3’},s[]={“123”};
```

```
    printf("%d %d\n", sizeof(r), sizeof(s));  
}
```

A. 4 3      B. 3 4      C. 3 3      D. 4 4

19. 下列程序运行后的输出结果是\_\_\_\_\_。

```
#define f(x) (x*x)  
main()  
{    int a,b;  
    a=f(8)/f(4); b=f(4+4)/f(2+2);  
    printf("%d,%d\n",a,b);  
}
```

A. 64,28      B. 64,64      C. 4,4      D. 4,3

20. 如果要打开一个非空文件 file 进行修改，正确的打开语句是\_\_\_\_\_。

- A. fp=fopen("file","r");      B. fp=fopen("file","ab+");  
C. fp=fopen("file","w+");      D. fp=fopen("file","r+");

#### 七、简答题（本题共 15 分，每小题各 5 分）

1. C 语言是一种结构化程序设计语言。结构化程序设计的基本特点是什么？
2. 在 C 语言中，什么是文件类型指针？通过文件指针访问文件有什么好处？
3. 函数 puts() 与函数 printf() 在功能上有什么相同之处与不同之处？

#### 八、（本题 20 分）

请写一 C 语言程序，该程序对输入的任意正整数 n，产生并且显示相应的 n 阶螺旋式数字方阵。

例如，当 n=5 时，要显示的螺旋式数字方阵如题八图所示。

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

题八图

#### 九、（本题 20 分）

使用命令行参数形式编写 C 语言程序以实现从 2 开始连续输出 n 个素数。

设命令行格式为

outprime n

例如：执行命令

```
c>outprime 10↙
```

的结果是：

```
2,3,5,7,11,13,17,19,23,29
```

要求：程序中必须有命令行的正确性检查。

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\_\_\_\_\_. ( $i \neq j$ )
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  $H(k)=k \bmod 9$ , 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)));`”, 语句中的实参数是\_\_\_\_\_。



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=%d,b=%d",a,b);
}
```

- A. a=2,b=3    B. a=%2,b=%3    C. a=%d,b=%d    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)

- 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？
- 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 fcat.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：

```
fcat file1 file2 file3
```

其中，fcat 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)。

北京航空航天大学 2005 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 8 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、单项选择题 (本题共 20 分, 每小题各 2 分)

1. 算法的时间复杂度取决于\_\_\_\_\_。  
A. 问题的规模 B. 待处理数据的存储结构  
C. 待处理数据的状态 D. 描述算法的语言
2. 与单(向)链表相比较, 双向链表的优点之一是\_\_\_\_\_。  
A. 可以省略头结点指针 B. 可以进行随机访问  
C. 插入、删除操作更简单 D. 顺序访问相邻结点更灵活
3. 在非空双向循环链表中由 q 所指的链结点后面插入一个由 p 指的链结点的动作依次为: \_\_\_\_\_。  
A.  $p->llink=q; p->rlink=q->rlink; q->rlink=p; q->rlink->llink=p;$   
B.  $p->rlink=q->rlink; p->llink=q; q->rlink=p; q->rlink->llink=p;$   
C.  $p->llink=q; p->rlink=q->rlink; q->rlink=p; p->llink->rlink=p;$   
D.  $p->llink=q; p->rlink=q->rlink; q->rlink=p; p->rlink->llink=p;$
4. 堆栈和队列的共同之处在于它们具有相同的\_\_\_\_\_。  
A. 逻辑特性 B. 物理特性 C. 运算方法 D. 元素类型
5. 某堆栈的输入序列为 1,2,3,4, 下面给出的四个序列中, \_\_\_\_\_ 不可能是该堆栈的输出序列。  
A. 1,3,2,4 B. 2,3,4,1 C. 4,3,1,2 D. 3,4,2,1
6. 当结点数目一定时, 具有最小深度的二叉树是\_\_\_\_\_。  
A. 满二叉树 B. 完全二叉树 C. 线索二叉树 D. 二叉排序树

7. 若一棵二叉树有 50 个叶结点，则该二叉树至少有\_\_\_\_\_个结点。  
A. 51      B. 99      C. 100      D. 101
8. 通过拓扑排序可以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
9. 只能在顺序存储结构上进行的查找方法是\_\_\_\_\_。  
A. 顺序查找法    B. 折半查找法    C. 树型查找法    D. 散列查找法
10. 从未排序序列中任意选出一个元素，该元素将当前参加排序的序列分成前后两个部分，前一部分中所有元素都小于等于所选元素，后一部分中所有元素都大于等于所选元素，而所选元素处在排序的最终位置；然后分别对被分成的两个部分中元素个数超过 1 的部分重复上述过程，直至排序结束。这种排序方法称为\_\_\_\_\_排序法。  
A. 选择      B. 插入      C. 快速      D. 二路归并

## 二、填空题（本题共 20 分，每小题各 2 分）

1. 数据的存储结构通常可以有四种，它们分别是\_\_\_\_\_。
2. 当一个算法的时间复杂度与处理问题的规模  $n$  的大小无关时，采用大 O 表示方法应该表示为\_\_\_\_\_。
3. 下面的递归算法的功能是求一个线性链表的长度。

```
typedef struct node{  
    datatype data;  
    struct node *link;  
} *LinkList;  
  
int LENGTH(LinkList list)  
{  
    if(list==NULL)  
        return 0;  
    else  
        return _____;
```

}

4. 下面算法的功能是将两个非空的线性链表连接成为一个线性链表。

```
typedef struct node {  
    datatype data;  
    struct node *link;  
}*LinkList;  
  
void FUN(LinkList lista, LinkList listb)  
{    LinkList p;  
    for( _____ )  
    {  
        ;  
        p->link=listb;  
    }  
}
```

5. 若某非空堆栈采用顺序存储结构, 栈顶指针为 top, 则删除栈顶元素时, top 的变化为\_\_\_\_\_。

6. 完全二叉树、满二叉树、线索二叉树和二叉排序树这四个名称中, 只有\_\_\_\_\_与数据的存储结构有关系。

7. 若有向图采用邻接矩阵存储方法, 则计算第 i 个顶点的入度的方法是\_\_\_\_\_。

8. 将数据元素 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 依次存放于一个一维数组中 (假设数组第一个元素的下标为 1), 然后采用折半查找方法查找元素 15, 被比较过的数组元素的下标依次为\_\_\_\_\_。

9. 假设排序过程中序列的变化情况如下:

初始状态: 50, 72, 28, 39, 81, 15

第一趟后: 15, 72, 28, 39, 81, 50

第二趟后: 15, 28, 72, 39, 81, 50

第三趟后: 15, 28, 39, 72, 81, 50

第四趟后: 15, 28, 39, 50, 81, 72

第五趟后: 15, 28, 39, 50, 72, 81

可以断定，所采用的排序方法是\_\_\_\_\_方法。

10. 根据(大顶)堆积的定义，对于序列(26,5,77,1,61,14,59,15)的堆积为\_\_\_\_\_。

### 三、(本题 10 分)

折半查找的过程可以利用一棵称之为“判定树”的二叉树来描述。请画出在长度为 13 的有序表中进行折半查找对应的判定树。

### 四、(本题 10 分)

若散列函数为  $H(key) = i \bmod 7$ ，其中， $i$  为关键字  $key$  的第一个字母在英文字母表中的序号，并且采用线性探测再散列方法处理冲突。请画出在一个初始状态为空、地址值域为  $[0 \cdots 6]$  的散列表中依次插入下列关键字 MON, TUE, WED, THU, FRI, SAT, SUN 以后的散列表。

### 五、(本题 15 分)

所谓二叉树相似，是指它们具有相同的拓扑结构。假设二叉树采用二叉链表存储结构，链结点构造为 

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

，请写一递归算法，判断根结点指针分别为  $T_1$  与  $T_2$  的两棵二叉树是否相似。若它们相似，算法返回 1，否则返回 0。

### 六、单项选择题(本题共 20 分，每小题各 2 分)

1. 以下正确的叙述是\_\_\_\_\_。

A. 在 C 语言中，语句之间必须用分号“;”分开

B. 若  $a$  是实型变量，C 语言程序中允许赋值  $a=10$ ，因此实型变量中允许存放整型数据

C. 在 C 语言中，无论是整数还是实数，都能够准确无误地表示

D. 在 C 语言中，% 是只能用于整数运算的运算符

2. 若有定义  $int A[4][10], *p, *q[4];$ ，且  $0 \leq i < 4$ ，则错误的赋值是\_\_\_\_\_。

A.  $p=A;$       B.  $q[i]=A[i];$       C.  $p=A[i];$       D.  $q[i]=\&A[2][0];$

3. 在说明一个结构体变量时，系统分配给它的存储空间是\_\_\_\_\_。

- A. 该结构体中第一个成员所需要的存储空间
  - B. 该结构体中最后一个成员所需要的存储空间
  - C. 该结构体中占用最大存储空间的成员所需要的存储空间
  - D. 该结构体中所有成员所需要的存储空间的总和
4. 函数 fgets(str,n,fp) 从文件中读入一个字符串, 以下正确的叙述是\_\_\_\_\_。

- A. 字符串读入后不会自动加入'\0'
- B. fp 是 file 类型的指针
- C. 函数 fgets 将从文件中最多读入 n-1 个字符
- D. 函数 fgets 将从文件中最多读入 n 个字符

5. 若有以下定义和语句, 则输出结果是\_\_\_\_\_。

```
char s[10] = "a book!";
printf("%d", strlen(s));
```

- A. 6
- B. 7
- C. 8
- D. 10

6. 下面的 COMP 函数的功能是按照字典顺序比较两个字符串 s 和 t 的大小。若 s 大于 t, 则返回一个正值; 若 s 等于 t, 则返回 0, 否则返回一个负值。请为程序的空白处选择一个正确的答案。

```
COMP(char *s, char *t)
{
    for( ; *s == *t; _____ )
        if(*s == '\0')
            return 0;
    return(*s - *t);
}
```

- A. s++, t++
- B. t++
- C. s++; t++
- D. s++

7. 下面程序的功能是\_\_\_\_\_。

```
#include <stdio.h>
main()
{
    FILE *fp1, *fp2;
    fp1 = fopen("f1", "r");
```

```

fp2=fopen("f2","w");
while(!feof(fp1))
    putchar(fgetc(fp1));
rewind(fp1);
while(!feof(fp1))
    fputc(fgetc(fp1),fp2);
fclose(fp1);
fclose(fp2);
}

```

- A. 将文本文件 f1 复制到文件 f2 中  
 B. 将二进制文件 f2 复制到文件 f1 中  
 C. 将二进制文件 f1 复制到文件 f2 中  
 D. 分别输出文本文件 f1 和 f2 的信息

8. 下列程序的执行结果是\_\_\_\_\_。

```

#include <stdio.h>
FUN(int a)
{
    static int x=10;
    int y=1;
    x+=a; a++; y++;
    return(x+y+a);
}
main()
{
    int k=3;
    while(k<8)
        printf("%d",FUN(k++));
}

```

- A. 17 24 30 37 45                    B. 18 24 30 37 45  
 C. 19 24 30 37 45                    D. 20 25 31 38 46

9. 下列程序的执行结果是\_\_\_\_\_。

```
typedef struct {
    long x[2];
    int y[4];
    char z[8];
} MYTYPE;
main()
{
    MYTYPE a;
    printf("%d\n", sizeof(a));
}
```

- A. 12      B. 24      C. 28      D. 48

10. 下列程序的输出结果是\_\_\_\_\_。

```
#define ADD(a) (a)+(a)
main()
{
    int x=4,y=6,z=7;
    int w=ADD(x+y)*z;
    printf("%d\n",w);
}
```

- A. 70      B. 140      C. 280      D. 80

### 七、(本题 15 分)

请编写一程序，该程序对于任意输入的正整数  $n$ ，输出  $n$  以内的所有素数。  
( $n \geq 2$ )

### 八、(本题 20 分)

约瑟夫问题(Josephus)可以这样描述： $n$  个人围着圆桌坐一圈，给每个人一个编号，现指定编号为  $k$  的人从 1 开始报数，报到第  $m$  的那个人离开圆桌，然后，又从他的下一个人开始继续从 1 开始报数，仍然是报到第  $m$  的那个人

离开圆桌，如此重复下去，直到圆桌周围剩下一个人为止。

请编写一程序，依次输出各个离开圆桌的人的编号。（n、k 和 m 分别通过键盘输入）。提示：可以采用不带头结点的循环链表解决该问题。

### 九、（本题 20 分）

假设某操作系统有一条命令，该命令的功能是将指定文本文件中的所有小写字母都转换成为大写字母，其他字符格式保持不变。命令格式为：

change filename

其中，change 为命令名；参数 filename 为文本文件名。

请编写实现该命令的程序。要求要进行命令行的正确性检查。

北京航空航天大学  
二〇〇四年硕士试题

题单号: 891

数据结构与程序设计 (共 4 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、单项选择题 (本题共 30 分, 每小题各 2 分)

1. 一个顺序表所占用的存储空间大小与\_\_\_\_\_无关。  
A. 表的长度      B. 元素的存放顺序  
C. 元素的类型      D. 元素中各字段的类型
2. 若长度为  $n$  的非空线性表采用顺序存储结构, 删表的第  $i$  个数据元素,首先需要移动表中\_\_\_\_\_个数据元素。  
\* A.  $n-i$       B.  $n+i$       C.  $n-i+1$       D.  $n-i-1$
3. 在非空双向循环链表中由  $q$  所指的链结点后面插入一个由  $p$  指的链结点的动作是依次执行语句  $p->llink=q; p->rlink=q->rlink; q->rlink=p;$  \_\_\_\_\_。  
(空白处为第四条赋值语句)  
A.  $q->llink=p;$       B.  $q->rlink->llink=p;$   
C.  $p->rlink->llink=q;$       D.  $p->llink->llink=p;$
4. 将一个 20 阶的五对角矩阵中所有非零元素压缩存储到一个一维数组中,该一维数组至少应该有\_\_\_\_\_个数组元素才行。  
A. 90      B. 92      C. 94      D. 96
5. 若 5 个元素的出栈序列为 1, 2, 3, 4, 5, 则进栈序列可能是\_\_\_\_\_。  
A. 2, 4, 3, 1, 5      B. 2, 3, 1, 5, 4  
C. 3, 1, 4, 2, 5      D. 3, 1, 2, 5, 4
6. 若某完全二叉树的深度为  $h$ , 则该完全二叉树中至少有\_\_\_\_\_个结点。  
A.  $2^h$       B.  $2^h-1$       C.  $2^h+1$       D.  $2^{h-1}$
7. 下列名词术语中, 与数据的存储结构有关系的是\_\_\_\_\_。  
A. 完全二叉树      B. 满二叉树      C. 线索二叉树      D. 二叉排序树
8. 在二叉排序树中进行查找的效率与\_\_\_\_\_有关。  
A. 二叉排序树的深度      B. 二叉排序树的结点的个数  
C. 被查找结点的度      D. 二叉排序树的存储结构

9. 若图的邻接矩阵中主对角线上的元素均为 0, 其余元素全为 1, 则可以断定该图一定\_\_\_\_\_。
- A. 是无向图 B. 是有向图  
C. 是完全图 D. 不是带权图
10. 已知某有向图  $G=(V, E)$ , 其中,  $V=\{v_1, v_2, v_3, v_4, v_5, v_6\}$ ,  $E=\{\langle v_1, v_2 \rangle, \langle v_1, v_4 \rangle, \langle v_2, v_4 \rangle, \langle v_3, v_1 \rangle, \langle v_3, v_4 \rangle, \langle v_4, v_5 \rangle, \langle v_5, v_2 \rangle, \langle v_5, v_6 \rangle\}$ ,  $G$  的拓扑序列是\_\_\_\_\_。
- A.  $v_3, v_4, v_4, v_6, v_2, v_6$  B.  $v_3, v_4, v_1, v_3, v_2, v_6$   
C.  $v_1, v_2, v_4, v_5, v_3, v_6$  D.  $v_1, v_4, v_3, v_5, v_2, v_6$
11. 衡量查找算法性能好坏的主要标准是\_\_\_\_\_。
- A. 参加比较的关键字值的多少  
B. 被查找的关键字值在关键字序列中的位置  
C. 关关键字序列中是否存在被查找关键字值  
D. 关关键字值的平均比较次数的多少
12. 若每个元素的查找概率相等, 则在具有  $n$  个记录的顺序文件中采用顺序查找方法的平均查找长度  $ASL=$ \_\_\_\_\_。
- A.  $(n+1)/2$  B.  $n/2$  C.  $(n-1)/2$  D.  $n$
13. 在具有 15 个记录的排序连续顺序文件上采用折半查找方法查找一个文件中不存在的记录, 需要进行\_\_\_\_\_次关键字值的比较。
- A. 0 B. 4 C. 5 D. 15
14. 若序列的原始状态为 1, 2, 3, 4, 5, 10, 6, 7, 8, 9, 要想使得排序过程中元素的比较次数最少, 则应该采用\_\_\_\_\_方法。
- A. 插入排序 B. 选择排序 C. 谢尔排序 D. 泡排序
15. 从未排序序列中任选出一个元素, 该元素将当前参加排序的序列分成前后两个部分, 前一部分中所有元素均小于等于所选元素, 后一部分中所有元素均大于等于所选元素, 而所选元素处在排序的最终位置, 然后分别对被分成的两部分中元素个数超过 1 的部分重复上述过程, 直至排序结束。这种排序方法称为\_\_\_\_\_。
- A. 选择排序法 B. 插入排序法  
C. 快速排序法 D. 堆积排序法

## 二、填空题 (本题共 30 分, 每小题各 3 分)

1. 在具有  $n$  个链结点的非空线性链表的已知位置插入一个链结点的时间复杂度为\_\_\_\_\_。
2. 在 C 语言中, 若符号  $p->link$  出现在表达式中, 则它表示\_\_\_\_\_。
3. 中缀形式的算术表达式  $A+(B-C)/D\times E$  的后缀形式为\_\_\_\_\_。
4. 若二叉树有  $n_0$  个叶结点,  $n_1$  个度为 2 的结点, 则该二叉树至少有\_\_\_\_\_个

结点。

5. 已知二叉树的中序遍历序列为 DAHCBEFG, 后序遍历序列为 DHCGFEBA, 其前序遍历列为\_\_\_\_\_。
6. 若一个非连通的无向图最多有 28 条边, 则该无向图至少有\_\_\_\_\_个顶点。
7. 已知某带权连通无向图采用邻接矩阵存储方法, 邻接矩阵以三元组表形式给出, 不包括主对角线元素在内的下三角形部分元素对应的各三元组分别为  $(2, 1, 7)$ ,  $(3, 1, 6)$ ,  $(3, 2, 8)$ ,  $(4, 1, 9)$ ,  $(4, 2, 4)$ ,  $(4, 3, 6)$ ,  $(5, 1, \infty)$ ,  $(5, 2, 4)$ ,  $(5, 3, \infty)$ ,  $(5, 4, 2)$ 。该连通图的最小生成树的权值之和为\_\_\_\_\_。
8. 在一棵 B+ 树上通常有两个头指针(即查找的入口地址), 其中一个指向 B+ 树的根结点, 另一个指向\_\_\_\_\_。
9. 一个好的散列函数是指\_\_\_\_\_。
10. 若对序列 (tang, deng, an, wang, shi, bai, fang, liu) 采用选择排序法按字典顺序进行排序, 第三趟排序结束时序列的状态是\_\_\_\_\_。

### 三、简答题 (本题共 20 分, 每小题各 10 分)

1. 若一棵树有  $n_1$  个度为 1 的结点, 有  $n_2$  个度为 2 的结点, 有  $n_3$  个度为 3 的结点, …, 有  $n_m$  个度为  $m$  的结点, 该树一共有多少个叶结点(即度为 0 的结点数  $n_0$ )? 请写出结论的推导过程。
2. 建立散列表时, 在散列地址范围与散列函数都分别相同的情况下, 通常采用链地址法比采用开放地址法处理冲突的时间效率要高, 为什么?

### 四、(本题共 15 分, 第 1 小题 8 分, 第 2 小题 7 分)

已知长度为 12 的线性表为

(May, Dec, Nov, Jun, Oct, Aug, Feb, Jul, Jan, Mar, Sept, Apr),

1. 请按照各元素的第一个字母在英文字母表中的先后次序大小建立一棵二叉排序树;
2. 若每个元素的查找概率均等, 则查找此二叉排序树中任意一个元素的平均查找长度是多少?

### 五、(本题 10 分)

折半查找过程可以利用一棵称之为“判定树”的二叉树来描述。请画出在长度为 13 的有序表中进行折半查找对应的判定树。

### 六、(本题 10 分)

已知带权的图采用邻接表存储方法。请用 C 语言分别写出边结点与顶点结点

的类型定义。

#### 七、(本题 15 分)

请用 C 语言编写计算  $s=a^1+a^2+a^3+\dots+a^n$  之值的程序，其中， $a$  为一个浮点数， $n$  为正整数，均通过键盘输入。

#### 八、(本题 20 分)

Unix 操作系统中有一条命令，该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。

请用 C 语言实现该程序。要求：该程序应该具有一定的错误处理能力，如处理非法命令行参数和非法文件名。

# 北航 2002 年程序设计与数据结构考研试题

## 一、简答题 (10')

1. 数据结构课程是计算机专业的基础课还是专业课，或者专业基础课？(2')
2. 学习数据结构课程需要哪些课程作为它的基础（举例两门课程）？若没有这些知识，对学习数据结构课程可能会产生哪些影响？请举例说明（不超过 100 字）。(4')
3. 数据结构课程将为那些课程学习奠定必要的基础？请举例说明哪些课程（举例两门课程）用到了数据结构课程的哪些知识（不超过 100 字）。(4')

## 二、(5')

请推导出结论：具有  $n_0$  个叶结点的哈夫曼树（Huffman）的分支总数为  $2(n_0 - 1)$ 。

## 三、单项选择题 (2' × 15)

1. 线性链表中各链接点之间的地址\_\_\_\_\_。  
A. 必须连续      B. 部分地址必须连续  
C. 不一定连续      D. 连续与否无所谓
2. 在非空线性链表中由 p 所指的链接点后面插入一个由 q 所致的链接点的过程是依次执行动作\_\_\_\_\_。  
A.  $\text{link}(q) \leftarrow p; \text{link}(p) \leftarrow q;$       B.  $\text{link}(q) \leftarrow \text{link}(p); \text{link}(p) \leftarrow q;$   
C.  $\text{link}(q) \leftarrow \text{link}(p); p \leftarrow q;$       D.  $\text{link}(p) \leftarrow q; \text{link}(q) \leftarrow p;$
3. 在非空双向循环链表中由 q 所指的那个链接点前插入一个 p 指的链接点的动作对应的语句依次为  $\text{rlink}(p) \leftarrow q, \text{llink}(p) \leftarrow \text{llink}(q), \text{llink}(q) \leftarrow p, \text{_____}$ 。(空白处为一条赋值语句)  
A.  $\text{rlink}(q) \leftarrow p$       B.  $\text{rlink}(\text{llink}(q)) \leftarrow p$   
C.  $\text{rlink}(\text{llink}(p)) \leftarrow p$       D.  $\text{rlink}(\text{rlink}(p)) \leftarrow p$
4. 在初始为空的堆栈中依次插入元素 f, e, d, c, b, a 以后，连续进行了三次删除操作，此时栈顶元素是\_\_\_\_\_。  
A. c      B. d      C. b      D. e
5. 若某堆栈的输入序列为  $1, 2, 3, \dots, n$ ，输出序列的第 1 个元素为  $n$ ，则第  $i$  个输出元素为\_\_\_\_\_。  
A.  $i$       B.  $n - i$       C.  $n - i + 1$       D. 哪个元素无所谓
6. 求字符串 T 在字符串 S 中首次出现的位置的操作称为\_\_\_\_\_。  
A. 求串的长度      B. 求子串      C. 串的模式匹配      D. 串的连接
7. 若一棵度为 7 的树有 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，该树一共有\_\_\_\_\_个叶结点。  
A. 35      B. 28      C. 77      D. 78
8. 若一棵二叉树有 1001 个结点，且无度为 1 的结点，则叶结点的个数为\_\_\_\_\_。  
A. 498      B. 499      C. 500      D. 501

9. 已知某完全二叉树采用顺序存储结构, 结点数据信息的存放顺序依次为 ABCDEFGH, 该完全二叉树的后序遍历序列为\_\_\_\_\_。

- A. HDEBFGCA      B. HEDBFGCA      C. HDEBAFGC      D. HDEFGBCA

10. 若某带权图为  $G = (V, E)$ , 其中  $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$ ,  $E = \{(v_1, v_2)_5, (v_1, v_3)_6, (v_2, v_5)_3, (v_3, v_5)_6, (v_3, v_4)_3, (v_4, v_5)_3, (v_4, v_7)_1, (v_4, v_8)_4, (v_5, v_6)_4, (v_5, v_7)_2, (v_6, v_{10})_4, (v_7, v_9)_5, (v_8, v_9)_2, (v_9, v_{10})_2\}$  (注: 顶点偶对右下角的数据表示边上的权值), 则  $G$  的关键路径的长度为\_\_\_\_\_。

- A. 19      B. 20      C. 21      D. 22

11. 顺序查找法适合于存储结构为\_\_\_\_\_的线性表。

- A. 顺序存储结构或链式存储结构      B. 散列存储结构  
C. 索引存储结构      D. 压缩存储结构

12. 当  $n$  足够大时, 在按值有序的顺序表中进行折半查找, 当查找概率相等的情况下, 其查找成功的平均查找长度是\_\_\_\_\_。

- A.  $\frac{n+1}{2}$       B.  $\frac{n}{2}$       C.  $\log_2(n+1)-1$       D.  $\log_2(n+1)$

13. 下述命题中, 不成立的应是\_\_\_\_\_。

- A.  $m$  阶 B 树中的每一个分支结点的子树的个数都小于或等于  $m$   
B.  $m$  阶 B 树中的每一个分支结点的子树的个数都大于或等于  $\lceil m/2 \rceil$   
C.  $m$  阶 B 树中的任何一个结点的子树的高度都相等  
D.  $m$  阶 B 树中有  $k$  个子树的分支结点包含  $k-1$  个关键字

14. 已知散列范围为  $[0..9]$ , 散列函数 (哈希函数) 为  $H(key) = key \bmod 9$ , 处理冲突的方法为线性探测再散列法, 依次插入关键字序列 8, 18, 25, 44, 34, 21, 19, 23 后的哈希表为\_\_\_\_\_。

- |    |    |    |    |    |   |    |   |    |   |    |
|----|----|----|----|----|---|----|---|----|---|----|
| A. | 0  | 1  | 2  | 3  | 4 | 5  | 6 | 7  | 8 | 9  |
|    | 18 | 44 | 19 | 21 |   | 23 |   | 25 | 8 | 34 |
- 
- |    |    |    |    |    |   |    |   |    |   |    |
|----|----|----|----|----|---|----|---|----|---|----|
| B. | 0  | 1  | 2  | 3  | 4 | 5  | 6 | 7  | 8 | 9  |
|    | 18 | 34 | 19 | 21 |   | 23 |   | 25 | 8 | 44 |
- 
- |    |    |   |    |    |   |    |    |    |   |    |
|----|----|---|----|----|---|----|----|----|---|----|
| C. | 0  | 1 | 2  | 3  | 4 | 5  | 6  | 7  | 8 | 9  |
|    | 18 |   | 19 | 21 |   | 23 | 34 | 25 | 8 | 44 |
- 
- |    |    |    |   |    |    |    |    |   |   |    |
|----|----|----|---|----|----|----|----|---|---|----|
| D. | 0  | 1  | 2 | 3  | 4  | 5  | 6  | 7 | 8 | 9  |
|    | 19 | 21 |   | 23 | 44 | 25 | 34 |   | 8 | 18 |

15. 在下述的排序方法中, 不属于内排序方法的是\_\_\_\_\_。

- A. 插入排序法      B. 选择排序法      C. 拓扑排序法      D. 归并排序法

#### 四、(15')

请设计一个时间复杂度为  $O(n)$ , 空间复杂度不超过  $O(2)$  的算法, 该算法将数组  $A[0:n-1]$  中所有元素循环右移  $k$  个位置。

#### 五、(15')

已知某二叉树采用广义表形式作为输入, 请写一非递归算法, 建立该二叉树的二叉链表存储结构。设该链接点构造为 

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

, 根结点地址为  $T$ 。

关于采用广义表形式表示二叉树的约定如下:

- (1) 表中的一个字母表示一个结点的数据信息;
- (2) 每个根结点作为由子树构成的表的构成的表的名字放在表的前面;
- (3) 每个结点的左子树与右子树之间用逗号分开; 若只有右子树而无左子树, 则逗号不能省略;
- (4) 整个广义表的末尾由一个特殊符号 @ 作为表的结束标志。

例如: (A(B(D),C(F,E),G))@ 表示某一棵二叉树, 该二叉树的根结点数据信息为 A。其中, 数据信息为 F 的结点只有右子树, 而无左子树。

## 六、(1'×10)

在下面给出的 C 函数实现中的空白处填上适当的内容, 使其完成正确的功能。

函数说明: 函数 **void ftoa(double f, char s[])** 将浮点数 *f* 转换成相应的字符串, 并存放在 *s* 中, 该函数最多只能转换小数点后四位, 如 123.45 将转换成 “123.45”, -123.456789 将转换成 “-123.4567”。

```
void ftoa(double f, char s[])
{
    int i, j, len, c, n;
    double sign;

    if ((sign = f) < 0)
        f = -f;
    n = (int)f;
    i = 0;
    do {
        s[i++] = n%10 + _____;
    } while (_____);
    if (sign < 0)
        _____;
    len = i;
    for (i=0, j=len-1; _____; _____)
    {
        c = s[i];
        _____;
        s[j] = c;
    }
    f -= (int)f;
    s[len++] = _____;
    for (i = 0; i < 4; i++)
    {
        f *= 10;
        s[len++] = _____;
    }
    while (s[len-1] == '0')
        _____;
    s[len] = _____;
}
```

## 七、(15')

命令 tail 用来打印文件中最后 *n* 行。命令格式为:

`tail [-n] filename`

其中:

*-n*: 要打印的行数, 当省略此参数时, *n* 的缺省值为 10。

*filename*: 文件名

例如, 命令 `tail -20 example.txt` 表示打印文件 `example.txt` 的最后 20 行。

请用 C 语言实现该程序, 该程序应具有一定的错误处理能力, 例如能处理非法命令参数和非法文件名。

提示一：使用命令行参数；

提示二：可以使用下面的 C 库函数：

- `int atoi(char *s)` 将数字串转换为相应整数；
- `fopen, fclose, printf, fprintf, exit;`
- `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。

# 北航 2001 年程序设计与数据结构考研试题

## 一、问答题 (10')

一般情况下，线性表可以采用哪几种存储结构？请分别叙述每一种存储结构的构造原理与特点。

## 二、(10')

已知 AOE 网为  $G = (V, E)$ ，其中  $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$ ，  
 $E = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}\}$ ，其中：  
 $a_1 : (v_1, v_2)_5 \quad a_2 : (v_1, v_3)_6 \quad a_3 : (v_2, v_5)_3 \quad a_4 : (v_3, v_4)_3 \quad a_5 : (v_3, v_5)_6$   
 $a_6 : (v_4, v_5)_3 \quad a_7 : (v_4, v_7)_1 \quad a_8 : (v_4, v_8)_4 \quad a_9 : (v_5, v_6)_4 \quad a_{10} : (v_5, v_7)_2$   
 $a_{11} : (v_6, v_{10})_4 \quad a_{12} : (v_7, v_9)_5 \quad a_{13} : (v_8, v_9)_2 \quad a_{14} : (v_9, v_{10})_2$

注：顶点偶对右下角的数字表示边上的权值。

请按上述过程指出所有关键路径：

ee[1:10] : 

--	--	--	--	--	--	--	--	--	--

  
le[1:10] : 

--	--	--	--	--	--	--	--	--	--

  
e[1:14] : 

--	--	--	--	--	--	--	--	--	--	--	--	--	--

  
l[1:14] : 

--	--	--	--	--	--	--	--	--	--	--	--	--	--

其中， $ee[i]$  与  $le[i]$  分别表示事件  $v_i$  的最早发生时间与最晚发生时间； $e[i]$  与  $l[i]$  分别表示活动  $a_i$  的最早开始时间与最晚开始时间。

## 三、(10')

欲建立一文献库，其正文（文献本身）存放在一个双向循环链表的各个链接点中。

1. 为便于链接点的插入、删除操作，以及按题目、发表日期、发表者名称、主题词（假设每文最多给出三个主题词）进行检索，请设计该链表的链接点结构（给出链接点每个域的名称，并说明该域内存放什么信息。注：以下各小题设计链结点结构也这样要求）。画出整个链表结构的示意图。
2. 设计一个三级索引结构，其中第三级索引称为题目索引，示按文献题目构造的稠密索引，通过该级索引并根据给定题目可得到每个文献的存放地址；该级索引按文献学科类分类存放。第二级索引称为中类索引，是题目索引的索引，指出同一中类的文献题目索引的存放位置（例如农林类、气象类，…，古代史类，近代史类，…）。第一级索引称为大类索引，指出同一大类（如：自然科学类、历史类，…）的文献的中类索引的存放位置。请设计每一级索引的结点结构，并画出该索引的整体示意图。
3. 在设计一种三级索引结构，其中第三级索引仍是题目索引（与 2 题所述相同），第二级索引把具有相同主题词的文献题目索引地址组织在一个单链表中。第一级索引称为主题词索引，用文献给出的主题词做关键字组成杂凑表，即该级索引为一个杂凑表，能够指出具有同一主题词的文献题目索引的索引链表的第一个链结点的存储位置。该杂凑表采用链地址法处理冲突。请设计每一级索引的结点结构，并画出该索引的整体示意图。

#### 四、(10')

已知非空线性链表由 list 指出, 链结点的构造 为 

data	link
------	------

, 请写一算法, 将链表中数据域值最小的那个链结点移至链表的最前面。要求: 不得额外申请新的链接点。

#### 五、(5'+10')

已知求两个正整数  $m$  与  $n$  的最大公因子的过程用自然语言可以表述为反复执行如下动作:

第一步: 若  $n$  等于零, 则返回  $m$ ;

第二步: 若  $m$  小于  $n$ , 则  $m$  与  $n$  相互交换;

否则保存  $m$ , 然后将  $n$  送  $m$ , 将保存的  $m$  除以  $n$  的余数送  $n$ 。

(1) 将上述过程用递归函数表达出来 (设求  $x$  除以  $y$  的余数可以用  $x \bmod y$  形式表示)。

(2) 写出求解该递归函数的非递归算法。

#### 六、(10')

函数 `void insert(char *s, char *t, int pos)` 将字符串  $t$  插入到字符串  $s$  中, 插入位置为  $pos$ 。请用 C 语言实现该函数。假设分配给字符串  $s$  的空间足够让字符串  $t$  插入。(说明: 不得使用任何库函数。)

#### 七、(15')

命令 `sgrep` 用来在文件中查找给定字符串, 并输出串所在行及行号。命令格式为:

`sgrep [-i] filename string`

其中:

`-i` 表示查找时大小写无关, 省略时表示大小写相关

`filename` 给定文件名

`string` 所要查找的字符串

用 C 语言实现该程序, 该程序应具有一定的错误处理能力。(提示: 使用命令行参数)

注意: 除文件及 I/O 操作可使用库函数外, 其它不允许使用库函数。

# 北航 2000 年程序设计与数据结构考研试题

## 一、选择题 (2'×10)

- 在非空双向循环链表中  $q$  所指的结点前插入一个由  $p$  所指的链接点的过程依次为:  $rlink(p) \leftarrow q$ ;  $llink(p) \leftarrow llink(q)$ ;  $llink(q) \leftarrow p$ ; \_\_\_\_\_。  
A.  $rlink(q) \leftarrow p$       B.  $rlink(llink(q)) \leftarrow p$   
C.  $rlink(llink(p)) \leftarrow p$       D.  $rlink(rlink(p)) \leftarrow p$
- 若对  $n$  阶对称矩阵  $A$  以行序为主序方式将其下三角形的元素 (包括主对角线上所有元素) 依次存放于一维数组  $B[1:n(n+1)/2]$  中, 则在  $B$  中确定  $a_{ij}$  ( $i < j$ ) 的位置  $k$  的关系为 \_\_\_\_\_。  
A.  $\frac{i(i-1)}{2} + j$       B.  $\frac{j(j-1)}{2} + i$   
C.  $\frac{i(i+1)}{2} + j$       D.  $\frac{j(j+1)}{2} + i$
- 某堆栈的输入序列为 1, 2, 3, 4 下面的四个序列中, \_\_\_\_\_ 不可能是它的输出序列。  
A. 1, 3, 2, 4      B. 2, 3, 4, 1  
C. 3, 4, 2, 1      D. 4, 3, 1, 2
- 深度为  $h$  的满  $m$  叉数的第  $k$  层有 \_\_\_\_\_ 个结点。 $(1 \leq k \leq h)$   
A.  $m^{k-1}$       B.  $m^k - 1$       C.  $m^{h-1}$       D.  $m^h - 1$
- 具有 10 个叶结点的二叉树中有 \_\_\_\_\_ 个度为 2 的结点。  
A. 8      B. 9      C. 10      D. 11
- 要连通具有  $n$  个顶点的有向图, 至少需要 \_\_\_\_\_ 条边。  
A.  $n-1$       B.  $n$       C.  $n+1$       D.  $2n$
- 已知有向图  $G = (V, E)$ , 其中  $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ ,  
 $E = \{< v_1, v_2 >, < v_1, v_3 >, < v_1, v_4 >, < v_2, v_5 >, < v_3, v_5 >, < v_3, v_6 >, < v_4, v_6 >, < v_5, v_7 >, < v_6, v_7 >\}$ ,  $G$  的拓扑序列是 \_\_\_\_\_。  
A.  $v_1, v_3, v_4, v_6, v_2, v_5, v_7$       B.  $v_1, v_3, v_2, v_6, v_4, v_5, v_7$   
C.  $v_1, v_3, v_4, v_5, v_2, v_6, v_7$       D.  $v_1, v_2, v_5, v_3, v_4, v_6, v_7$
- 若查找每个记录的概率均等, 则在具有  $n$  个记录的连续顺序文件中采用顺序查找法查找一个记录, 其平均查找长度 ASL 为 \_\_\_\_\_。  
A.  $\frac{n-1}{2}$       B.  $\frac{n}{2}$       C.  $\frac{n+1}{2}$       D.  $n$
- 下面关于 B 树和 B+ 树的叙述中, 不正确的是 \_\_\_\_\_。  
A. B 树和 B+ 树都是平衡的多分树  
B. B 树和 B+ 树都可用于文件的索引结构  
C. B 树和 B+ 树都能有效地支持随机检索  
D. B 树和 B+ 树都能有效地支持顺序检索
- 下面给出的四种排序方法中, 排序过程中的比较次数与排序方法无关的是 \_\_\_\_\_。  
A. 选择排序法      B. 插入排序法  
C. 快速排序法      D. 堆积排序法

## 二、(10')

有实现同一功能的两个算法  $A_1$  和  $A_2$ ，其中  $A_1$  的时间复杂度为  $T_1 = O(2^n)$ ， $A_2$  的时间复杂度为  $T_2 = O(n^2)$ ，仅就时间复杂度而言，请具体分析这两个算法哪一个好。

## 三、(5'+10'+10'+5')

为建立一个具有  $n$  份档案的档案库需要设计如下数据结构：所有档案存储在一个动态存储的双向循环链表中，每份档案占用一个地址连续的存储块成为该链表中的一个结点，整个链表为一个链接顺序文件，取名为 *dossier*（档案），同时分别建立两个索引，其中一个为稠密索引，取名为 *dense*，另一个是表长为  $m$  的杂凑表索引，取名为 *bucket*，该杂凑表采用链地址法处理冲突。上述两种索引中都分别存储在每一份档案的存储地址。

- (1) 请分别画出 *dossier*、*dense*、*bucket* 的结构示意图。
- (2) 分别设计出 *dossier*、*dense*、*bucket* 的数据结点的结构，即为了满足档案的插入、删除、查找的操作，每个结点必要的数据项的名称及其作用。
- (3) 针对上述结构，用简明的文字分别说明所有可能的查找方法（查找路径）。
- (4) 分别给出每一种查找方法在查找成功时的平均查找长度。

## 四、(10')

已知  $num$  为无符号十进制整数，请写一非递归算法，该算法依次输出  $num$  对应的  $r$  进制的各位数字。要求算法中用到的堆栈采用线性链表存储结构。 $(1 < r < 10)$

## 五、(10')

已知长度为  $n$  的线性表  $A$  采用顺序存储结构，请写一时间复杂度为  $O(n)$ 、空间复杂度为  $O(1)$  的算法，该算法删除线性表中所有值为 *item* 的数据元素。 $(O(1)$  表示算法的辅助空间为常量)

## 六、(10'×2)

设有一集合，其成员为任意类型的数据元素，基本操作为插入、删除和成员测试。若为该集合设计一个集合类型，则

- (1) 该集合可以采用哪几种存储结构？就存储空间开销以及操作而言，分别说明每种存储结构的特点。
- (2) 分别写出上述三种操作在你所确定的一种存储结构上的具体体现（算法）。

# 北航 1999 年程序设计与数据结构考研试题

## 一、单项选择题（20 分，每小题 2 分）

1. 若长度为  $n$  的线性表采用顺序存储结构, 在其第  $i$  个位置插入一个新元素的算法的时间复杂度为 ( )。 $(1 \leq i \leq n+1)$   
A.  $O(0)$       B.  $O(1)$       C.  $O(n)$       D.  $O(n^2)$
2. 若在线性表中采用折半查找法查找元素, 该线性表应该 ( )。  
A. 元素按值有序  
B. 采用顺序存储结构  
C. 元素按值有序, 且采用顺序存储结构  
D. 元素按值有序, 且采用链式存储结构
3. 已知一算术表达式的中缀形式为  $A + B \times C - D / E$ , 后缀形式为  $ABC \times + DE / -$ , 其前缀形式为 ( )。  
A.  $-A + B \times C / DE$       B.  $-A + B \times CD / E$       C.  $- + \times ABC / DE$       D.  $- + A \times BC / DE$
4. 若二叉树采用二叉链表存储结构, 要交换其所有分支结点左右子树的位置, 利用 ( ) 遍历方法最合适。  
A. 前序      B. 中序      C. 后序      D. 按层次
5. 利用逐点插入法建立序列  $(50, 72, 43, 85, 75, 20, 35, 45, 65, 30)$  对应的二叉排序树以后, 查找元素 35 要进行 ( ) 次元素间的比较。  
A. 4      B. 5      C. 7      D. 10
6. 对二叉排序树进行 ( ) 遍历, 可以得到该二叉树所有结点构成的排序序列。  
A. 前序      B. 中序      C. 后序      D. 按层次
7. 具有  $n$  个顶点的有向图最多有 ( ) 条边。  
A.  $n$       B.  $n(n-1)$       C.  $n(n+1)$       D.  $n^2$
8. 从未排序序列中依次取出一个元素与已排序序列中的元素依次进行比较, 然后将其放在已排序序列的合适位置, 该排序方法称为 ( ) 排序法。  
A. 插入      B. 选择      C. 希尔      D. 二路归并
9. 排序趟数与序列的原始状态有关的排序方法是 ( ) 排序法。  
A. 插入      B. 选择      C. 冒泡      D. 快速
10. 下面给出的四种排序法中, ( ) 排序法是不稳定性排序法。  
A. 插入      B. 泡      C. 二路归并      D. 堆积

## 二、(10 分)

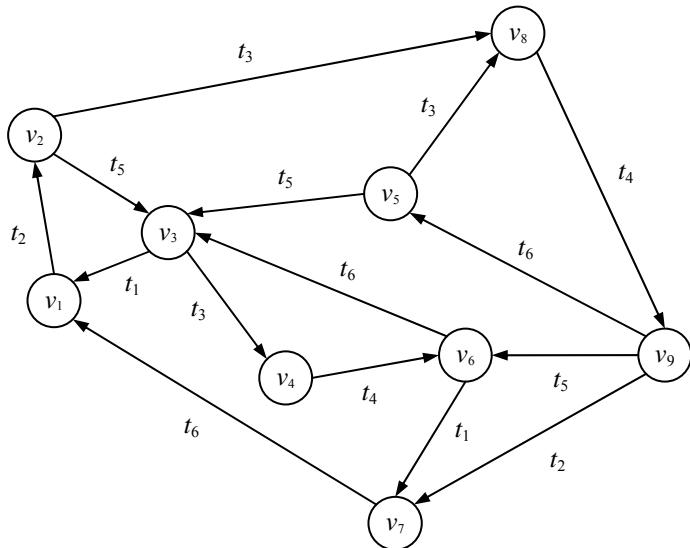
已知  $n$  阶下三角矩阵  $A$  (即当  $i < j$  时, 有  $a_{ij} < 0$ ), 按照压缩存储的思想, 可以将其主对角线以下所有元素 (包括主对角线上元素) 依次存放于一维数组  $B$  中。请写出从第一列开始采用列序为主序分配方式时在  $B$  中确定元素  $a_{ij}$  的存放位置的公式。

### 三、(20分)

已知在某并发处理系统的 Petri 网基础上建立的可达图如图 1 所示。图中每个顶点表示系统运行中的一种状态，有向边（弧）表示事件（用  $t$  表示），例如有向边  $(v_i, v_j)$  表示系统在状态  $v_i$  时出现该事件将引发系统由状态  $v_i$  到状态  $v_j$ 。

- (1) 请分别给出该可达图的邻接表、邻接矩阵以及邻接矩阵的三元组形式，要求每种存储结构能够表达出该可达图的全部信息，并分别对这三种形式中每个部分的含义（物理意义）予以简要说明。

(2) 若假设每个域（包括指针域）的长度为一个字节，请分别计算出这三种结构所占用的空间大小。



四、(本题 10 分, 每小题 5 分)

在排序连续顺序文件中采用折半查找方法查找记录存在与否的过程可以借助于一棵平衡二叉树（也称判定树）来模拟，树中结点的值为记录在文件中的位置序号。

- (1) 若文件中有 19 个记录, 请画出这棵判定树;  
(2) 当文件中有  $n$  个记录时, 求出该判定树的深度。

## 五、（10分）

已知非空线性链表第一个节点由 `list` 指出, 请写一算法, 交换 `p` 所指的节点与其下一个节点在链表中的位置 (设 `p` 指向的不是链表最后那个节点)。

## 六、（15分）

已知 Ackermann 函数定义如下：

$$\text{Ack}(m, n) = \begin{cases} n+1 & m = 0 \\ \text{Ack}(m-1, 1) & m \neq 0, n = 0 \\ \text{Ack}(m-1, \text{Ack}(m, n-1)) & m \neq 0, n \neq 0 \end{cases}$$

- (1) 写出  $\text{Ack}(2, 1)$  的计算过程。
  - (2) 写出计算  $\text{Ack}(m, n)$  的非递归算法。

## 七、（15分）

已知深度为  $h$  的二叉树采用顺序存储结构已存放于数组  $BT[1:2h-1]$  中, 请写一非递归算法, 产生该二叉树的二叉链表结构。设二叉链表中链结点的构造为 

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

, 根结点所在链结点的指针由  $T$  给出。

# 北京航空航天大学

## 一九九八年招收研究生

题单号: 461

数据结构试题(共 5 页) 本试卷共六大题。写算法的语言可以用类 PASCAL 语言, 也可以用某一种程序设计语言, 但不得采用生僻古怪的表示方法。

### 一、(本题共 40 分, 每小题 4 分)

简要回答下列各题:

1. 对一个算法进行分析的前提条件是什么? 主要从哪几个大的方面对一个算法进行分析?
2. 若较频繁地对一个线性表进行插入和删除操作, 该线性表宜采用何种存储结构? 为什么?
3. 在非空双向循环链表中  $q$  所指的结点后面插入  $p$  所指的结点的过程已经依次进行了以下三步:  $llink(p) \leftarrow q; rlink(p) \leftarrow rlink(q); rlink(q) \leftarrow p$ ; 第四步应是什么动作? (写出该动作对应的语句)。
6. 在一个算法中需要建立多个堆栈时可以选用下列三种方案之一, 试问: 这三种方案之间相比较各有什么优缺点?
  - (1) 分别用多个顺序存储空间建立多个独立的堆栈;
  - (2) 多个堆栈共享一个顺序存储空间;
  - (3) 分别建立多个独立的链接堆栈。
7. 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唯一的因素有哪些?
8. 若对一个线性表进行折半查找, 该线性表须满足什么条件?
9. 什么是哈希 (Hash) 表? 如何在哈希表中进行查找?
10. 对于堆积排序法, 快速排序法和归并排序法, 若仅从节省存储空间考虑, 则应该首先选取其中哪种方法? 其次选取哪种方法? 若仅考虑排序结果的稳定性, 则应该选取其中哪种方法? 若仅从平均情况下排序最快这一点考虑, 则应该选取其中哪种方法?

### 五、(本题 15)

已知不带头结点的线性链表 list，链表中结点构造为  

data	link
------	------

，其中 data 为数据域，link 为指针域。请写一算法，将该链表按结点数据域的值的大小从小到大重新链接。要求链接过程中不得使用除该链表以外的任何链结点空间。

### 六、(本题 15 分)

已知一具有  $n$  个结点的二叉树的中序遍历序列与后序遍历序列分别存放于数组 IN[1:n] 和 POST[1:n] 中，(设该二叉树各结点的数据值均不相同)。请写一建立该二叉树的二叉链表结构的非递归算法。设二叉链表的链结点构造为  

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

，其中 data 为数据域，lchild 与 rchild 分别为指向该结点左、右孩子的指针域 (当孩子结点不存在时，相应指针域值为空，用 nil 表示)。

## 说明：

本资料有代表性地搜集和选择了北京航空航天大学软件学院 2008 年 3 月和 9 月、2009 年 6 月和 12 月以及 2010 年 4 月和 7 月自主招收软件工程硕士研究生入学考试的 6 份专业课程试题，并附带相应的参考答案，仅供有志于报考北航软件学院工程硕士研究生的考生复习和参考。

# 北京航空航天大学软件学院 2008 年 3 月自主招收软件工程硕士研究生 专业课试题

## 第一部分 数据结构

### 一、简答题

1. 我们通常采用大 O 形式来表示算法的时间复杂度。例如，在一个长度为 n 的顺序表中顺序查找一个数据元素的过程的时间复杂度为  $O(n)$ ，其中，n 表示问题的规模。那么， $O(1)$  表示什么？请举出一个例子加以说明。
2. 若 5 个元素 A,B,C,D,E 按此先后次序进入一个初始为空的堆栈，那么，在所有可能的出栈序列中，第一个元素为 C、且第二个元素为 D 的出栈序列有哪些？（写出结论即可）
3. 具有 n 个结点的非空满二叉树有多少叶结点？（要求写出结论的推导过程）
4. 若已知有向图  $G=(V,E)$ ，其中，顶点的集合为  $V=\{v_1, v_2, v_3, v_4, v_5\}$ ，弧的集合为  $E=\{<v_1, v_2>, <v_1, v_3>, <v_1, v_4>, <v_2, v_5>, <v_3, v_4>, <v_4, v_5>\}$ ，则  $G$  的拓扑序列有哪些？（写出结论即可）
5. 有人说，采用折半查找法一定比采用顺序查找法的时间效率高，你认为如何？请说明你的理由。

### 二、综合题

1. 简要列出影响一个算法时间效率的主要因素，并指出其中与算法本身直接有关的因素。

2. 请写出下列递归算法的功能。

```
typedef struct node{
    datatype data;
    struct node *link;
} *LinkList;
int ALGORISM(LinkList list)
{
    if(list==NULL)
        return 0;
    else
        return 1+ALGORISM(list->link);
}
```

3. 已知长度为 12 的线性表 (Nov,Dec,Jul,Feb,Oct,Sept,Aug,Apr,May,Jun,Jan,Mar)，请依次按照表中各数据元素的第一个字母在英文字母表中的先后顺序构造一棵二叉排序树。

4. 证明：具有  $n$  个顶点的无向图最多有  $n \times (n-1)/2$  条边。(即写出结论的推导过程)
  5. 请按照(大顶)堆积的定义写出对已知序列 (26,5,77,1,61,11) 进行堆积排序时第 1 趟排序结束时刻序列的状态。

### 三、算法设计题

已知长度为  $n$  的非空顺序表  $A[0..n-1]$ ，请写一算法，该算法删除表中重复出现的数据元素。

#### 四、算法设计题

已知对二叉排序树进行中序遍历可以得到该二叉树所有结点组成的按值从小到大排列的中序序列。若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，请写一非递归算法，判断该二叉树是否为二叉排序树。若是二叉排序树，算法返回 1，否则，算法返回 0。

第二部分 C 语言程序设计

## 五、单项选择题

1. 在下面给出的四个选择中, 合法的实型常数是\_\_\_\_\_。  
A. 5E2.0                    B. 2E0                    C. E-3                    D. 1.3E
  2. 在 C 语言中, 5 种基本数据类型的存储空间长度的排列顺序是\_\_\_\_\_。  
A. char<int<long int<=float<double            B. char=int<long int<=float<double  
C. char<int<long int<float=double                D. char=int=long int<=float<double
  3. 若变量 a,b,c 被定义为 int 类型, 要通过键盘分别给 a,b,c 输入数据, 则正确的输入语句是\_\_\_\_\_。  
A. INPUT a,b,c;                    B. read(“%d%d%d”,&a,&b,&c);  
C. scanf(“%d%d%d”,a,b,c);        D. scanf(“%d%d%d”,&a,&b,&c);
  4. 当接收用户输入的含空格的字符串时, 应该使用的函数是\_\_\_\_\_。  
A. scanf()                    B. gets()                    C. getchar()                    D. getc()
  5. 假设变量 e 的类型为整型, 比较 “if(e!=0);” 与 “if(e);” 两条语句, 下面给出的四个选择中, 正确的答案是\_\_\_\_\_。  
A. 两者作用相反                    B. 两者作用不同  
C. 两者作用相同                    D. if(k)语法错误
  6. 以下四个关于 C 语言的叙述中, 错误的是\_\_\_\_\_。  
A. 可以用 while 语句实现的循环一定可以用 for 语句实现  
B. 可以用 for 语句实现的循环一定可以用 while 语句实现  
C. 可以用 do-while 语句实现的循环一定可以用 while 语句实现  
D. do-while 语句与 while 语句的区别在于关键字 “while” 出现的位置不同
  7. 下面给出的四个关于函数的隐含存储类别的叙述中, 正确的是\_\_\_\_\_。  
A. 在 C 语言中, 函数的隐含存储类别是 auto  
B. 在 C 语言中, 函数的隐含存储类别是 static  
C. 在 C 语言中, 函数的隐含存储类别是 extern  
D. 在 C 语言中, 函数的隐含存储类别不存在

8. 下面给出的四个关于函数定义形式中，正确的是\_\_\_\_\_。  
A. double FUN(int x,int y);      B. double FUN(int x,int y)  
C. double FUN(int x;int y);      D. double FUN(int x,y)

9. 以下能对二维数组 A 进行正确初始化的语句是\_\_\_\_\_。  
A. int a[][3]={{1,2,3},{4,5,6}};      B. int a[2][]={{1,0,1},{5,2,3}};  
C. int a[2][4]={{1,2,3},{4,5},{6}};      D. int a[][3]={{1,0,1},{},{1,1}};

10. 若有定义 “float a[][3]={0,3,8,0,9,0};”，则 a[1][1]的值是\_\_\_\_\_。  
A. 3      B. 0      C. 9      D. 8

11. 定义 “double \*p[6];” 的含义是\_\_\_\_\_。  
A. p 是一个指向 double 类型变量的指针      B. p 是 double 类型数组  
C. p 是指针数组      D. p 是数组指针

12. 若有说明 “int a[10]={1,2,3,4,5,6,7,8,9,10}, \*p=a;”，则数值为 9 的表达式是\_\_\_\_\_。  
A. \*p+9      B. \*(p+8)      C. \*p+=9      D. p+8

13. 以下不能进行字符串赋初值的语句是\_\_\_\_\_。  
A. char \*str=“good!”;      B. char str[]=“good!”;  
C. char str[5]={'g','o','o','d'};      D. char str[5]=“good!”;

14. 若已有以下定义和语句：

```
#include <stdio.h>
int x=4,y=3,*p,*q,*s;
p=&x; q=&y; s=q; q=NULL;
```

则下面分别给出的四条语句中，错误的是\_\_\_\_\_。

- A. \*q=0;      B. s=p;      C. \*p=x;      D. \*p=\*s;

15. 若有下列函数定义：

```
setw(int *x,int m,int n,int data)
{
    int k;
    for(k=0;k<m*n;k++){
        *x=data; x++;
    }
}
```

则调用此函数的正确写法是(假设变量的说明为 int a[50];)\_\_\_\_\_。

- A. setw(\*a,5,8,1);      B. setw(&a,5,8,1);  
C. setw((int\*)a,5,8,1);      D. setw(a,5,8,1);

16. 若有以下说明和语句：

```
struct student{
    int age;
    int num;
}std, *p;
p=&std;
```

则下面对该结构体变量 std 中成员 age 的引用方式错误的是\_\_\_\_\_。

- A. std.age      B. \*p.age      C. (\*p).age      D. p->age

17. 在宏定义 “#define MAX 100” 中，用宏名代替一个\_\_\_\_\_。

- A. 整数      B. 长整数      C. 字符串      D. 常量

18. C 语言可以处理的文件类型是\_\_\_\_\_。

- A. 文本文件和数据文件      B. 二进制文件和数据文件  
 C. 数据代码文件      D. 文本文件和二进制文件
- 19.** 当顺利地执行了文件关闭操作时, 函数 `fclose` 的返回值是\_\_\_\_\_。  
 A. -1      B. TURE      C. 0      D. 1
- 20.** 若需要打开一个已经存在的非空文件“FILE”, 并对其进行修改, 正确的打开语句是\_\_\_\_\_。  
 A. `fp=fopen(“FILE”, “r”);`      B. `fp=fopen(“FILE”, “r”);`  
 C. `fp=fopen(“FILE”, “ab”);`      D. `fp=fopen(“FILE”, “w”);`

## 六、填空题

- 1.** 执行下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
main()
{   int x=10;
    do{
        x--;
    }while(--x);
    printf("%d\n",x--);
}
```

- 2.** 执行下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
#include <string.h>
main()
{   char a[80]=“AB”, b[80]=“LMNP”;
    int i=0;
    strcat(a,b);
    while(a[i++]!=‘\0’)
        b[i]=a[i];
    puts(b);
}
```

- 3.** 下列程序的运行结果是\_\_\_\_\_。

```
#include <stdio.h>
#include <string.h>
FUN(char *w,int n)
{   char temp,*s1,*s2;
    s1=w;
    s2=w+n-1;
    while(s1<s2){
        temp=*s1++;
        *s1=*s2--;
        *s2=temp;
    }
}
main()
{   char *p;
    p=“1234567”;
    FUN(p,strlen(p));
    puts(p);
}
```

- 4.** 下列程序的输出结果是\_\_\_\_\_。

```
#include <stdlib.h>
FUN(int **b,int p[2][3])
{
    **b=p[1][1];
}
```

```

main( )
{
    int a[2][3]={2,4,6,8,10,12},*p;
    p=(int *)malloc(sizeof(int));
    FUN(&p,a);
    printf("%d\n",*p);
}

```

5. 下面的程序用变量 count 统计文件 letter.dat 中字符的个数。请写出程序的横线处应该填入的内容。

```

#include <stdio.h>
main( )
{
    FILE *fp;
    long count=0;
    if((fp=fopen("letter.dat", "r"))==NULL){
        printf("Cannot open file!\n");
        exit(0);
    }
    while(!feof(fp)){
        _____;
        count++;
    }
    printf("count=%d\n",count);
    fclose(fp);
}

```

## 七、程序设计题

下面程序的功能是根据下列近似公式计算  $e$  的  $n$  次方。其中，函数 FUN1 用来计算每一项分子的值，函数 FUN2 用来计算每一项分母的值。请编写函数 FUN1 和函数 FUN2。

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots \dots \quad (\text{前 20 项之和})$$

```

float FUN2(int n)
{
    .....
}

float FUN1(int x,int n)
{
    .....
}

main( )
{
    float exp=1.0;
    int n,x;
    printf("Input a number:");
    scanf("%d",&x);
    printf("%d\n",x);
    exp=exp+x;
    for(n=2;n<=19;n++)
        exp=exp+FUN1(x,n)/FUN2(n);
    printf("\nThe is exp(%d)=%8.4f\n",x,exp);
}

```

## 八、程序设计题

请编写一程序，该程序将通过键盘输入的一个字符串中的小写字母全部转换成为大写字母以后输出到名为 upper.txt 的磁盘文件中保存（输入的字符串以“！”结束），然后再将文件 upper.txt 中的内容读出显示在屏幕上。

## 参考答案：

### 一、简答题

- 答:  $O(1)$  表示时间复杂度与问题规模无关。例如，在堆栈或者队列中插入一个新的元素的过程的时间复杂度为  $O(1)$ 。
- 答: 满足题目要求的出栈序列一共有 3 个，分别是 C,D,B,A,E, C,D,E,B,A 和 C,D,B,E,A。
- $(n+1)/2$
- 答: G 的拓扑序列有 3 个，分别是  $v_1, v_2, v_3, v_4, v_5$ ,  $v_1, v_3, v_2, v_4, v_5$  和  $v_1, v_3, v_4, v_2, v_5$ 。
- 答: 这种说法不正确。如果被查找的元素处在序列的前端，则采用顺序查找法比采用折半查找法所进行的元素之间的比次数少，因而时间效率要高。

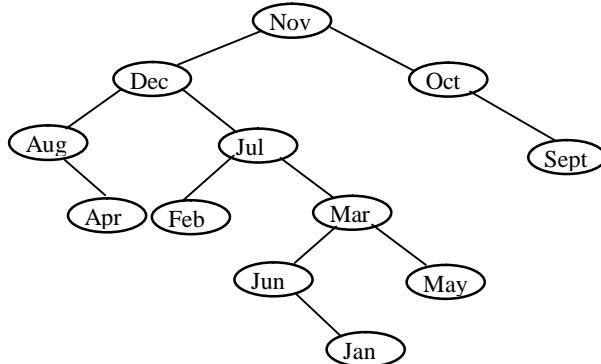
### 二、综合题

- 解: 影响一个算法（或程序）时间效率的主要因素有以下几点：
  - 算法涉及的问题的规模大小；
  - 编译程序功能的强弱以及所产生的机器代码质量的优劣；
  - 机器执行一条指令的时间长短；
  - 算法（或程序）中诸如循环语句的那些关键语句的执行次数。

其中，在很多情况下，因素④与因素①密切相关，它们是与算法（或程序）本身直接有关的因素。

- 计算由 list 所指的线性链表的长度。

3.



#### 4. 方法 1:

证明：设该无向图的边数为  $e$ ，顶点  $v_i$  的度为  $TD(v_i)$ ，有关系

$$2e = \sum TD(v_i) \quad (1 \leq i \leq n)$$

由于每一个顶点最多与图中其他的  $n-1$  个顶点有关系，即每一个顶点的度最大值为  $n-1$ ，因此， $n$  个顶点的度的最大值之和为  $n \times (n-1)$ ，即

$$2e = \sum TD(v_i) = n \times (n-1) \quad (1 \leq i \leq n)$$

于是，得到

$$e = n \times (n-1)/2$$

证毕。

#### 方法 2:

证明：如果无向图的边数达到最大，则图中每一个顶点必须与其余的  $n-1$  个顶点都存在一条边，即图中邻接每个顶点的边数均为  $n-1$ ，总的边数为  $n \times (n-1)$ 。由于对于无向图而言，有  $(v_i, v_j) = (v_j, v_i)$ ，因此，达到的最大边数为  $n \times (n-1)/2$ 。

证毕

- 第 1 趟 11, 61, 26, 1, 5, 77

### 三、算法设计题

```
void PURGE(ElemType A[], int &n)
{
    int i=0,j,k;
    while(i<n){
        for(j=i+1;j<n;j++)
            if(A[j]==A[i]){
                for(k=j;k<n;k++)
                    A[k-1]=A[k];
                n--;
                break;
            }
        i++;
    }
}
```

### 四、算法设计题

```
#define NodeNum 100
int TESTSORTTREE(BTREE T)
{
    BTREE STACK[NodeNum],p=T;
    int top=-1;
    datatype last=MinValue; /* 假设 minValue 为最小值 */
    if(T!=NULL){
        do{
            while(p!=NULL){
                STACK[++top]=p;
                p=p->lchild;
            }
            p=STACK[top--];
            if(p->data<last)
                return 0; /* 不是二叉排序树 */
            last=p->data;
            p=p->rchild;
        }while(p!=NULL||top!=-1);
    }
    return 1; /* 是二叉排序树 */
}
```

### 五、单项选择题

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

### 六、填空题

1. 0 2. LBLMNP 3. 1711717 4. 10 5. “r” fgetc(fp)

### 七、程序设计题

```
float FUN2(int n)
{
    if(n==1)
        return 1;
    else
        return FUN2(n-1)*n;
}

float FUN1(int x,int n)
{
    int i;
    float j=1;
    for(i=1;i<=n;i++)
        j=j*x;
    return j;
}
```

## 八、程序设计题

```
#include <stdio.h>
main()
{
    FILE *fp;
    char str[100],filename[10];
    int i=0;
    if((fp=fopen("upper.txt", "w"))==NULL){
        printf("Can not open file!\n");
        exit(0);
    }
    printf("Input a string:\n");
    gets(str);
    while(str[i]!='!'){
        if(str[i]>'a' && str[i]<'z')
            str[i]=str[i]-32;
        fputc(str[i],fp);
        i++;
    }
    fclose(fp);
    fp=fopen("upper.txt", "r");
    fgets(str,strlen(str)+1,fp);
    printf("%s\n",str);
    fclose(fp);
}
```

**北京航空航天大学软件学院  
2008 年 9 月自主招收软件工程硕士研究生  
专业课试题**

### 第一部分 数据结构

#### 一、是非判断题

(对于下面给出的论述, 你认为正确, 请在小题后的括号中填入符号√, 否则, 填入符号×)

1. 顺序存储结构只适用于存储线性结构。 ( )
2. 线性表的链式存储结构通过指针来反映数据元素之间的逻辑关系。 ( )
3. 在链接堆栈中插入一个新的元素等价于在链表的最前面插入一个新的链结点。 ( )
4. 非空完全二叉树的第  $i$  层一定有  $2^{i-1}$  个结点。 ( )
5. 非空二叉排序树中的任意一棵子树也是二叉排序树。 ( )
6. 任何带权的无向图都存在最小(代价)生成树。 ( )
7. 无向图的邻接矩阵一定是对称矩阵, 有向图的邻接矩阵一定是非对称矩阵。 ( )
8. 在任何情况下, 折半查找方法都要比顺序查找方法要快。 ( )
9. 对于选择排序法, 排序过程中元素之间的比较次数与原始序列的状态有关。 ( )
10. 对于具有  $n$  个元素的序列采用堆积排序法进行排序, 排序的总趟数为  $n-1$ 。 ( )

## 二、单项选择题

1. 删除长度为  $n$  的顺序表的第  $i$  个数据元素时需要移动表中\_\_\_\_\_个数据元素。  
A.  $i$       B.  $n-i$       C.  $n+i$       D.  $n-i+1$

2. 在非空双向循环链表中由  $q$  所指的那个链结点后面插入一个  $p$  指的链结点的动作对应的语句依次为:  $p->llink=q$ ;  $p->rlink=q->rlink$ ;  $q->rlink=p$ ; \_\_\_\_\_. (空白处为一条赋值语句)  
A.  $q->llink=p$ ;      B.  $q->rlink->llink=p$ ;  
C.  $p->rlink->llink=p$ ;      D.  $p->llink->llink=p$ ;

3. 在设计解决递归问题的非递归算法时, 大多数情况下都要用到\_\_\_\_\_结构。  
A. 堆栈      B. 队列      C. 树      D. 图

4. 若 3 个元素  $a, b, c$  按此先后次序进入一个初始为空的堆栈, 那么, 下面给出的四个选择中, 不可能是该堆栈的出栈序列的是\_\_\_\_\_。  
A.  $a, b, c$       B.  $c, b, a$       C.  $b, a, c$       D.  $c, a, b$

5. 若非空队列采用链式存储结构, 队头指针与队尾指针分别为  $front$  和  $rear$ , 则删除队列的一个元素的过程是依次执行:  $p=front$ ; \_\_\_\_\_,  $free(p)$ 。  
A.  $rear=p$ ;      B.  $rear=p->link$ ;  
C.  $front=p->link$ ;      D.  $front=rear->link$ ;

6. 若一棵二叉树有 10 个度为 2 的结点, 则该二叉树的叶结点的个数是\_\_\_\_\_。  
A. 9      B. 11      C. 12      D. 不确定

7. 若某完全二叉树的深度为  $h$ , 则该完全二叉树中至少有\_\_\_\_\_个结点。  
A.  $2^h$       B.  $2^h-1$       C.  $2^h+1$       D.  $2^{h-1}$

8. 若某二叉树的前序遍历序列为 ABDCEFG, 中序遍历序列为 DBCAFEG, 则其后序遍历序列为\_\_\_\_\_。  
A. DCBFGEA      B. DCBAFGE      C. GFCDEBA      D. DCFGBEA

9. 在带权图中, 两个顶点之间的路径长度是\_\_\_\_\_。  
A. 路径上的顶点数目      B. 路径上的边的数目  
C. 路径上顶点和边的数目      D. 路径上所有边上的权值之和

10. 若具有  $n$  个顶点的无向图采用邻接矩阵存储方法, 则该邻接矩阵一定为一个\_\_\_\_\_。  
A. 一般矩阵      B. 对角矩阵      C. 对称矩阵      D. 稀疏矩阵

11. 下面给出的四种操作中, 能够检测出一个有向图是否存在回路的是\_\_\_\_\_。  
A. 广度优先搜索      B. 拓扑排序      C. 求最短路径      D. 求关键路径

12. 若在线性表中采用折半查找方法进行查找, 该线性表必须\_\_\_\_\_。  
A. 元素按值有序排列  
B. 采用顺序结构  
C. 元素按值有序排列, 并且采用顺序存储结构  
D. 元素按值有序排列, 并且采用链式存储结构

13. 在建立散列表时, 若散列函数为  $H(k)$ ,  $a$  与  $b$  分别为关键字值, 则当\_\_\_\_\_时, 称此现象为散列冲突。  
A.  $a=b$       B.  $a \neq b$       C.  $a=b$  且  $H(a)=H(b)$       D.  $a \neq b$  且  $H(a)=H(b)$

14. 每一趟排序都从未排序序列中依次取出一个元素依次与已排序序列中的元素进行比较，然后将其放在已排序序列中的合适位置，这种排序方法称为\_\_\_\_\_。

- A. 选择排序法      B. 插入排序法      C. 泡排序法      D. 堆积排序法

15. 根据(大顶)堆积的定义，下面给出的四个序列中，\_\_\_\_\_是一个堆积。

- A. 75,45,65,30,15,25,20,10      B. 75,65,45,10,30,25,20,15  
C. 75,65,30,15,25,45,20,10      D. 75,45,65,10,25,30,20,15

### 三、简答题

1. 什么情况下，线性表采用顺序存储结构比采用链式存储结构要更合适？
2. 一棵度为 2 的树与一棵二叉树有何区别？
3. 要使得具有  $n$  个顶点的有向图成为强连通图，至少需要有多少条边？
4. 在采用线性探测再散列方法处理冲突的散列表中，同义词(即散列地址相同的关键字值)在散列表中的位置一定是相邻的，这种说法正确吗？为什么？
5. 若对序列(1, 4, 6, 2, 5)采用泡排序法进行从小到大排序，则排序过程中一共要进行多少次元素之间的比较？

### 四、算法填空题

已知非空二叉排序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  $T$ 。下面给出的是在该二叉排序树中查找数据信息为  $item$  的结点的非递归算法，若查找成功，则算法返回被查到结点所在链结点指针，否则，算法返回信息  $NULL$ 。

请在算法中的空白处(横线上方)填入必要的内容，使得算法完整正确。

BTREE SREACHBST(BTREE T,datatype item)

```
{  
    BTREE p=T;  
    while( ① ){  
        if(p->data==item)  
            return p;  
        if( ② )  
            p=p->lchild;  
        else  
            ③  
    }  
    return NULL;  
}
```

## 第二部分 C 语言程序设计

### 五、单项选择题

1. 下列关于 C 语言的叙述中，正确的是\_\_\_\_\_。

- A. C 程序中的注释部分可以出现在程序中任意合适的地方  
B. 花括号 “{” 和 “}” 只能作为函数体的定界符  
C. 构成 C 程序的基本单位是函数，所有函数名都可以由用户命名  
D. 分号是 C 语句之间的分隔符，不是语句的一部分

2. 在 C 语言中，要求运算数必须是整型的运算符是\_\_\_\_\_。

- A. /      B. %      C. !=      D. ++

3. 下面给出的四个选择中，不能作为 C 语言合法的表达式的是\_\_\_\_\_。

- A.  $0 \leq y < 100$       B.  $i=j==0$       C.  $(char)(65+3)$       D.  $y+1=y+1$

4. 若  $x, i, j$  和  $k$  分别是  $int$  类型的变量, 则计算表达式  $x=(i=4,j=16,k=32)$  以后, 变量  $x$  的值是\_\_\_\_\_。

- A. 4      B. 16      C. 32      D. 52

5. 若变量  $y$  为  $float$  类型, 且已经被赋值, 则下列语句中能够将  $y$  中数值保留到小数点后面两位, 并且将第三位四舍五入的是\_\_\_\_\_。

- A.  $y=y*100+0.5/100.0;$       B.  $y=(y*100+0.5)/100.0;$   
C.  $y=(y/100+0.5)*100.0;$       D.  $y=(int)(y*100+0.5)/100.0;$

6. 若已定义  $ch$  为字符型变量, 则下列赋值语句中, 错误的是\_\_\_\_\_。

- A.  $ch='\';$       B.  $ch=62+3;$       C.  $ch=NULL;$       D.  $ch='\\xaa';$

7. 若有如下程序段, 其中  $s, a, b, c$  均被定义为整型变量, 并且  $a$  和  $c$  已经赋值( $c>0$ ),

```
s=a;  
for(b=1;b<=c;b++) s=s+1;
```

则与上述程序段功能等价的赋值语句是\_\_\_\_\_。

- A.  $s=a+b;$       B.  $s=a+c;$       C.  $s=s+c;$       D.  $s=b+c;$

8. 下列选择中, 不能正确定义二维数组的是\_\_\_\_\_。

- A.  $int a[2][2]=\{\{1\},\{2\}\};$       B.  $int a[][2]=\{1,2,3,4\};$   
C.  $int a[2][]=\{\{1,2\},\{3,4\}\};$       D.  $int a[2][2]=\{\{1\},2,3\};$

9. 为避免在嵌套的条件语句  $if-else$  中产生二义性, C 语言规定  $else$  子句总是与\_\_\_\_\_配对。

- A. 缩排位置相同的  $if$       B. 同一行上的  $if$   
C. 其之后最近的  $if$       D. 其之前最近的  $if$

10. 在 C 语言中,  $while$  循环和  $do-while$  循环的主要区别在于\_\_\_\_\_。

- A.  $do-while$  的循环体至少无条件被执行一次  
B.  $while$  的循环控制条件比  $do-while$  的循环控制条件要严格  
C.  $do-while$  允许从外部转到循环体内  
D.  $do-while$  的循环体不能是复合语句

11. 以下正确的函数定义形式是\_\_\_\_\_。

- A.  $double FUN(int x;int y)$       B.  $double FUN(int x,int y)$   
C.  $double FUN(int x,int y);$       D.  $double FUN(int x,y)$

12. 若已经定义了如下函数

```
FUN(*p)  
{ return *p; }
```

则该函数的返回值是\_\_\_\_\_。

- A. 不确定的值      B. 形参  $p$  中存放的值  
C. 形参  $p$  所指的存储单元中的值      D. 形参  $p$  的地址值

13. 下列能够正确进行字符串赋值操作的是\_\_\_\_\_。

- A.  $char s[5]=\{\text{“ABCDE”}\};$       B.  $char *s; s=\text{“ABCDE”};$   
C.  $char s[5]=\{\text{‘A’},\text{‘B’},\text{‘C’},\text{‘D’},\text{‘E’}\};$       D.  $char *s; scanf(\%s,s);$

14. 下列四个程序段中, 正确的是\_\_\_\_\_。

- A.  $char str[20];$   
 $scanf(\%s,&str);$       B.  $char *p;$   
 $scanf(\%s,p);$

C. char str[20];  
scanf("%s", &str[2]);  
D. char str[20], \*p=str;  
scanf("%s", p[2]);

15. 若程序中已经包含头文件 stdio.h, 则下列程序段中, 正确运用指针变量的是\_\_\_\_\_。

A. int \*i=NULL;  
scanf("%d", i);  
B. float \*f=NULL;  
\*f=10.5;

C. char t='m', \*c=&t;  
\*c=&t;  
D. long \*L;  
L='0';

16. 当说明一个结构体变量时, 系统分配给它的内存是\_\_\_\_\_。

A. 各成员所需要的内存量的总和  
B. 结构中第一个成员所需要的内存量  
C. 成员中占内存量最大者所需要的内存量  
D. 结构中最后那个成员所需要的内存量

17. 以下 scanf 函数调用语句中对结构体变量成员的不正确引用的是\_\_\_\_\_。

```
struct node{  
    char name[20];  
    int age;  
    int sex;  
}student[5], *p;  
p=student;
```

A. scanf("%s", student[0].name);  
B. scanf("%d", &student[0].age);  
C. scanf("%d", &(p->sex));  
D. scanf("%d", p->age);

18. 若执行函数 fopen 时发生错误, 则函数的返回值是\_\_\_\_\_。

A. 地址值  
B. 0  
C. 1  
D. EOF

19. 函数 fgetc 的作用是从指定文件中读入一个字符, 该文件的打开方式必须是\_\_\_\_\_。

A. 只写  
B. 追加  
C. 读或读写  
D. 答案 B 和 C 都正确

20. 若以 “a+” 方式打开一个已经存在的文件, 则下列叙述中, 正确的是\_\_\_\_\_。

A. 文件打开时, 原有文件内容不被删除, 位置指针移到文件的末尾, 可作添加和读操作  
B. 文件打开时, 原有文件内容不被删除, 位置指针移到文件的开头, 可作重写和读操作  
C. 文件打开时, 原有文件内容被删除, 只可作写操作  
D. 以上各种叙述都不正确

## 六、填空题

1. 若已有如下定义:

```
struct node{  
    int data;  
    struct node *link;  
} *p;
```

并且希望通过语句 “p=(struct node)malloc(\_\_\_\_);” 使得指针 p 指向一个具有 struct node 类型的动态存储空间, 则该语句中的空白处 (横线上方) 应该填入\_\_\_\_\_。

2. 若有定义: “int x[10], \*p, p=x;”, 则在程序中引用数组元素 x[i] 的四种形式中, 除了 x[i]、p[i] 和\*(p+i) 以外, 还有\_\_\_\_\_。

3. 下列程序的功能是将从键盘输入的一对整数由小到大排序输出, 当输入的一对整数相等时结束循环。程序中的空白处(横线上方)应该填入\_\_\_\_\_。

```
#include <stdio.h>  
main()  
{  
    int a,b,temp;
```

```

scanf("%d%d",&a,&b);
while( _____ ){
    if(a>b){
        temp=a;
        a=b;
        b=temp;
    }
    printf("%d,%d\n",a,b);
    scanf("%d%d",&a,&b);
}

```

4. 若下列程序中的函数 scmp 功能是返回形参指针 s1 和 s2 所指字符串中较小字符串的首地址，并且运行程序时依次输入 abcd、abba 和 abc 三个字符串，则该程序的输出结果是\_\_\_\_\_。

```

#include <stdio.h>
#include <string.h>
char *scmp(char *s1,char *s2)
{
    if(strcmp(s1,s2)<0)
        return s1;
    else
        return s2;
}

main()
{
    int i;
    char string[20],str[3][20];
    for(i=0;i<3;i++)
        gets(str[i]);
    strcpy(string,scmp(str[0],str[1]));
    strcpy(string,scmp(string,str[2]));
    printf("%s\n",string);
}

```

5. 下列程序的主要功能是\_\_\_\_\_。

```

#include <stdio.h>
main()
{
    FILE *in,*out;
    char ch,infile[10],outfile[10];
    printf("Enter the infile name:\n");
    scanf("%s",infile);
    printf("Enter the outfile name:\n");
    scanf("%s",outfile);
    if((in=fopen(infile, "r"))==NULL){
        printf("Cannot open infile\n");
        exit(0);
    }
    if((out=fopen(outfile, "w"))==NULL){
        printf("Cannot open outfile\n");
        exit(0);
    }
    while(!feof(in))
        fputc(fgetc(in),out);
    fclose(in);
    fclose(out);
}

```

## 七、程序设计题

已知整型数组 A[0..m-1][0..n-1]，请写一**函数**，该函数返回数组最外围一圈元素之和。

## 八、程序设计题

请写一程序，统计通过键盘输入的命令行中的第二个参数所包含的英文字符的个数。

**提示：**使用带参数的 main 函数形式。

### 参考答案：

#### 一、是非判断题

1. × 2. √ 3. √ 4. × 5. √ 6. × 7. × 8. × 9. × 10. √

#### 二、单项选择题

1. B 2. C 3. A 4. D 5. C 6. B 7. D 8. A 9. D 10. C  
11. B 12. C 13. D 14. B 15. A

#### 三、综合题

1. 答：当对线性表进行操作的过程中不需要大量移动数据元素时，线性表采用顺序存储结构比采用链式存储结构要更合适。

2. 答：二叉树的子树有严格的左、右之分，其次序不能随意颠倒。

3. 答：要使得具有 n 个顶点的有向图成为强连通图，至少需要有 n 条边。

4. 答：这种说法不正确，同义词在散列表中的位置不一定相邻。因为，若当发生散列冲突时的“下一个”位置是空闲的，则同义词在散列表中位置是相邻的；若发生散列冲突时的“下一个”位置此前已被分配(或者说被其他关键字占用)，此时同义词在散列表中的位置会不相邻。

5. 答：对序列(1, 4, 6, 2, 5)采用泡排序法进行排序，排序过程中进行的元素间的比较次数为 9 次。

#### 四、算法填空题

- ① p!=NULL            ② item<p->data            ③ p=p->rchild;

#### 五、单项选择题

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

#### 六、填空题

1. sizeof(struct node) 或 4            2. \*(x+i)            3. a!=b  
4. Abba                                    5. 将一个磁盘文件复制到另一个磁盘文件中

#### 七、程序设计题

```
int COUNT(int A[][n])
{
    int k,sum=0;
    for(k=0;k<n;k++) { /* 累加第 1 行与第 m 行所有元素之值 */
        sum+=A[0][k];
        sum+=A[m-1][k];
    }
    for(k=0;k<m;k++) { /* 累加第 1 列与第 n 列所有元素之值 */
        sum+=A[k][0];
        sum+=A[k][n-1];
    }
    return sum-A[0][0]-A[0][n-1]-A[m-1][0]-A[m-1][n-1];
    /* 从累加和中减去四个角上的元素之值，并返回结果 */
}
```

#### 八、程序设计题

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

```

#include <ctype.h>
main(int argc,char *argv[])
{
    char *ptr;
    int num=0;
    if(argc<2)
        exit(1);
    ptr=argv[1];
    while(*ptr)
        if(isalpha(*ptr++))
            num++;
    printf("\nThe count is :%d.\n",num);
}

```

**北京航空航天大学软件学院**  
**2009 年 6 月自主招收软件工程硕士研究生**  
**专业课试题**

**第一部分 数据结构**

**一、单项选择题**

1. 算法分析的主要任务是分析\_\_\_\_\_。
  - A. 算法的执行效率与问题规模之间的关系
  - B. 算法中是否存在语法错误
  - C. 算法的功能是否符合设计要求
  - D. 算法是否具有较好的可读性
  
2. 下面关于线性表的叙述中, 错误的是\_\_\_\_\_。
  - A. 线性表采用顺序存储结构, 必须占用一片连续的存储单元
  - B. 线性表采用顺序存储结构, 便于进行插入和删除操作
  - C. 线性表采用链式存储结构, 不必占用一片连续的存储单元
  - D. 线性表采用链式存储结构, 便于进行插入和删除操作
  
3. 在非空线性链表中由 p 所指的结点后面插入一个由 q 所指的结点的过程是依次执行\_\_\_\_\_。
  - A. q->link=p; p->link=q;
  - B. q->link=p->link; p=q;
  - C. q->link=p->link; p->link=q;
  - D. p->link=q; q->link=p;
  
4. 若 4 个元素进栈的先后次序为 a,b,c,d, 下面给出的 4 个选择中, 不可能是该堆栈的输出序列的是\_\_\_\_\_。
  - A. a,c,b,d
  - B. b,c,d,a
  - C. d,b,c,a
  - D. c,d,b,a
  
5. 树型结构最适合用来描述\_\_\_\_\_。
  - A. 有序的数据
  - B. 无序的数据
  - C. 数据元素之间没有关系的数据
  - D. 数据元素之间具有层次关系的数据
  
6. 下面关于二叉树的叙述中, 正确的是\_\_\_\_\_。
  - A. 二叉树的度为 2
  - B. 二叉树的度可以小于 2
  - C. 二叉树中至少有一个结点的度为 2
  - D. 二叉树中任何一个结点的度都为 2
  
7. 深度为 h 的满二叉树的第 i 层的结点总数是\_\_\_\_\_。 $(i \leq h)$ 
  - A.  $2^{i-1}$
  - B.  $2^i - 1$
  - C.  $2^{h-1}$
  - D.  $2^h - 1$

8. 有向图的邻接表的第  $i$  个链表中的边结点数目是第  $i$  个顶点的\_\_\_\_\_。  
A. 边数      B. 度数      C. 入度      D. 出度
9. 具有  $n$  个顶点的无向图的边数最大是\_\_\_\_\_。  
A.  $n(n+1)/2$       B.  $n^2$       C.  $n(n-1)$       D.  $n(n-1)/2$
10. 在一个图中，所有顶点的度数之和等于所有边数的\_\_\_\_\_。  
A.  $1/2$  倍      B. 1 倍      C. 2 倍      D. 4 倍
11. 在具有  $n$  个数据元素的线性表中进行顺序查找，若查找每个元素的概率相等，则平均查找长度  $ASL=$ \_\_\_\_\_。  
A.  $(n+1)/2$       B.  $n/2$       C.  $n$       D.  $(n-1)/2$
12. 下面关于折半查找法的叙述中，正确的是\_\_\_\_\_。  
A. 只要线性表中元素按值有序排列，就可以采用折半查找  
B. 只要线性表采用顺序存储结构，就可以采用折半查找  
C. 线性表中元素按值有序排列，并且采用顺序存储结构时才能采用折半查找  
D. 在链表中也可以采用折半查找
13. 假设  $n$  个关键字互为同义词，若采用线性探测再散列法处理冲突，把这些关键字散列到一个散列表中，则进行的探测次数是\_\_\_\_\_。  
A.  $n-1$       B.  $n$       C.  $n+1$       D.  $n(n-1)/2$
14. 对具有  $n$  个元素的序列采用插入排序法进行排序，排序总趟数为\_\_\_\_\_。  
A.  $n$       B.  $n-1$       C.  $n+1$       D.  $\lfloor \log_2 n \rfloor$
15. 下面关于内排序方法的时间效率的叙述中，正确的是\_\_\_\_\_。  
A. 时间效率主要与排序趟数的多少有关  
B. 时间效率主要与参加排序的序列中元素的多少有关  
C. 时间效率主要与排序过程中元素移动或者交换次数的多少有关  
D. 时间效率主要与排序过程中元素之间的比较次数的多少有关

## 二、填空题

- 删除非空顺序表的\_\_\_\_\_那个数据元素时不必移动表中其他元素的位置。
- 线性表的链式存储结构主要有\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_3 种形式。
- 元素进/出堆栈满足\_\_\_\_\_的规律。
- 在长度为  $n$  的队列中插入一个新元素的操作的时间复杂度为\_\_\_\_\_。
- 若二叉树中叶结点的个数为  $n_0$ ，则度为 2 的结点的个数为\_\_\_\_\_。
- 若某完全二叉树采用顺序存储结构，结点信息存放的次序是 A,C,B,E,F,D，则该二叉树的后序遍历序列为\_\_\_\_\_。
- 要得到二叉排序树所有结点组成的按值有序的序列，可以对二叉排序树进行\_\_\_\_\_遍历。
- 若具有  $n$  个顶点且不带权的连通图采用邻接矩阵存储，则该邻接矩阵中至少有\_\_\_\_\_个非零元素。
- 图的深度优先遍历类似于二叉树的\_\_\_\_\_遍历。
- 通过拓扑排序能够得到拓扑序列的图一定是一个\_\_\_\_\_的图。

11. 在顺序表(2,4,6,8,10)中采用折半查找方法查找元素 5, 要经过\_\_\_\_次元素之间的比较。
12. 若散列函数为  $H(k)$ ,  $a$  和  $b$  为两个不同的关键字值, 则当出现\_\_\_\_时, 称此现象为散列冲突。
13. 对序列(1,2,4,3,5)采用泡排序法进行排序, 整个排序过程中进行了\_\_\_\_次元素之间的比较。
14. 在\_\_\_\_的情况下, 快速排序法就会成为“慢速排序法”。
15. 对序列(50,72,28,39,81,15)中的元素按值从小到大进行排序, 若已知第 1 趟排序的结果是(15,72,28,39,50,81), 则可以断定采用的排序方法是\_\_\_\_。

### 三、综合题

1. 某堆栈初始为空, 符号 PUSH 和 POP 分别表示 1 次进栈操作和 1 次出栈操作。对于进栈序列 a,b,c,d,e, 经过 PUSH, PUSH, POP, PUSH, POP, PUSH, PUSH 时, 得到的出栈序列是什么?
2. 若具有  $n$  个结点的非空二叉树采用二叉链表作为存储结构, 则链表中一共有  $n+1$  个指针域存放 NULL。请写出该结论的推导过程。
3. 已知无回路的有向图  $G=(V,E)$ , 其中,  $V=\{a,b,c,d,e\}$ ,  $E=\{<a,b>,<a,c>,<b,e>,<c,b>,<c,e>,<d,c>\}$ , 请首先画出该有向图, 然后写出该图的任意一个拓扑序列。

### 四、算法设计题

已知单链表的结点结构为 

data	link
------	------

, 第 1 个结点的指针为 list。请写一算法, 找到链表的倒数第  $k$  个结点。若找到这样的结点, 算法给出该结点的地址, 否则, 算法给出信息 NULL。

**限制:** 算法中不得求链表长度与逆转链表, 也不允许使用除指针变量和控制变量以外的其他辅助空间。

## 第二部分 C 语言程序设计

### 五、单项选择题

1. 以下说法中, 正确的是\_\_\_\_。
  - A. C 语言程序总是从第一个函数开始执行
  - B. C 语言程序总是从 main() 函数开始执行
  - C. 在 C 语言程序中, 要调用的函数必须在 main() 函数中定义
  - D. C 语言程序中的 main() 函数必须放在程序的开始部分
2. 若变量已正确定义并赋值, 下面给出的表达式中, 符合 C 语言语法要求的是\_\_\_\_。
  - A.  $a:=b+1$
  - B.  $int 18.5\%3$
  - C.  $a=a+7=c-b$
  - D.  $a=b=c+2$
3. 若有说明:  $char w; int x; float y; double z;$  则表达式  $w*x+y-z$  的值的数据类型是\_\_\_\_。
  - A. double
  - B. char
  - C. int
  - D. float
4. 若  $a,b$  均为 float 类型变量, 则以下不符合 C 语言语法的赋值语句是\_\_\_\_。
  - A.  $++a;$
  - B.  $a*=b+8;$
  - C.  $b=(a\%2)/10;$
  - D.  $a=b=0;$
5. 判断 char 类型变量  $c$  是否为小写字母的正确表达式是\_\_\_\_。
  - A.  $‘a’ \leq c \leq ‘z’$
  - B.  $(c \geq a) \&\& (c \leq z)$
  - C.  $(‘a’ \geq c) \|(‘z’ \leq c)$
  - D.  $(c \geq ‘a’) \&\& (c \leq ‘z’)$
6. 从循环体内某一层跳出, 继续执行本循环的下一次循环的语句是\_\_\_\_。
  - A. break 语句
  - B. continue 语句
  - C. return 语句
  - D. 空语句

7. 以下程序段\_\_\_\_\_。

```
x=-1;  
do{ x=x*x;  
}while(!x);
```

- A. 是死循环      B. 循环执行 2 次      C. 循环执行 1 次      D. 有语法错误

8. 若有说明: int a[3][4]; 则对 a 数组元素的非法引用的是\_\_\_\_\_。

- A. a[0][2\*1]      B. a[0][4]      C. a[4-2][0]      D. a[1][3]

9. 若有定义: int a[10],\*p; 则\*(p+5)表示\_\_\_\_\_。

- A. 元素 a[5]的值      B. 元素 a[5]的地址  
C. 元素 a[6]的值      D. 元素 a[6]的地址

10. 下列选项中, 正确的语句组是\_\_\_\_\_。

- A. char s[8]; s={"Beijing"};      B. char \*s; s={"Beijing"};  
C. char s[8]; s="Beijing";      D. char \*s; s="Beijing";

11. 若有说明: int \*p1,\*p2,m=5,n; 以下都是正确的赋值语句的选项是\_\_\_\_\_。

- A. p1=&m; p2=&p1;      B. p1=&m; p2=&n; \*p1=\*p2;  
C. p1=&m; p2=p1;      D. p1=&m; \*p2=\*p1;

12. 若有如下定义:

```
struct sk{  
    int a;  
    int b;  
}data,*p=&data;
```

则对 data 中的成员 x 的正确引用的是\_\_\_\_\_。

- A. (\*p).a      B. (\*p).data.a      C. p->data.a      D. p.data.a

13. 对于函数定义:

```
void FUN(int n, double x)  
{ ... }
```

若以下选项中的变量都已定义并赋值, 则对函数 FUN 的正确调用的语句是\_\_\_\_\_。

- A. k=FUN(x,n);      B. void FUN(n,x);  
C. FUN(int y,double m);      D. FUN(10,12.5);

14. printf( )函数中用到格式符%4s, 其中, 数字 4 表示输出的字符串占用 4 列。若字符串长度大于 4, 则输出按下列 4 种方式之一进行, 该方式是\_\_\_\_\_。

- A. 从左起输出该字符串, 右补空格      B. 按原字符串长度从左向右全部输出  
C. 右对齐输出该字符串, 左补空格      D. 输出错误信息

15. 在下列关于对文件进行操作的叙述中, 正确的是\_\_\_\_\_。

- A. 对文件操作必须先关闭文件  
B. 对文件操作必须先打开文件  
C. 对文件操作之前必须先测试文件是否存在, 然后再打开文件  
D. 对文件操作的顺序没有要求

16. 若要打开一个已经存在的非空文件 “file” 并进行修改, 则正确的打开语句是\_\_\_\_\_。

- A. fp=fopen("file","r");      B. fp=fopen("file","ab+");  
C. fp=fopen("file","w+");      D. fp=fopen("file","r+");

17. 下面程序的运行结果是\_\_\_\_\_。

```
#include <stdio.h>
main()
{    int num=0;
    while(num<=2){
        num++;
        printf("%d\n",num);
    }
}
```

A. 1

B. 1  
2

C. 1  
2  
3

D. 1  
2  
3  
4

18. 下面程序段的运行结果是\_\_\_\_\_。

```
char a[7]=“abcdef”;
char b[4]=“ABC”;
strcpy(a,b);
printf("%c",a[5]);
```

A. f

B. e

C. \0

D.  $\cup$  ( $\cup$  表示空格)

19. 下面程序的运行结果是\_\_\_\_\_。

```
#define DOUBLE(r) r*r
main()
{    int a=1,b=2,temp;
    temp=DOUBLE(a+b);
    printf("%d\n",temp);
}
```

A. 3

B. 5

C. 7

D. 9

20. 下面程序段的运行结果是\_\_\_\_\_。

```
char *p=“abcdefgh”;
p+=3;
printf("%d\n",strlen(strcpy(p,“ABCD”)));
```

A. 4

B. 7

C. 8

D. 12

## 六、综合题

1. 阅读下列函数，简单写出该函数的功能。

```
void FUNCTION(int K[],int n)
{
    int i,j,d;
    int temp;
    for(i=0; i<n-1; i++){
        d=i;
        for(j=i+1; j<n; j++)
            if(K[j]<K[d])
                d=j;
        if(d!=i){
            temp=K[d];
            K[d]=K[i];
            K[i]=temp;
        }
    }
}
```

/\* 每次先假设 K[i] 为最小值元素 \*/  
/\* 寻找真正的最小值元素，记录其位置 d \*/  
/\* 若最小值元素为 K[d] \*/  
/\* 交换 K[d] 与 K[i] 的位置 \*/

2. 假设整型数组 a 中的数据已经按值从小到大顺序存放。下面的程序首先删除该数组中值相同的多余数据，然后以每一行 4 个数据的格式输出该数组。请先阅读该程序，然后分别在程序的空白处(横线上方，共 3 处)填上必要内容。(注：所谓删除值相同的多余数据即把值相同的数据删除得只剩一个)。

```

#define M 50
main( )
{
    int a[M],i,j,n;
    for(i=0; i<M; i++)
        scanf("%d", a+i);
    n=i=M-1;
    while(i>=0){
        if(*(a+i)==*(a+i-1)){
            for(j=____; j<=n; j++)
                *(a+j-1)=*(____);
            n--;
        }
        i--;
    }
    for(i=1; i<=n+1; i++){
        printf("%4d", *(____));
        if(i%4==0)
            printf("\n");
    }
    printf("\n");
}

```

/\* 若相邻的两个数据相同 \*/  
/\* 删除其中的一个 \*/  
/\* 以每一行 4 个数据的格式输出数组 \*/  
/\* 换行 \*/

## 七、程序设计题

请编写一程序，该程序统计并输出文本文件 file.dat 中的字符个数。

## 八、程序设计题

已知一整型数组 a 中包含 100 个整数，请写一程序，该程序求出(记录)该数组中最大值元素的位置。

- 提示：**① 先通过键盘输入使数组得到数据；  
② 数组中最大值元素可能不止一个，甚至可能都是最大值元素。

## 参考答案：

### 一、单项选择题

1. A    2. B    3. C    4. C    5. D    6. B    7. A    8. D    9. D    10. C  
11. A   12. C   13. D   14. B   15. D

### 二、填空题

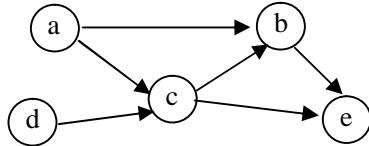
- |                 |                    |
|-----------------|--------------------|
| 1. 最后           | 2. 单链表、循环链表 和 双向链表 |
| 3. 先进后出 (或后进先出) | 4. O(1)            |
| 5. $n_0 - 1$    | 6. E,F,C,D,B,A     |
| 7. 中序           | 8. $2(n-1)$        |
| 9. 前序           | 10. 有向无环           |
| 11. 3           | 12. $H(a)=H(b)$    |
| 13. 7           | 14. 参加排序的元素按值基本有序  |
| 15. 堆积排序法       |                    |

### 三、综合题

1. 得到的出栈序列是 b,c  
2. 链表中一共有  $2n$  个指针域。已知具有 n 个结点的非空二叉树的分支总数为  $n-1$ ，而每一个分支对应一个指针，因此，有

$$2n-(n-1)=n+1$$

3.



拓扑序列分别为 a,d,c,b,e 与 d,a,c,b,e。

#### 四、算法设计题

```
LinkList SEARCHNODE(LinkList list,int k)
{
    LinkList p,r;
    int i;
    if(list!=NULL && k>0){
        p=list;
        for(i=1;i<k;i++){ /* 循环结束时, p 指向链表的第 k 个结点(不是倒数第 k 个结点) */
            p=p->link;
            if(p==NULL){
                printf("链表中不存在倒数第 k 个结点! ")
                return NULL;
            }
        }
        r=list;
        while(p->link!=NULL){
            p=p->link;
            r=r->link;
        }
        return r; /* 循环结束时, p 指向链表最后那个结点, r 指向倒数第 k 个结点 */
        /* 给出链表倒数第 k 个结点(r 指向的那个结点)的地址 */
    }
}
```

#### 五、单项选择题

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

#### 六、填空题

1. 该函数的功能是采用选择排序法将一整数序列的元素按值从小到大进行排序。  
2. ① i            ② a+j            ③ a+i-1

#### 七、程序设计题

```
#include <stdio.h>
main()
{
    FILE fp;
    int num=0;
    if((fp=fopen("file.dat","r"))==NULL){
        printf("Can not open file!\n");
        exit(0);
    }
    while(!feof(fp)){
       getc(fp);
        num++;
    }
    printf("num=%d\n",num);
    fclose(fp);
}
```

#### 八、程序设计题

**基本思想:** 由于最大元素可能不止一个, 甚至都可能是最大元素, 因此, 定义一个长度为 101 的整型数组 pos[101], 用来分别记录最大元素的位置。在确定最大元素之前, 假设第一个元素为最

大元素，其位置用整型变量 position 记录；从第二个元素开始至最后那个元素，依次与当前最大元素进行比较。若有新的元素大于当前最大元素，position 记录新的元素的位置，新的元素成为当前最大元素，同时置整型变量 k 为 1；若有新的元素与当前最大元素相等，将当前比较的元素的位置保存在 pos 数组由整型变量 k 所指的位置，并将 k 后移一个位置。在数组所有元素都比较完毕时，在 pos[k] 中记录一个标记值 -1 来标明数组 pos 下标 0 至 k-1 的元素中记录了 k 个最大元素的位置。

```
main()
{
    int a[100];
    int pos[101];
    int i,k=1,position=0;
    printf("\nInput Number:");
    for(i=0;i<100;i++)
        scanf("%d",&a[i]);
    for(i=1;i<100;i++)
        if(a[i]>a[position]){
            position=i;
            k=1;
        }
        else if(a[i]==a[position])
            pos[k++]=position;
    pos[0]=position;
    pos[k]=-1;
}
```

**北京航空航天大学软件学院**  
**2009 年 12 月自主招收软件工程硕士研究生**  
**专业课试题**

**第一部分 数据结构**

**一、填空题**

1. 对于线性表的顺序存储结构与链式存储结构而言，若表的长度基本稳定，并且很少进行插入和删除操作，但要以尽可能快的时间效率存取表中元素，则应该选择的存储结构是\_\_\_\_\_。
2. 若已知由 list 所指的非空单链表的结点构造为 

data	link
------	------

，则在链表的最前面插入一个由指针 p 指的结点的过程是依次执行\_\_\_\_\_。
3. “删除栈顶元素”、“删除栈底元素”、“判断堆栈是否为空”和“将堆栈置为空栈”这 4 种操作中，\_\_\_\_\_不是堆栈的基本操作。
4. 按层次从上至下，每一层从左至右的顺序将二叉树的结点信息依次存放在数组元素 BT[1]~BT[n] 中，结点 BT[i] 如果存在右孩子，则该右孩子是\_\_\_\_\_。
5. 对二叉排序树进行\_\_\_\_\_遍历，得到的遍历序列一定是一个按结点值的大小排列的序列。
6. 若具有 n 个顶点、e 条边且不带权的无向图采用邻接矩阵存储，则邻接矩阵中的零元素的数目是\_\_\_\_\_。
7. 在实现图的广度优先遍历时要用到队列，在遍历过程中，图的每个顶点最多进队\_\_\_\_\_次。
8. 折半查找的过程可以借助于一棵称之为“判定树”的二叉树来描述。在表长为 n 的有序表中进行折半查找所对应的判定树的高度为\_\_\_\_\_。

9. 一个“好的”散列函数是指\_\_\_\_\_。

10. 若对序列(tang, deng, an, wang, shi, bai, fang, liu)按字典顺序进行排序, 采用的排序方法是选择排序法, 那么, 第二趟排序结束时, 序列的状态是\_\_\_\_\_。

## 二、简答题

- 相对于线性表的顺序存储结构, 线性表的链式存储结构有什么优点?
- 对于一个带权连通图, 在什么情况下, 利用普里姆(Prim)算法与利用克鲁斯卡尔(Kruskal)算法可能生成不同的最小生成树?
- 如果说堆栈与队列是同一类的两种不同的数据结构, 那么, 它们的相同点和不同点分别是什么?
- 若选择当前排序的第 1 个元素作为分界元素(也称枢轴或支点), 什么情况下, 快速排序法的时间效率会退化到简单排序法的程度? 请说明理由。

## 三、综合题

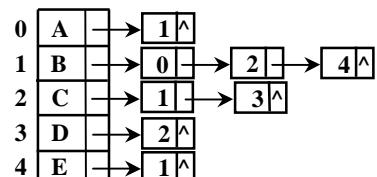
1. 下面算法的功能是依次打印十进制数 num 对应的八进制形式的各位数字。算法中用到了一个采用链式存储结构的堆栈。

请在算法的空白处(方框内)填入必要的内容, 使算法完整。

```
typedef struct node {
    int data;
    struct node *link;
}STNode, *STLink;

void CHANGE( int num )
{
    STLink p, top=NULL;
    do{
        p=(STLink)malloc(sizeof(STNode));
        p->data=num % 8;
        [ ] /* 进栈 */
        top=p;
        num=num / 8;
    }while(num!=0);
    while( [ ] ){ /* 如果栈不空 */
        printf("%d",top->data);
        p=top;
        [ ] /* 退栈 */
        free(p);
    }
}
```

2. 若已知某无向图的邻接表如图三—2 所示, 请分别写出根据该邻接表从顶点 A 开始进行深度优先遍历与广度优先遍历得到的遍历序列。



图三—2

- 已知对一棵二叉排序树进行前序遍历得到的遍历序列为 50,45,35,15,40,46,65,75,70 请画出该二叉排序树。
- 请用完全二叉树的形式画出序列(26,5,77,1,61,11,59,15,48,19) 对应的大顶堆积(Heap)。

## 四、算法设计题

已知指针为 list 且不带头结点的非空线性链表的结点构造为 

data	link
------	------

，结点按 data 域值的大小从小到大链接。请写出在该链表中插入一个数据信息为 item 的新结点的算法，要求插入新结点后链表中结点仍然按 data 域值的大小从小到大链接。

## 第二部分 C 语言程序设计

### 五、单项选择题

1. C 语言中最简单的数据类型包括\_\_\_\_\_。  
A. 整型、实型、逻辑型      B. 整型、实型、字符型  
C. 整型、字符型、逻辑型      D. 整型、实型、逻辑型、字符型
2. 逻辑运算符两侧的运算对象的数据类型\_\_\_\_\_。  
A. 只能是 0 或者 1      B. 只能是 0 或者非 0 正数  
C. 只能是整数或者字符型数据      D. 可以是任何类型的数据
3. 以下不符合 C 语言语法的赋值语句是\_\_\_\_\_。  
A. `n=(i2,i++);`      B. `x=y>0;`      C. `++(i+1);`      D. `j++;`
4. 若变量 x 为 int 类型，并且值为 4，则执行表达式 `x+=x-=x*x` 以后，x 的值为\_\_\_\_\_。  
A. -24      B. -16      C. 16      D. 24
5. 若 t=1, a=2, b=3, c=4，则条件表达式 `t<a? t:b<c? b:c` 的值是\_\_\_\_\_。  
A. 0      B. 1      C. 2      D. 3
6. 设 x=12、y=12345，执行语句 `printf("%4d,%4d",x,y);` 的输出结果是\_\_\_\_\_。  
A. 12,123      B. 12,1234      C. 12,12345      D. 12,123456
7. 在 C 语言中，用于结构化程序设计的 3 种基本结构是\_\_\_\_\_。  
A. 顺序结构、选择结构、循环结构      B. if、switch、break  
C. for、while、do-while      D. if、for、continue
8. 若有如下程序段  

```
int j=10;
while(j>0) j--;

```

则下面描述中，正确的是\_\_\_\_\_。  
A. while 循环执行 10 次      B. 循环是无限循环  
C. 循环体语句执行一次      D. 循环体语句一次也不执行
9. 以下程序段中循环体的执行次数是\_\_\_\_\_。  

```
x=10; y=0;
do{ y+=2;
    x-=2+y;
}while(x>=0);

```

  
A. 1      B. 2      C. 3      D. 4
10. 以下能对一维数组 a 进行正确初始化的语句是\_\_\_\_\_。  
A. `int a[10]=(0,0,0,0,0);`      B. `int a[ ]={0};`  
C. `int a[10]={ };`      D. `int a[10]={10*1};`
11. 若有定义： `int a[2][3],*p[3];`，则以下语句中，正确的是\_\_\_\_\_。  
A. `p[0]=&a[1][2];`      B. `p=a;`      C. `p[0]=a;`      D. `p[1]=&a;`

12. 下面的说明中，错误的是\_\_\_\_\_。
- A. `char a[10] = "china";`      B. `char a[10], *p=a; p="china";`  
 C. `char *a; a="china";`      D. `char a[10], *p; p=a="china";`
13. 以下程序中调用 `scanf` 函数给变量 `a` 输入数值的方法是错误的，其错误的原因是\_\_\_\_\_。
- ```
main()
{
    int *p,*q,a,b;
    p=&a;
    printf("input a:");
    scanf("%d",*p);
    .....
}
```
- A. `*p` 表示的是指针变量 `p` 的地址      B. `*p` 表示的是变量 `a` 的值，而不是变量 `a` 的地址  
 C. `*p` 表示的是指针变量 `p` 的值      D. `*p` 只能用来说明 `p` 是一个指针变量
14. 以下对 C 语言函数的有关叙述中，正确的是\_\_\_\_\_。
- A. C 语言程序由一个或者多个函数组成      B. C 语言函数既可以嵌套定义，也可以递归定义  
 C. 函数必须有返回值，否则不能使用函数      D. 在 C 语言程序中，存在调用关系的所有函数必须放在同一个源程序文件中
15. 以下关于形参或实参的叙述中，错误的是\_\_\_\_\_。
- A. 实参可以是常量、变量或表达式      B. 形参可以是任意类型  
 C. 形参可以是常量、变量或表达式      D. 实参应与其对应的形参类型一致
16. 下面的函数调用语句含有的实在参数的个数是\_\_\_\_\_。
- ```
func((exp1,exp2),(exp3,exp4,exp5));
```
- A. 5      B. 4      C. 3      D. 2
17. 以下在任何情况下计算平方数时都不会引起二义性的宏定义是\_\_\_\_\_。
- A. `#define POWER(x) x*x`      B. `#define POWER(x) (x)*(x)`  
 C. `#define POWER(x) ((x)*(x))`      D. `#define POWER(x) (x*x)`
18. 在 C 语言中，结构体类型变量在程序执行期间\_\_\_\_\_。
- A. 所有成员一直驻留在内存中      B. 只有一个成员驻留在内存中  
 C. 部分成员驻留在内存中      D. 没有成员驻留在内存中
19. `fscanf` 函数的正确调用形式是\_\_\_\_\_。
- A. `fscanf(文件指针, 格式字符串, 输出列表);`      B. `fscanf(格式字符串, 输出列表, 文件指针);`  
 C. `fscanf(格式字符串, 文件指针, 输出列表);`      D. `fscanf(文件指针, 格式字符串, 输入列表);`
20. 在执行 `fopen` 函数时，`ferror` 函数的初值是\_\_\_\_\_。
- A. -1      B. 0      C. 1      D. TURE

## 六、填空题

1. 下列程序运行后的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
main()
{
    int x=9;
    for( ; x>0; x--) {
```

```
if(x%3==0){  
    printf("%d", -x);  
    continue;  
}  
}
```

2. 下列程序运行后的输出结果是\_\_\_\_\_。

```
main()
{    int m=4,x=3,y=2,z=1;
    printf("%d",m<x ? m:z<y ? z:x);
}
```

3. 若输入 60 和 13, 以下程序运行后的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
#define SURPLUS(x,y) ((x)%(y))
main()
{
    int x,y;
    scanf("%d,%d",&x,&y);
    printf("%d",SURPLUS(x,y));
}
```

4. 下列程序中有错误的行是第\_\_\_\_行。 (注: 行号在注释中标出)

```
#include <stdio.h>
main()
{
    int a[10],k,x,*p,*q;
    scanf("%d",&x);
    for(k=0;k<10;k++)
        scanf("%d",a+k);
    p=a;
    *q=x;
    for(k=0;k<10;k++)
        if(*p++==*q)
            break;
    printf("%d",k+1);
}
```

5. 下列程序的功能是\_\_\_\_\_。

```
#include <stdio.h>
main()
{
    FILE *fp1,*fp2;
    fp1=fopen("d1.dat","r");
    fp2=fopen("d2.dat","w");
    while(!feof(fp1))
        fputc(fgetchar(fp1),fp2);
    fclose(fp1);
    fclose(fp2);
}
```

## 七、程序设计题

请编写求 N 个完全数的程序。

所谓完全数是一个整数，该整数等于除自身以外的所有约数之和。例如：6是一个完全数，因为 $6=1+2+3$ ；28也是一个完全数，因为 $28=1+2+4+7+14$ 。

**约定：**通过键盘输入  $N$  的值，并且  $N=3$ 。

## 八、程序设计题

请编写一 C 程序，该程序先通过键盘输入获得若干行字符(行数也通过键盘输入；每一行长度不相等)，并且依次将它们存储到一磁盘文件中，然后再从该磁盘文件中依次读出这些数据，将其中的小写字母均转换成大写字母后在屏幕上输出。

## 参考答案：

### 一、填空题

- |                     |  |   |
|---------------------|--|---|
| 1. 顺序存储结构           | 2. $p \rightarrow \text{link} = \text{list}; \text{list} = p;$ | 3. “删除栈底元素”                                   |
| 4. $BT[2i+1]$       | 5. 中序  | 6. $n^2 - 2e$                                 |
| 7. 1                | 8. $\lfloor \log_2 n \rfloor + 1$                              |   |
| 9. 利用它进行散列发生冲突的可能性小 |  | 10. an, bai, tang, wang, shi, deng, fang, liu |

### 二、简答题

1. 答: ① 存储空间动态分配, 根据实际需要使用, 可以做到尽可能节约空间;  
② 不要求地址连续的存储空间;  
③ 插入/删除操作只须通过修改指针实现, 不必移动数据元素, 操作的时间效率较高。

2. 答: 当图中出现权值相同的边时, 利用普里姆(Prim)算法与利用克鲁斯卡尔(Kruskal)算法可能生成不同的最小生成树。

3. 答: 相同点: 从逻辑上来看, 堆栈和队列都是特殊的线性表; 从操作的角度来看, 堆栈和队列的基本操作分别都是一般线性表的操作的子集, 因此, 他们都是操作受限制的线性表。

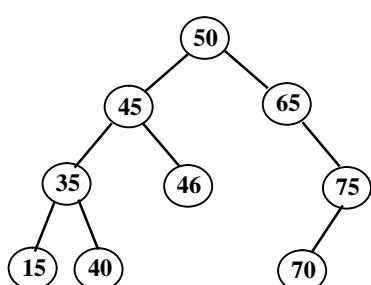
不同点: 堆栈将插入和删除操作限制在表尾(栈顶)进行, 元素进出表的特征是“后进先出”; 队列将插入和删除操作分别限制在表尾(队尾)和表头(队头)进行, 元素进出表的特征是“先进先出”。

4. 答: 在待排序的原始序列中元素已经按值从小到大排好序的情况下, 快速排序法的时间效率会变得很差, 因为在排序过程中, 每次选取的“分界元素”都是当前序列的最小值(最前那个元素), 经过一趟排序后, 将原序列分解成为一个空序列和一个原序列长度仅减 1 的子序列, 这样, 对于长度为  $n$  的原始序列, 必须经过  $n-1$  趟排序才能把所有元素定位, 而且第  $i$  趟排序需要经过  $n-1$  次元素之间的比较才能为第  $i$  个元素定位, 因此, 总的排序时间达到  $O(n^2)$ 。

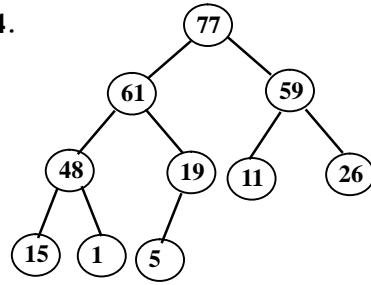
### 三、综合题

1.  $p \rightarrow \text{link} = \text{top}; \text{top} \neq \text{NULL} \quad \text{top} = \text{top} \rightarrow \text{link};$
2. 深度优先搜索序列是 ABCDE; 广度优先搜索序列是 ABCED。

3.



4.



### 四、算法设计题

```
LinkList INSERT(LinkList list, ElemType)
{
    LinkList p,q,r;
    p=(LinkList)malloc(sizeof(LNode));           /* 申请一个新的链结点 */
    p->data=item;
    if(item<list->data){                      /* 若 item 小于第一个结点 */
        p->link=list;                          /* 将新结点插在链表最前面 */
        list=p;
    }
    else{
        q=list;
        while(q!=NULL && item>q->data){      /* 寻找插入位置 */
            r=q;
            q=q->link;
        }
        p->link=q;
    }
}
```

```

        r->link=p;                                /* 将新结点插入 q 指结点后面 */
    }
    return list;
}

```

## 五、单项选择题

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

## 六、填空题

1. 852            2. 1            3. 8            4. 第 8 行

5. 将文件 d1.dat 的内容复制到文件 d2.dat 中

## 七、程序设计题

```

main( )
{
    int i,j,m,n,p;
    printf("Input N=");
    scanf("%d",&n);
    if(n<1)
        exit(0);
    for(m=0,i<6;m<n;i++){
        for(p=0,j=1;j<=i/2;j++)
            if(i%j==0)
                p+=j;
        if(p==i){
            m++;
            printf("%5d",i);
        }
    }
}

```

## 八、程序设计题

```

#include <stdio.h>
#define N 100
main( )
{
    FILE *fp;
    char c,*p,s[N][20];
    int I,n;
    printf("Input n=");
    scanf("%d",&n);
    if(n<1||n>N)
        exit(0);
    printf("Input %d string:\n",n);
    for(i=0;i<n;i++)
        scanf("%s",s[i]);
    fp=fopen("test","w");
    for(i=0;i<n;i++){
        p=s[i];
        while(*p!='\0')
            if(!ferror(fp))
                fputc(*p++,fp);
    }
    fclose(fp);
    printf("\n");
    fp=fopen("test","r");
    while((c=fgetc(fp))!=EOF){
        if(c>='a' && c<='z')
            c-=32;
        putchar(c);
    }
    printf("\n");
    fclose(fp);
}

```

**北京航空航天大学软件学院**  
**2010 年 4 月自主招收软件工程硕士研究生**  
**专业课试题**

**第一部分 数据结构**

**一、单项选择题**

1. 设非空单链表的结点构造为 

data	link
------	------

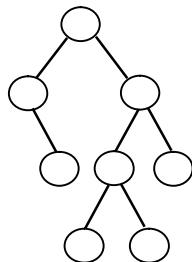
。若已知 q 指结点是 p 指结点的直接前驱，则在 q 与 p 之间插入由 s 所指结点的过程是依次执行\_\_\_\_\_。  
A.  $s \rightarrow link = p \rightarrow link$ ;  $p \rightarrow link = s$ ;      B.  $p \rightarrow link = s \rightarrow link$ ;  $s \rightarrow link = p$ ;  
C.  $q \rightarrow link = s$ ;  $s \rightarrow link = p$ ;      D.  $p \rightarrow link = s$ ;  $s \rightarrow link = q$ ;
2. 若堆栈的进栈序列是 1,2,3,4，则下列 4 个序列中，不可能为该堆栈的出栈序列的是\_\_\_\_\_。  
A. 4,3,2,1      B. 3,2,4,1      C. 1,3,2,4      D. 3,1,2,4
3. 下列 4 种操作中，不是队列基本操作的是\_\_\_\_\_。  
A. 删除队尾元素      B. 删除队头元素  
C. 将队列置为一个空队列      D. 判断一个队列是否为空
4. 树型结构最适合用来描述\_\_\_\_\_。  
A. 线性结构      B. 层次结构      C. 网状结构      D. 任何结构
5. 下列关于二叉树的叙述中，正确的是\_\_\_\_\_。  
A. 非空二叉树的度不一定是 2  
B. 满二叉树一定是完全二叉树，完全二叉树也一定是满二叉树  
C. 已知二叉树的前序序列和后序序列可以惟一地确定该二叉树  
D. 二叉树只能采用二叉链表存储结构
6. 采用邻接表存储图所用的空间大小\_\_\_\_\_。  
A. 与图的顶点数和边数都有关      B. 与图的顶点数和边数都无关  
C. 只与图的顶点数有关      D. 只与图的边数有关
7. 若在线性表中进行顺序查找，则该线性表应该采用\_\_\_\_\_。  
A. 散列存储结构      B. 顺序存储结构  
C. 链式存储结构      D. 顺序存储结构或者链式存储结构
8. 下列 4 种排序中，不属于内排序方法的是\_\_\_\_\_。  
A. 归并排序      B. 堆积排序      C. 拓扑排序      D. 快速排序
9. 若序列(12,13,14,8,9,10,25,6,8)是采用下列排序方法之一得到的第 2 趟排序后的结果，则该排序方法只能是\_\_\_\_\_。  
A. 选择排序法      B. 插入排序法      C. 泡排序法      D. 二路归并排序法
10. 在参加排序的序列中元素按值基本有序的情况下，下列 4 种排序方法中，时间效率最差的是\_\_\_\_\_。  
A. Shell 排序法      B. 堆积排序法      C. 二路归并排序法      D. 快速排序法

**二、简答题**

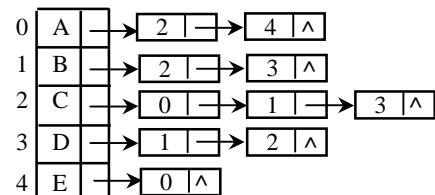
1. 线性表在什么情况下采用顺序存储结构比较合适?
2. 什么是递归算法? 通常情况下, 递归算法在执行过程中需要借助何种数据结构?
3. 若度为  $m$  且有  $n$  个结点的树采用多重链表存储结构, 即每个链结点设置  $m+1$  个域, 其中有 1 个数据域,  $m$  个指针域, 则该链表中空指针的数目是多少? 这种存储结构有何利弊?
4. 在长度为  $2^h-1$  的有序表中进行折半查找, 查找成功的情况下最多需要进行多少次元素之间的比较?

### 三、综合题

1. 若符号  $PUSH(k)$  表示整数  $k$  进栈, 符号  $POP$  表示栈顶元素出栈, 那么, 请画出依次执行  $PUSH(1)$ 、 $PUSH(2)$ 、 $POP$ 、 $PUSH(5)$ 、 $PUSH(7)$ 、 $POP$ 、 $PUSH(6)$  以后堆栈的状态。
2. 已知一棵二叉排序树的形状如图三—2 所示, 其结点的值分别为 1,2,3,4,5,6,7,8, 请在该二叉排序树中标出各结点的值。
3. 已知无向图采用邻接表存储, 邻接表如图三—3 所示。请分别写出从顶点 A 开始进行深度优先遍历与广度优先遍历后得到的遍历序列。



图三—2



图三—3

4. 请根据大顶堆积的定义, 写出对序列(26,5,77,1,61,11,59,15,48,19)进行堆积排序第 1 趟排序结束时序列的状态。(注: 按序列中元素的值从小到大排序)

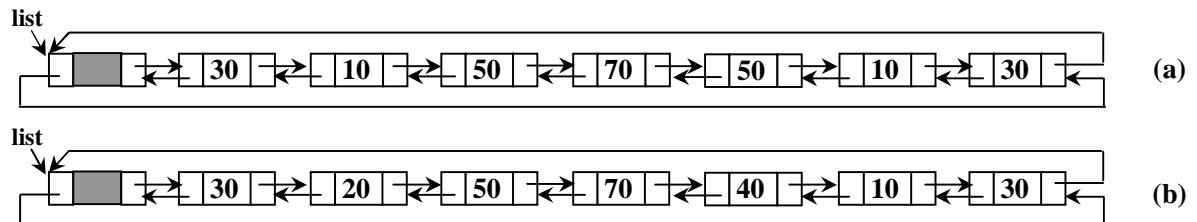
### 四、算法设计题

已知带有头结点的非空双向循环链表的链结点构造为 

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

, 头结点指针为 list, 请写一算法, 判断该双向循环链表是否对称, 若对称, 算法返回 1, 否则, 返回 0。

**说明:** 所谓链表对称是指除头结点外, 链表中前后所有对称位置的结点的数据域值相同。例如下面的两个链表中, (a)是对称的, 而(b)不是对称的。



**限制:** 算法中不得使用除指针变量和控制变量以外的其他辅助空间。

## 第二部分 C 语言程序设计

### 五、填空题

1. 以下函数的功能是按照从大到小的顺序输出两个整数。请分别在程序的空白处(横线上方)

填入一条语句或者一个表达式。

```
void FUN1(int a,int b)
{
    int temp;
    if( ① ){
        temp=a;
        ②
    }
    printf("%d,%d",a,b);
}
```

2. 下面程序的功能是输出 100 以内能够被 3 整除且个位数为 6 的所有正整数。请分别在程序的空白处(横线上方)填入一个表达式。

```
#include <stdio.h>
main()
{
    int i,j;
    for(i=0; ①;i++){
        j=i*10+6;
        if( ② )
            continue;
        printf("%5d",j);
    }
}
```

3. 下面给出的函数 strcat(s1,s2)的功能是实现将字符串 s2 拼接到字符串 s1 后面。请分别在程序的空白处(横线上方)填入一个表达式。

```
char *strcat(char *s1,char *s2)
{
    char *t=s1;
    while( ① )
        s1++;
    while( ② );
    return t;
}
```

4. 若已定义

```
struct num{
    int a;
    int b;
    float f;
} n={1,3,5.0};
struct num *p=&n;
```

则表达式 p->b/n.a\*++p->b 的值是 ①， 表达式(\*p).a+p->f 的值是 ②。

5. 以下程序的功能是先通过键盘输入一个文件名，然后把从键盘输入的字符依次存放到该文件中 (用符号#作为输入结束标志)。请分别在程序的空白处(横线上方)填入合适的内容。

```
#include <stdio.h>
main()
{
    FILE *fp;
    char ch,filename[10];
    printf("Input the name of file\n");
    gets(filename);
    if((fp= ① ))==NULL{
        printf("Cannot open file!\n");
        exit(0);
    }
    printf("Enter data\n");
    while((ch=getchar())!='#')
        fputc( ② ,fp);
    fclose(fp);
}
```

## 六、综合题

1. 以下程序的输出结果是\_\_\_\_\_。

```
main()
{
    int x=2,y=7,z=5;
    switch(x>0){
        case 1: switch(y<0){
            case 1: printf("@");
            break;
            case 2: printf("!");
            break;
        }
        case 0: switch(z==5){
            case 0: printf("*");
            break;
            case 1: printf("#");
            break;
            default: printf("#");
            break;
        }
        default : printf("&");
    }
    printf("\n");
}
```

2. 对于以下程序,

```
main()
{
    int a[5],*p;
    int k;
    for(k=0,p=a;k<5;k++,p++)
        scanf("%d",p);
    for(;k>0;k--)
        printf("%d",*(--p));
}
```

若输入为: 1 2 3 4 5<CR>(注: <CR>表示回车), 则输出结果是\_\_\_\_\_。

3. 下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
#include <string.h>
main()
{
    char *p1,*p2,str[50]= "*$#";
    p1="Beijing";
    p2="Shanghai";
    strcpy(str+1,strcat(p2,p1));
    printf("%s\n",str);
}
```

4. 下列程序的输出结果是\_\_\_\_\_。

```
struct ks{
    int a;
    int *b;
}s[4],*p;
main()
{
    int n=1,k;
    for(k=0;k<4;k++){
        s[k].a=n;
        s[k].b=&s[k].a;
        n+=2;
    }
    p=&s[0];
    p++;
    printf("%d,%d\n",(*p)->a,(*p)->a);
}
```

5. 若有以下宏定义:

```
#define N 2
#define Y(n) ((N+1)*n)
```

则执行赋值语句  $z=2*(N+Y(5));$  后, 变量  $z$  的值是\_\_\_\_\_。

## 七、程序设计题

请编写一程序, 该程序对于输入的字符串 (该字符串包含数字字符和非数字字符), 如:

ab123xy45? 2010nian4yue &05

将串中连续的数字作为一个整数依次存放到数组  $a$  中, 例如, 将 123 存放在  $a[0]$  中, 45 存放在  $a[1]$  中, 将 2010 存放在  $a[2]$  中, ……; 统计这些整数的数目, 并输出这些整数。

## 八、程序设计题

请编写一程序, 该程序的功能是对命令行中指定的两个文本文件进行比较, 并打印两个文件首次不同的行和该行中第一个不相同字符的位置。

**提示:** 文件中用‘\n’标记一行的结束。

## 参考答案:

### 一、单项选择题

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

### 二、简答题

1. 答: 当线性表经常进行的操作是查找而很少进行插入和删除操作, 并且表中元素的最大数量已知的情况下, 线性表采用顺序存储结构比较合适。

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。递归算法在执行过程中通常需要借助于堆栈这种数据结构来完成。

3. 答: 整个链表一共有  $n \times m$  个指针域, 除根结点外, 每一个结点都有一个指针指向它, 故链表中空的指针域数目为  $n \times m - (n-1) = n \times (m-1) + 1$  个。

采用这种存储结构的优点是结构统一, 便于操作, 缺点是空的指针域较多, 造成存储效率低。

4. 答: 设有序表的长度为  $n=2^h-1$ 。对长度为  $n$  的有序表进行折半查找, 其折半查找过程对应的判定树的深度为  $\lfloor \log_2 n \rfloor + 1 = h$ , 因此, 在查找成功的情况下最多需要进行  $h$  次元素之间的比较。

### 三、综合题

1.

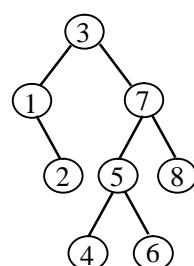
1	5	6		
栈底元素		栈顶元素		

3. 深度优先序列: ACBDE    广度优先序列: ACEBD

4.

原 始	26, 5, 77, 1, 61, 11, 59, 15, 48, 19
第 1 趟后	5, 61, 59, 48, 19, 11, 26, 15, 1, 77

2.



### 四、算法设计题

```
int SYMMETRY(DLinkList list)
{
    DLinkList p,q;
    p=list->rlink; /* p 指向头结点后的第 1 个结点 */
    q=list->llink; /* q 指向链表最右边那个结点 */
    while(p!=q && p->rlink!=q){
        if(p->data==q->data){
            p=p->rlink; /* p 右移到后一个结点 */
            q=q->llink; /* q 左移到前一个结点 */
        }
    }
}
```

```

        q=q->llink;           /* q 左移到前一个结点 */
    }
    else
        return 0;           /* 该链表不是对称的 */
    }
    return 1;           /* 该链表是对称的 */
}

```

## 五、填空题

- |                |                          |                     |
|----------------|--------------------------|---------------------|
| 1. ① a<b       | 2. ① i<10                | 3. ① *s1 或 *s1!='0' |
| ② a=b; b=temp; | ② j%3!=0                 | ② *s1++=*s2++       |
| 4. ① 12        | 5. ① fopen(filename,“w”) |                     |
| ② 6.0          | ② ch                     |                     |

## 六、简答题

1. #&      2. 54321      3. \*ShanghaiBeijing      4. 7,3      5. 34

## 七、程序设计题

```

#include <string.h>
main()
{
    char s[80],*ptr=s;
    int i,j=0,num=0,flag=0,a[40];
    printf("Input string:");
    gets(s);                                /* 输入一字符串 */
    for(i=0;i<strlen(s);i++,ptr++)
        if(*p>='0'&&*p<='9'){
            flag=1;                          /* 标志 flag 为 1 表示处理完毕一个整数 */
            num=10*num+*p-'0';
        }
        else if(flag){
            a[j++]=num;
            flag=0;
            num=0;
        }
    if(flag)                                /* 此时 flag 为 1 表示串的最后是一个正数 */
        a[j++]=num;
    printf("Total:%d\n",j);
    for(i=0;i<j;i++)
        printf("%5d",a[i]);
    printf("\n");
    getch();
}

```

## 八、程序设计题

```

#include <stdio.h>
main(int argc,char *argv[ ])
{
    FILE *fp1,*fp2;
    int line=0,count=0;
    char c1,c2;
    if(argc>2){
        fp1=fopen(argv[1], "r");
        fp2=fopen(argv[2], "r");
        if(fp1!=NULL && fp2!=NULL)
            while((c1=getc(fp1))!=EOF){
                c2=getc(fp2);
                if(c1==c2)
                    count++;
                else
                    break;
                /* 从第 1 个文件中读取一个字符 */
                /* 从第 2 个文件中读取一个字符 */
                /* 比较对应字符 */
                /* 字符位置记数 */
            }
    }
}

```

```
        if(c1=='\n')
            line++;
        /* 行记数 */
    }
    else{
        printf("Input filename is not exist!\n");
        exit(1);
    }
    if(c1==EOF)
        printf("Two files is same!\n"); /* 两个文件相同 */
    else
        printf("Two files is not same. Line=%d char_num=%d\n",line,count);
    fclose(fp1);
    fclose(fp2);
}
else{
    printf("Usage:comppfile filename1 filename2\n");
    exit(1);
}
```

北京航空航天大学软件学院  
2010年7月自主招收软件工程硕士研究生  
专业课试题

## 第一部分 数据结构

## 一、单项选择题

1. 下面关于线性表的叙述中, 错误的是\_\_\_\_。

  - A. 线性表采用顺序存储结构, 必须占用一片地址连续的存储空间
  - B. 线性表采用顺序存储结构, 便于进行插入和删除操作
  - C. 线性表采用链式存储结构, 不必须占用一片地址连续的存储空间
  - D. 线性表采用链式存储结构, 便于进行插入和删除操作

2. 设非空单链表的结点构造为 

data	link
------	------

。若要删除该链表中 p 指结点的后面那个结点存在), 则需要执行的操作是\_\_\_\_。(不含被删除结点的空间释放)

  - A.  $p=p->link->link;$
  - B.  $p->link=p;$
  - C.  $p=p->link;$
  - D.  $p->link=p->link->link;$

3. 堆栈与队列的共同点是\_\_\_\_。

  - A. 元素的进/出满足“先进先出”的规律
  - B. 元素的进/出满足“先进后出”的规律
  - C. 只允许在表的端点处进行插入和删除元素的操作
  - D. 不存在共同点

4. “二叉树为空”意味着\_\_\_\_。

  - A. 二叉树由一些未赋值的空结点组成
  - B. 二叉树的根结点没有子树
  - C. 该二叉树不存在
  - D. 该二叉树没有结点

5. 若一棵满二叉树有 2047 个结点, 则该二叉树中叶结点的个数是\_\_\_\_。

A. 512

B. 1024

C. 2048

D. 4096

6. 若从无向图中任意一个顶点出发进行 1 次深度优先搜索便可以访问到该图的所有顶点，则该图一定是一个\_\_\_\_\_。

A. 非连通图

B. 强连通图

C. 连通图

D. 完全图

7. 对采用邻接表方法存储的图进行广度优先搜索的过程中用到的一个关键数据结构是\_\_\_\_\_。

A. 队列

B. 堆栈

C. 二叉树

D. 图

8. 散列技术中的散列冲突是指\_\_\_\_\_。

A. 两个元素具有相同的序号

B. 两个元素的键值不同，而其他属性相同

C. 元素过多

D. 不同键值的元素对应着相同的存储地址

9. 与直接插入排序法比较，折半插入排序法减少了排序过程中的\_\_\_\_\_。

A. 排序总的趟数

B. 元素的移动次数

C. 元素之间的比较次数

D. 使用的辅助空间的数量

10. 下面给出的四种排序法中，排序过程中元素之间的比较次数与排序法无关的是\_\_\_\_\_。

A. 选择排序法

B. 插入排序法

C. 快速排序法

D. 堆积排序法

## 二、简答题

1. 若 5 个元素的进栈序列是 a、b、c、d、e，利用堆栈操作能否得到出栈序列 b、c、a、e、d 和 d、b、a、c、e？对于不能得到的出栈序列，请说明理由。

2. 有人说：“在一棵二叉树中，对于除叶结点外的任意结点，如果其值大于它的左孩子结点(若存在)的值，并且小于或等于它的右孩子结点(若存在)的值，则该二叉树一定是二叉排序树”，该说法正确吗？若你认为不正确，请举一例说明。

3. 拓扑排序的主要功能是什么？对于一个存在拓扑序列的有向图，通过拓扑排序得到的拓扑序列是否惟一？

4. 数据文件的基本操作有插入、删除、修改和查找等，请问：其中最基本的操作是哪一个？为什么？

## 三、综合题

1. 下列算法的功能是在由 list 所指的非空线性链表的第 i 个结点(假设链表中存在第 i 个结点，且  $i \geq 1$ )后面插入一个由 p 指的结点。

请在算法的空白处(横线上方)填上必要的内容，使算法完整。

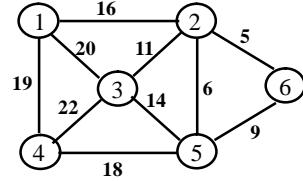
```
typedef struct node{  
    ELEMTYPE data;  
    struct node *link;  
} *LINKLIST; /* 定义线性链表类型 */  
void INSERTLINK(LINKLIST list, int i, LINKLIST p)  
{  
    LinkList q=list;  
    int j=1;  
    while( _____ ){  
        q=q->link;  
        j++;
```

```

    }
    /* 找到插入位置 */
    q->link=p;
}
/* 插入 */
/* 插入 */
}

```

2. 请任意指出两个在“数据结构”课程中学习过的算法，其中一个用到堆栈，另一个用到队列。(注：不必给出具体算法)  
 3. 请画出图三—3 所示的连通图的最小生成树。



图三—3

4. 请根据堆积(Heap)的定义，以二叉树的形式画出序列(26,5,77,1,61,11,59,15,48,19)对应的(大顶)堆积。

#### 四、算法设计题

请写一**非递归**算法，对于任意给定的  $k$  值，该算法在长度为  $n$ 、且元素按值严格递增排列的顺序表  $A[1..n]$  中采用**折半查找法**查找值不大于  $k$  的最大元素，若表中存在这样的元素，则算法返回该元素在表中的位置，否则，算法返回信息 0。(假设表中元素分别为一个正整数)

例如，对于顺序表  $A[1..10]=(2, 4, 6, 8, 10, 12, 14, 16, 18, 20)$ ，

当  $k=8$  时，满足条件的元素为 8，返回位置 4；

当  $k=13$  时，满足条件的元素为 12，返回位置 6；

当  $k=1$  时，无满足条件的元素，返回信息 0。

## 第二部分 C 语言程序设计

#### 五、填空题

1. 在 C 语言中，实型变量被分为两种类型，这两种类型分别是\_\_\_\_\_和\_\_\_\_\_。

2. 若  $x$  和  $y$  均为 int 类型的变量，则依次执行语句  $x+=y$ ;  $y=x-y$ ;  $x-=y$ ; 的效果是\_\_\_\_\_。(用文字描述)

3. 若  $a$  是 int 类型的变量，则描述“ $a$  是奇数”的 C 语言表达式是\_\_\_\_\_。

4. 若有 int  $x=3, y=4, z=5$ ;，则表达式  $!(x+y)+z-1 \&& y+z/2$  的值是\_\_\_\_\_。

5. 若 for 循环语句用以下形式表示：

for(表达式 1; 表达式 2; 表达式 3)

循环体语句

则执行  $for(i=0;i<3;i++) \text{printf}("*)$ ; 时，表达式 1 执行了\_\_\_\_\_次，表达式 3 执行了\_\_\_\_\_次。

6. 若有定义： int  $x[3][4]=\{\{1,2\},\{0\},\{4,6,8,10\}\}$ ;，则初始化后  $x[1][2]$  得到的初值是\_\_\_\_\_。

7. 在 C 语言中，一个函数由两个部分组成，它们分别是\_\_\_\_\_和\_\_\_\_\_。

8. 若有定义： int  $a[]=\{2,4,6,8,10,12\}$ , \* $p=a$ ;，则  $*(p+1)$  的值是\_\_\_\_\_。

9. 若已有如下宏定义：

`#define MIN(x,y) (x)>(y)?(x):(y)`

以及定义： int a=1,b=3,c;， 则执行语句 c=MIN(a=b,b-a);以后， 变量 c 的值是\_\_\_\_\_。

10. 下列程序的功能是统计一个文本文件中的字符数量。请在程序的空白处(横线上方)填上必要的内容，使之完整。

```
#include <stdio.h>
main()
{
    FILE *fp;
    long int num=0;
    if((fp=(“data”,“r”))==NULL){
        printf(“Can not open file.\n”);
        exit(0);
    }
    while( _____ )
        num++;
    _____;
    printf(“num=%ld\n”,num);
}
```

## 六、程序或程序段阅读

1. 下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
main()
{
    int x,y,z,w;
    x=y=z=0;
    w=5;
    if(!x&&w--)
        w--;
    if(z&&!w--)
        w=15;
    else
        w=10;
    printf(“%d”,w);
}
```

2. 下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
main()
{
    int x=1,y=10;
    do{
        y-=x;
        x++;
    }while(y--<0);
    printf(“x=%d,y=%d\n”,x,y);
}
```

3. 下列程序段的输出结果是\_\_\_\_\_。

```
char str[ ]=“abc\0def\0ghi”, *q=str;
printf(“%s”,q+5);
```

4. 下列程序的输出结果是\_\_\_\_\_。

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

```

#include <string.h>
main( )
{
    int i=0;
    char str1[10]={"1234"}, str2[10]={"567"};
    strcat(str1,str2);
    while(str2[i++]!=='\0')
        str2[i]=str1[i];
    puts(str2);
}

```

5. 下列程序的输出结果是\_\_\_\_\_。

```

struct ss{
    int a;
    struct ss *b;
} t[5];
main( )
{
    int i;
    for(i=0;i<5;i++){
        t[i].a=i*i;
        t[i].b=&t[i+1];
    }
    t[i].b=t;
    printf("%d\n",*t[3].b);
}

```

6. 对于如下程序：

```

#include <stdio.h>
main( )
{
    FILE *fp;
    fp=fopen("file.txt","w");
    fprintf(fp,"%s","xyz");
    fclose(fp);
}

```

若文件 file.txt 中原有的内容为 good，则运行该程序以后，文件 file.txt 中的内容为\_\_\_\_\_。

## 七、程序补充题

下列程序的功能是根据如下计算公式计算 sum 的值，请将程序中名为 calculate 的函数补充完整。

$$\text{sum} = 1 + \frac{1}{1+2} + \frac{1}{1+2+3} + \dots + \frac{1}{1+2+3+\dots+n}$$

```

float calculate (int n);
main( )
{
    int n;
    float sum;
    printf("\nInput a integer number n(>=1):\n");
    scanf("%d",&n);
    sum=calculate(n); /* 调用名为 calculate 的函数计算 sum 的值 */
    printf("The result is :%f\n",sum);
}

```

```

float calculate (int n)
{ /* 请在下面补充函数体 */
}

```

## 八、程序设计题

请设计一 C 语言函数(注: 只要求写出该函数, 不要求写出完整程序), 该函数的功能是将一个 int 类型的数组 A[0..n-1]的所有元素循环右移 k 个位置。

例如, 对于某数组, 当 k=3(即把数组所有元素循环右移 3 位)时, 是将

10	20	30	40	50	60	70	转换为	50	60	70	10	20	30	40
0	1	2	3	4	5	6		0	1	2	3	4	5	6

### 参考答案:

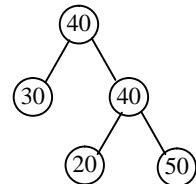
#### 一、单项选择题

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

#### 二、简答题

1. 答: 能够得到出栈序列 b、c、a、e、d, 但不能得到出栈序列 d、b、a、c、e。因为若出栈序列以元素 d 开始, 则说明在 d 之前的进栈元素有 a、b 和 c, 三个元素中 c 是此刻的栈顶元素, b 和 a 不可能先于元素 c 出栈, 因此, 不可能得到出栈序列 d、b、a、c、e。

2. 答: 此说法不正确。二叉排序树的定义是一个递归定义, 要求其左、右子树都是二叉排序树, 因此, 该说法不符合定义。以右图所示的二叉树为例, 它虽然满足题目中的条件, 但它不是二叉排序树。



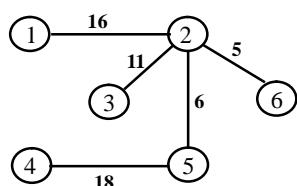
3. 答: 拓扑排序的主要功能是检测一个有向图中是否存在回路。对于一个存在拓扑序列的有向图, 通过拓扑排序得到的拓扑序列不一定唯一。

4. 答: 数据文件最基本的操作是查找, 插入、删除和修改等操作都是建立在查找操作之上的, 因为在进行这些操作之前都需要先通过查找操作来确定操作的位置。

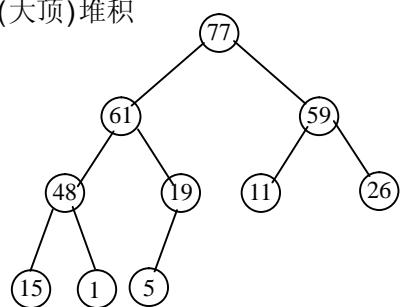
#### 三、综合题

1. ①  $j < i$     ②  $p->link = q->link;$   
 2. 二叉树的遍历、图的深度优先搜索等用到堆栈; 二叉树的按层次遍历、图的广度优先搜索等用到队列。

#### 3. 最小生成树



#### 4. 初始(大顶)堆积



#### 四、算法设计题

```
int BINSEARCH(int A[ ],int n,int k)
{
    int low=1, high=n, mid;
    while(low<=high){
        mid=(low+high)/2;
        if(A[mid]==k)
            return mid; /* 返回 mid */
        if(k>A[mid])
            low=mid+1; /* 准备查找后半部分 */
        else
            high=mid-1; /* 准备查找前半部分 */
    }
    return high; /* 返回信息 0 */
}
```

#### 五、填空题

1. 单精度类型(或 float 型) 双精度类型(或 double 型)
2. 不借助任何中间辅助变量交换变量 x 和 y 的值
3. (a%2)==1      4. 1      5. 1 3      6. 0      7. 函数说明(部分) 函数体
8. 4      9. 3      10. fgetc(fp)!=EOF      fclose(fp)

#### 六、程序或程序段阅读

1. 10
2. x=2,y=8
3. ef
4. 5234567
5. 16
6. xyz

#### 七、程序补充题

```
float fun(int n)
{
    int i;
    float temp=0.0, sum=0.0;
    for(i=1;i<=n;i++){
        temp=temp+i;
        sum=sum+1.0/temp;
    }
    return sum;
}
```

#### 八、程序设计题

```
void CIRCULATE(int A[ ], int n, int k)
{
    int t,j;
    int temp;
    for(t=1;t<=k;t++){
        temp=A[n-1];
        for(j=n-2;j>=0;j--)
            A[j+1]=A[j];
        A[0]=temp;
    }
}
```

或

```
void CIRCULATE(int A[ ], int n, int k)
{
    int t,j;
    int temp;
    for(t=1;t<=k;t++)/* 每循环 1 次，所有元素依次循环右移 1 位 */
    {
        temp=A[n-1];
        /* 保存数组最后那个元素 */
        for(j=n-2;j>=0;j--)
            A[j+1]=A[j];
        /* 元素循环右移 1 位 */
        A[0]=temp;
    }
}
```

```
temp=A[0];           /* 保存数组第 1 个元素 */
for(j=n-1;j>0;j--)
    A[(j+1)%n]=A[j]; /* 元素循环右移 1 位 */
A[1]=temp;
}
}
```

建议选用 2010 年 7 月  
第 6 次印刷的书



### 指定参考书：

1. 《数据结构教程 第二版》唐发根 编著 北京航空航天大学出版社 2005
2. 《C 程序设计 第三版》 谭浩强 编著 清华大学出版社 2005

### 推荐使用

《数据结构教程 第二版》

唐发根 编著 北京航空航天大学出版社 2005

(2010 年 7 月第 6 次印刷)