

2020 浙江大学软件工程考研

历年真题汇总及习题解答



学校_____

姓名_____

目录

2001 年攻读硕士学位研究生入学考试试题	1
2002 年攻读硕士学位研究生入学考试试题	4
2003 年攻读硕士学位研究生入学考试试题	7
2004 年攻读硕士学位研究生入学考试试题	9
2005 年攻读硕士学位研究生入学考试试题	12
2006 年攻读硕士学位研究生入学考试试题	16
二〇〇七年攻读硕士学位研究生入学考试试题	19
二〇〇八年攻读硕士学位研究生入学考试试题	25
二〇一〇年攻读硕士研究生入学考试试题	30
二〇一一年攻读硕士研究生入学考试试题	34
二〇一二年攻读硕士研究生入学考试试题	38
二〇一三年攻读硕士研究生入学考试试题	43
二〇一七年攻读硕士研究生入学考试试题	46
二〇一八年攻读硕士研究生入学考试试题	47
2001 年攻读硕士学位研究生入学考试试题答案	53
2002 年攻读硕士学位研究生入学考试试题答案	55
2003 年攻读硕士学位研究生入学考试试题答案	57
2004 年攻读硕士学位研究生入学考试试题答案	59
2005 年攻读硕士学位研究生入学考试试题答案	60
2006 年攻读硕士学位研究生入学考试试题答案	61
2007 年攻读硕士学位研究生入学考试试题答案	62
2008 年攻读硕士学位研究生入学考试试题答案	66
2010 年攻读硕士学位研究生入学考试试题答案	68
2011 年攻读硕士学位研究生入学考试试题答案	71
2012 年攻读硕士学位研究生入学考试试题答案	74
2013 年攻读硕士学位研究生入学考试试题答案	77
2000~2001 学年《程序设计基础-C》试卷秋季	79
2000~2001 学年下《程序设计基础 C》秋季答案	84

2000~2001 学年《程序设计基础-C》试卷春季.....	86
2000~2001 学年下《程序设计基础 C》春季答案.....	91
2000~2001 学年《程序设计基础-C》试卷上学期.....	93
2000~2001 学年下《程序设计基础 C》上学期答案.....	99
2001~2002 学年《程序设计基础-C》试卷.....	101
2001~2002 学年《程序设计基础 C》答案.....	106
2002~2003 学年《程序设计基础-C》试卷.....	109
2002~2003 学年《程序设计基础 C》答案.....	116
2003~2004 学年《程序设计基础-C》试卷.....	117
2003~2004 学年《程序设计基础 C》答案.....	123
2004~2005 学年《程序设计基础-C》试卷.....	125
2004~2005 学年《程序设计基础 C》答案.....	130
2005~2006 学年《程序设计基础-C》试卷.....	131
2005~2006 学年《程序设计基础 C》答案.....	138
2006~2007 学年《程序设计基础-C》试卷.....	140
2006~2007 学年《程序设计基础 C》答案.....	148
2007~2008 学年《程序设计基础-C》试卷.....	151
2007~2008 学年《程序设计基础 C》答案.....	160
2008~2009 学年《程序设计基础-C》试卷.....	162
2008~2009 学年《程序设计基础 C》答案.....	172



浙江大学

2001 年攻读硕士学位研究生入学考试试题

注意：答案必须写在答题纸上，写在试题纸或草稿上均无效。

一、根据下列各题要求填空(每空 3 分， 24 分)：

1、设 `char s[10],*p;` 则下列最正确的语句为_____

- A) `p=s+4;` B) `*p=s[0];` C) `s="graduate";` D) `*s-*p;`

2、对于 `int a;` 设 `a>0,` 则表达式 `(a&3)==(a%4)` 值是_____

- A) 0 B) 1 C) 不定 D) 表达式语法有误

3、对于如下程序 `test.c`, 编译后运行 `test hello world` 将输出_____

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

```
main( int argc,char *argv[ ])
```

```
{
```

```
    printf("%d%s",argc,angv[ 1]+ 1);
```

```
}
```

4、定义 `int a[]={ 1,2,3,4}, y, *p=&a[2];` 执行 `y=(*--p)++` 后， `y` 的值为_____

5、用 `for` 循环将整型数组 `a(int a[100])` 按 5 个一行输出，其语句应为：

```
for (i=0; i<100; i++)      printf("%5d",a[i])_____
```

6、`int i,s,a[10];`

```
for(i=0 s=0; i<10; s+=a[i++])      if(a[i]<0) break;
```

等价于

```
int i=0,s=0,a[10];
```

```
while (i<10&&a[i]>0_____) s+=a[i++];
```

7、下面的函数 `strcat(char *s,char *t)` 将字符串 `t` 拼接到字符串 `s` 之后。

```
char *strcat(char *s,char *t){
```

```
    char *str=s;
```

```
    while (_____)      s++;
```

```
    while(_____)
```

```
        retum str;
```

```
}
```

二、阅读下列程序并回答相应问题(12 分):

1、(5 分) float px(float x,int n)

```
{    if(n==1)    return (x);  
    else return (x*(1-px(x,n-1)));  
}
```

问题：上述质数是计算某多项式 $px(x, n)$ 的值，请写出该多项式 $px(x, n) = \underline{\hspace{2cm}}$

2、(7 分)

```
#include <stdio.h>  
  
#include <stdlib.h>  
  
struct node {  
    char c;  
    struct node *next;  
};  
  
main()  
{  
    struct node *head=NULL, *p=NULL,*q, *t;  
    char *s="2-The Shining C-2";  
    while(*s){  
        q=(struct node *)malloc(sizeof(struct node));  
        q->c=*s; q->next=NULL;  
        if(head==NULL) head=q;  
        else p->next=q;  
        p=q;s++;  
    }  
    while(q!=NULL){  
        if(! ( q->c>='A' && q->c <='Z')){  
            if(q==head) head=q->next; /* Note */  
            else p->next=q->next; /* Note */  
            t=q; q=q->next; free(t); /* Note */  
        }  
    }
```

```
else {p=q; q=q->next )  
}  
  
q=head;  
  
while(q!=NULL) (putchar((*q).c); q=q->next;  
}
```

可题： 1、上述程序的运行结果为_____

2、程序中 Note 注解所对应的复合语句实现什么功能？



《浙大软件考研 878》微信公众号，扫描二维码关注，获取更多考研资讯
微信：tqh4546

浙江大学

2002 年攻读硕士学位研究生入学考试试题

注意：答案必须写在答题纸上，写在试题纸或草稿上均无效。

一、单项选择或填空题(每空 3 分，共 24 分)：

1. 对于 int a, 要使 $(1 << 2 >> 1) | a == a$; 则 a 可以是：

- A) 2 B) 6 C) 10 D) 2, 6, 10 均可

2. 数组定义为 int a[3][2]={1, 2, 3, 4 5, 6}, 数组元素 a[2][1]的值为_____。

- A) 3
B) 5
C) 6
D) 不确定

3. 不正确的赋值或赋初值的方式是。

- A) char st[]="abc";
B) char st[4]={'a' , 'b', 'c' };
C) char str[4];str="abc";
D) char st[4]={'a' , 'b', 'c' , '\0' };

4. 语句 print("%d%d",123); 将输出：

- A)%123%d
B)%d123
C)%d123
D) 上述语句语法有错

5. 循环 for(i=1; j=6; ++i!=--j;) s=itj; 将执行

- A) 2 次
B) 3 次
C) 4 次
D) 以上均不是

6. 执行程序 find-nx ould text.txt 时， *++argv[1]的值是_____

7. 设有变量定义 int i, j, k=5, *p; 则执行下列语句后， i, j, k 的值分别为: _____

p=&k; i=*p++; j==++*p;

8. 判断字符 c 为非小写字母的表达式(若非小写表达式值为 1 否则为 0)是_____。

二、下列程序读入无符号 16 进制数(出现的字母为小写)，将其转换为十进制数输出。请将程序空缺部分补全(6 分)。

```
int f(char *s){  
    int n=0,i;  
    for(i=0;s[i]!='\0'; i++)  
        n=n*16+_____;  
    return n;  
}  
  
main( ){  
    char s[10];  
    scanf("%s",s);  
    printf("%d\n",_____);  
}
```

阅读下列程序回答相应问题(8 分)：

```
/*函数 sprint(char *buffer, char *format, m)用用于将格式化输出写入从 buffer 所指  
位置开始的字符串中*/  
  
#include <stdio.h>  
#include <string.h>  
  
void f(int n,int m  char *s){  
    int i;  
    char t[80];  
    strcpy(t,s);  
    for(i=m;i<=n;i++)  
        if(n-i>=i){  
            sprint(t+strlen(s);” %d+” ,i);  
            f(n-i,i,t);  
        }  
    else if(n-i==0&&strlen(s)>0){
```

```
sprint(t+strlen(s);” %d ” ,i);
printf("%s\n",i);
}

main(){
    Char s[80]=” ” ;
    f(3,1,s);
}
```

1) 上述程序的运行结果是什么?

如果将上面的 $f(3, 1, s)$ 改为 $f(6, 2, s)$ 则结果又是什么?



浙江大学

2003 年攻读硕士学位研究生入学考试试题

注意：答案必须写在答题纸上，写在试题纸或草稿上均无效。

一、单项选择题或填空题。

1、下列 `for(i=0;(i<4)&&(j!=3);i++,j++);`

- A.无限循环 B.循环 0 次 C.循环 3 次 D.循环 4 次

2、设变量定义为 `char a[10],*p=a`, 则下列赋值语句中正确的是

- A.`a[10]= "hello world"` B.`a= "hello world"`
C.`*p= "hello world"` D.`Bp= "hello world"`

3、设有变量定义：`int a[6]={0,1,2,3,4,5}`, `*p=a, i,j;` 以下对 a 数组不正确的引用为

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

4、下列程序段的输出结果为_____

```
Char *s[]={"program","lang","java2","vc"};
Printf("%d %s,sizeof(s)/sizeof(char)",*(s+2)+1);
```

5、下面的 c 代码段用于计算 $s = x - x^3/3 + x^5/5 - x^7/7 + x^9/9 - \dots$ 的近似值，空缺部分的内容应该是

```
int i=1; double s,t;
```

```
t= x, S = t;
```

```
while (fabs(t) > 0.0001) { /* fabs(t):t 的绝对值 */
    i=i+2;
    t= t* _____;
    s += t;
}
```

6、执行下列语句后，`a` 的值是_____。

```
int x=4, y=25,z=5, a;
```

```
a= (-y/++x)*z--;
```

程序理解(10 分，每题 5 分)

1、#include <stdio.h>

```
#define macro(a,b) (a^=b^=a^=b);
```

```
void f(char a[])
```

```
{  
    int i j;  
    for (i=0,j=strlen(a)-1; i<j;i++,j--)  
        macro(a[i], a[j]);  
}  
  
main()  
{  
    char *s="1234567";  
    f(s);  
    print("%s\n",s);  
}
```

(1) 程序运行结果是_____。 (2) 宏 macro(a, b)的功能是_____。

2、#include <stdio.h>

```
void f(char *s )  
{  
    int j;  
    for(i=0,j=0; s[i]!='\0';i++)  
        if[s[i]!='t'&& s[i]!='n'] s[j++]=s[i];  
    s[j]='\0';  
}  
main(){  
    char *s="A Hello\t World\n";  
    f(s);  
    printf("%s\n", s);  
}
```

(1)程序的运行结果是_____。
(2)函数 f(char *s)的功能是_____。



浙江大学

2004 年攻读硕士学位研究生入学考试试题

注意：答案必须写在答题纸上，写在试题纸或草稿上均无效。

一、根据下列各题要求填空(相应选择为单选)(每空 3 分, 共 21 分)：

1、设 8 位二进制数 x, 若想通过位运算使 x 中的低 4 位不变, 高 4 位清零, 则以下能实现此功能的是:

- A) $x=x|0x0f;$ B) $x=x&0x0f;$ C) $x=x|0xf0;$ D) $x=x&0 xf0;$

2、以下不是无限循环的语句是

- A) `for(y=0,x=1; x>y++; x++);`
B) `for(; ;x++);`
C) `while(0) {x++;}`
D) `for(i=0; ;i--) sum+=i;`

3、若有说明： int a[4][10]; 则是对数组元素 a[i][j] 的错误引用

- A) `*(&a[0][0]+10*i+j)`
B) `*(a+i)[j];`
C) `*(*(a+i)+j)`
D) `*(a[i]+j)`

4、若有以下说明和语句

```
struct worker{
```

```
    int no;  
    char name[20];  
} work,*p=&work;
```

则以下语句中不正确的是_____

- A). `work.no=1234;` B). `(*P).no=1234;`
C). `p->name[0]='J';` D). `p->name="Jone";`

5、若有以下定义和说明, 以下不能将二进制文件内容读入数组 x 中的语句(组)是

```
struct ss{
```

```
    int n;  
    float x;
```

```
}x[30];  
  
FILE *fp;  
  
A). for(i=0; i<30; i++)fread(x[i],sizeof(stuct ss) 1,fp);  
B). for(i=0 ;i<10; i++;i++) fread(&x[i].sizeof(struct ss) 2,fp);  
C). for(i=0; i<10; i+=3) fread(x+i,sizeof(struct ss),3,fp) ;  
D). fread(x,sizeof(struct ss),30,fp);
```

6、下面程序的功能是用递归算法将一个整数按逆序存放到一个字符数组中。如 123 存放成 321. 请填空：

```
#include <stdio.h>  
  
void conver(char *a,int n)  
{  
    int i;  
    if(i=n/10) convert(_____,i);  
    *a=_____;  
}
```

```
main(){  
    int number;  
    char str[10]="";  
    scanf("%d",&number);  
    conven(str,number);  
    puts(str);  
}
```

二、程序理解(10 分, 每题 5 分)

1、以下程序的运行结果是

```
main(){  
    int a=2,k;  
    for(k=0;k<3;k++)printf("%d",f1(a));  
}  
  
f1(int a){  
    int b=0;
```

```
static int c=3;  
b++; c++;  
return (a+b+c);  
}
```

2、当将下列程序 test.c 编译后，运行 test Computer Science College Zhejiang University 时，输出结果是

```
main(int argc,char *argv[] )  
{  
    int i;  
    char c;  
    while (--argc>0){  
        c=(*++argv)[i];  
        if(c<='z'&&c>='a')  
            printf("%c",c+'A'-'a');  
        else print("%c",c+i);  
        i++;  
    }  
    return 0;  
}
```

三、语言编程(9分)。

请编写一个 C 语言程序。该程序读入 n，输出以下数列的前 n 项和：

1/2, 2/3, 3/5, 5/8, 8/13, 13/21,



《浙大软件考研 878》微信公众号，扫描二维码关注，获取更多考研资讯
微信：tqh4546

浙江大学

2005 年攻读硕士学位研究生入学考试试题

注意：答案必须写在答题纸上，写在试题纸或草稿上均无效。

一、根据下列各题要求选择（相应选择为单选） 或填空（每空 3 分共 18 分），

1. 表达式 $-1 < 0 < 1$ 的值是

- A) 0 B) 1 C) 3 D) 语法错误

2. 对于以下变量定义，哪个赋值是错误的？

```
struct node{  
    char s[10];  
    int i;  
}p[5];  
  
char *t="abc";  
  
A) p[0].i=*&+t-'c';  
B) (p+2)->s=t;  
C) (*p).i=2;  
D) *(p->s)=*(t+1);
```

3. 执行命令行“prog-nla zhejiang university” 时 $*++\text{argv}[1]$ 的值是多少
(argv 是 main 函数的第二个参数)：

- A) '-' B) 'z' C) 'u' D) 以上都不是

4. 执行下列语句后，其输出是：

```
int k,x;  
  
for(k=0,x=0;k<=9&&x!=10;k++)  
    x+=2;  
  
printf("%d,%d",k,x);
```

5. 下面程序用递归算法将一个实数字符串转换成相应的实数，如
 $\text{atof}(\text{"12.34"})$ 将返回 12.34。请将程序中的两个空缺填上：

```
#include<ctype.h>  
  
double atof(char *s)  
{
```

```
Double val,power;  
int i,sign;  
for(i=0;isspace(s[i]);i++);  
sign=(s[i]=='-?-1:1);  
if(s[i]=='+'||s[i]=='-');  
    i++;  
for(val=0.0;isdigit(s[i]);i++)  
  
_____  
if(s[i]=='.')i++;  
for (power=1.0;isdigit(s[i]);i++){  
    val=10.0*val+(s[i]-'0');  
_____  
}  
return sign*val/power;  
}
```

二、程序理解(10 分, 每题 5 分)

1. 对程序

```
#include<stdio.h>  
#include<stdlib.h>  
#define LEN sizeof(struct line)  
struct line  
{  
    int num;  
    struct line *next;  
};  
void main()  
{  
    int k;  
    struct line *p, *head;  
    head=NULL;
```

```
for(k=1;k<10;k++){
    if(k%2==0) continue;
    p=(struct line*)malloc (LEN);
    p->num=k;
    p->next=head;
    head=p;
}
while((p=p->next)!=NULL){
    printf("%d",p->num);
    p=p->next;
}
}
```

问题，上述程序的输出结果是什么？

2. 对程序

```
# include <stdio.h>

#define MAX 10

main(){
    int a[MAX],i,j,n;
    for(i=0;i<MAX;i++)
        scan("%d",&a[i]);
    n=i=MAX-1;
    while(i>=1){
        if(a[i]==a[i-1])
            if(i<n)&&(a[i]<a[i+1])||i==n){
                for(j=i+1;j<=n;j++) a[j-1]=a[j];
                n--;
            }
        i--;
    }
}
```

```
}

for(i=0;i<=n;i++)
    printf("%d",a[i]);

}
```

问题：当输入序列为 1122448833 时，输出是多少？



《浙大软件考研 878》微信公众号，扫描二维码关注，获取更多考研资讯
微信：tqh4546

浙江大学

2006 年攻读硕士学位研究生入学考试试题

注意：答案必须写在答题纸上，写在试题纸或草稿上均无效。

一、单项选择题。

1、下列 `for(i=0;(i<4)&&(j!=3);i++,j++)`；

- A.无限循环 B.循环 0 次 C.循环 3 次 D.循环 4 次

2、对于以下变量定义，表达式`*((int*)(p+1)+2)`的值是_____

`static struct{`

`int x,y[3];`

`}`

`A[3]={{1,2,3,4},{5,6,7,8},{9,10,11,12}},*p=a;`

- A.3 B.7 C.11 D.以上均不是

3、设 `char *s;` 以下表达式正确的是：

- A. `s="Hangzhou"` B. `strcpy(s,"Hangzhou");` C. `*s='H'` D. `s='H'`

4、下列程序的执行结果是

`Main(){`

`Int i;`

`char *s="a\b%d\n";`

`for(i=0;*s++;i++)`

`printf("%d",i);`

`}`

- A.5 B.6 C.7 D.以上均不是

二、程序理解(每题 5 分，共 20 分)

对程序

```
#include<stdio.h>
main(){
    int i,j;
    char *s="abc123d4ef";
    for(i=0,j=0;s[i]!='\n';i++)
        if(s[i]<'0' || s[i]>'9')    s[j++]=s[i];
    printf("%s",s);}
```

}

问题： 上述程序的输出结果是什么？

2. 对程序

```
#include<stdio.h>

void f(int n){

    int i,j;

    if(n%2||n==2) return;

    for(i=n-2;i>=n/2; i--){
        for(j=2;i%j;j++);
        if(j!=i) continue;
        for(j=2;(n-i)%j;j++);
        if(j!=n-i) continue;
        printf("%d:%d,%d\n",n,i,n-i);
    }
}

int test[]={6,8,10};

main()
{
    int i;
    for(i=0;i<3;i++) f(test[i]);
}
```

问题： 上述程序的输出结果是什么？

3. 对程序

```
#include <stdio.h>

main( ){
    int i,n;

    FILE *fp;

    if((fp=fopen("temp","w+"))==NULL)    exit(0);

    for(i=1;i<=10;i++)      fprintf(fp,"%3d",i);

    for(i=0;i<5;i++){

        fseek(fp,i*6L,SEEK_SET);
```

```
fscanf(fp,"%d",&n);  
printf("%3d",n);  
}  
fclose(fp);  
}
```

问题： 上述程序的输出结果是什么？



浙江 大学

二〇〇七年攻读硕士学位研究生入学考试试题

程序设计（共 40 分）

一、选择（每小题 2 分，共 8 分）

1. 下列程序的运行结果是什么？

```
main ()  
{  
    char s[]="Zhejiang";  
    int x=3,y=3;  
    printf("%d,%d", (++x, y++), sizeof(s));  
}
```

- A) 3 9 B) 4 9 C) 4 3 D) 3 8

2. 设 x、y、z 均为 int 类型，若 $x \& y \parallel z$ 的值为 1，则

- A) 若 z 为 0，则 x 和 y 必为 1 B) 若 z 为 0，则 x 和 y 均不为 0
C) x、y、z 必都为 1 D) 若 x 和 y 均为 1，则 z 必为 0

3. 下列程序的结果是什么？

```
main()  
{  
    int a[][4]={1,3,5,7,9,11,13,15,17,19,21,23};  
    int(*p)[4];  
    p=a;  
    printf("%d",*(p+2));  
}
```

4. 下列程序 prog.c 编译后执行命令行 “prog 123 456 ABC” ,其运行结果是什么？

```
Main( int argc,char*argv[])  
{  
    printf("%c",*(++argv[2]));  
}  
A)3            B)4            C)5            D)6
```

二、 程序填空。

下列函数 inverse 将一单向链表 h 就地逆转。请将空缺的部分填上。如，若 h 为 a->b->c->d，调用 h=inverse(h)后 h 为： d->c->b->a。 (6 分)

```
struct node {  
    int info;  
    struct node *next;  
};  
  
struct node *inverse(struct node *h)  
{  
    struct node *p1,*p2,*p3;  
    p1=NULL;  
    while(_2A_){  
        p3=p2->next;  
        p2->next=_2B_;  
        p1=p2;  
        p2=p3;  
    }  
    return p1;  
}
```

三、 程序理解（每题 5 分，共 15 分）

1.写出下列程序的运行结果。

```
int a[]={1,2,3,21,22,23,11,12,13};  
  
int f(int a[],int n)  
{  
    if(n==0)    return(a[0]);  
    else if(f(a,n-1)>a[n])    return f(a,n-1);  
    else return (a[n]);  
}
```

```
main()
{
    printf("%d",f(a,8));
}
```

2.写出下列程序的运行结果。

```
Void f(char s[],int start,char t[])
{
    int m,n,k,i,j;
    n=0;m=0;k=0;
    for(i=0;s[i]!='\0'; i++) m++;
    for(i=0;t[i]!='\0';i++){
        n++;
        if(t[i]>='0'&&t[i]<='9')k++;
    }
    for(i=m+n-k,j=m;i>start;i--,j--) s[i]=s[j];
    for(i=0;i<n;i++)
        if(t[i]<'0' || t[i]>'9') s[start++]=t[i];
}
main()
{
    char s[30]="abc123d4ef",t[]="1a2b3c";
    f(s,6,t);
    printf("%s\n",s);
}
```

3 写出下列程序的运行结果。

```
#include<stdio.h>
int f(int n,int m)
{
    int n1,m1,result=0;
    if(n&&m){

```

```
n1=n%10;  
  
m1=m%10;  
  
if(n1==m1) result=f(n/10,m/10)*10;  
  
else result = f(n/10,m/10)*10+((n1>m1)?n1:m1);  
  
return result;  
  
}else return n+m;  
  
}  
  
void main()  
{  
  
printf("%d\n",f(12345,3333));  
  
}
```

四、C 语言编程（11 分）

有一个整数序列代表一电梯在运行过程中所需要停的楼层顺序（楼层号为非负整数）。若该电梯上移一层需要 6 秒，下移一层需要 4 秒，在楼层中停需要 5 秒。假定电梯开始时在第 0 层，且电梯满足申请要求后不必回到第 0 层。请编写 C 程序，输入整数系列，输出电梯满足该系列的请求共需要多少时间（设用 -1 代表整数序列结束）。如输入 2 3 1 -1，将输出 41。

第 3 部分 数据结构

1. 将 5 个字母 ‘oops’ 按此顺序入栈，则有几种不同的出栈顺序可以仍然得到 ‘oops’。 (3 分)

A. 1 B. 3 C. 5 D. 6

2. 下面哪一个方法可以判断出一个有向图中是否有环（回路）？ (3 分)

A. 深度优先遍历 B. 拓扑排序 C. 求最短路径 D. 求关键路径

3. 在下列查找的方法中，平均查找长度与结点个数 n 无关的查找方法是哪种？ (3 分)

A. 顺序查找 B. 二分法 C. 利用二叉搜索树 D. 利用哈希(hash)表。

4. 一个三叉树有度(degree)为 1 的结点 5 个，度为 2 的结点为 3 个，度为 3 的结点有 2 个，则这棵三叉树的叶子结点的个数有多少。 (3 分)

A. 10 B. 12 C. 13 D. 8

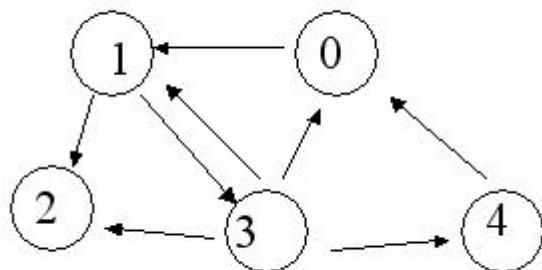
5. 给定一个整数序列 25、84、21、47、15、27、68、35、20，请给出用步长为 3 的希尔排序(shell sorting)算法进行第一轮排序后的结果。 (5 分)

6. 右图所示为一有向图，请给出该图的下述要求：

(1) 每个顶点的入/出度。 (2 分)

(2) 邻接矩阵。 (2 分)

(3) 逆邻接表。 (2 分)



(4) 强连通分量。 (2 分)

7. 下面是将任意序列调整为最大堆(max heap)的算法，将空白部分补充完整： (6 分)

将任意序列调整为最大堆通过不断调用 `adjust` 函数，即：

```
for(l=n/2;l>0;l--)    adjust(list ,l,n);
```

其中 `list` 为待调整序列所在数组（从下标 1 开始），`n` 为序列元素个数，`adjust` 函数为：

```
void adjust(int list[],int root ,int  n )  
{  
    int child ,rootkey;  
    rootkey=list[root];  
    child=2*root;  
    while (child <=n){  
        if((child<n)&&(list[child]<list[child+1]))  
            _____;  
        if(rootkey>list[child])  
            break;  
        else{  
            list[____]=list[child];  
            chuld*=2;  
        }  
    }  
    _____=rootkey;  
}
```

8. 用 C 语言完成一个在二叉搜索树中删除关键字为 `x` 的结点的程序（9 分）



浙江大学

二〇〇八年攻读硕士学位研究生入学考试试题

一、根据下列各题要求选择一个正确答案（每小题 2 分，共 8 分）：

1. 在 while (E) {...} 中, E 是判断字符 c 是否是大写字母, 则 E 可表达为 ①;

A. ‘A’ <=c<= ‘Z’ B. (c>= ‘A’)&(c<= ‘Z’)

C. ((c< ‘A’)||(c> ‘Z’))==0 D. (‘A’ <=c)AND(c<= ‘Z’)

2. 下列四个程序段, 正确的是 ②

A. char s[20],t[20]; t=” program” ; strcpy(s,t);

B. char s[20], *t=” program” ; s=t;

C. char *s,*t=” program” ; strcpy(s,t);

D. char s[20],t[20]=” program” ; strcpy(s,t);

3. 下列程序的运行结果是 ③

```
char str[]=" abc\0def\0ghi" ,*p=str;  
printf( "%s" , p+5);
```

字符原样输出。 (6分)

```
#include <stdio.h>
#include <ctype.h>
main(int argc,char*argv[])
{
    FILE *fp;
    char c;
    while(--argc>0)
        if((fp=fopen( ⑤
                    printf(” Can
                        exit(1);
                }else{
                    while( ⑥ )
                        if(isupper(c))
```

```
        putchar( ⑦ );  
    else putchar(c);  
    fclose(fp);  
}  
}
```

三、程序理解（每题 5 分，共 15 分）

1. 写出下列程序的运行结果⑧。

```
int f(int a[],int n)  
{    int s1,s2,i,j;  
    s2=2;  
    for(i=0;i<n;i++){  
        s1=0;  
        for(j=i;j<n;j++){  
            s1+=a[j];  
            if(s1>s2)    s2=s1;  
        }  
    }  
    return s2;  
}  
main()  
{    int a[]={4,-3,5,-2,-1,2,6,-2};  
    printf("%d",f(a,8));  
}
```

2. 写出下列程序的运行结果⑨。

```
int f(int x,int y)  
{    if(y==1)    return x;  
    else return f(x,y-1)+x;  
}  
main()  
{
```

```
    printf("%d",f(21,12));  
}
```

3. 写出下列程序运行结果 ⑩。

```
#include<stdio.h>  
  
#define LEN sizeof(struct line)  
  
struct line  
{    int mun;  
    struct line *next;  
};  
  
main()  
{    int k;  
    struct line *p,*head,*tail;  
    head=tail=NULL;  
    for(k=1;k<8;k++){  
        p=(struct line *)malloc(LEN);  
        p->num=k;  
        if(head==NULL){head=p; tail=p;}  
        else if (k%2){tail->next=p; tail=p;}  
        else{p->next=head; head=p;}  
    }  
    if(tail)tail->next=NULL;  
    for(p=head;p!=NULL;p=p->next)  
        printf(" %d ",p->num);  
}
```

四、C 语言编程 (11 分)

数组 `int a[]` 中存放若干段整数，每一个段由相同的整数组成。请编写函数 `int max(int a[], int n)` 求最长段的整数个数，其中 `n` 为数组大小（即所有整数的总数）。例如，`a[]={1, 1, 2, 2, 2, 3, 4, 4, 5, 5, 5, 5, 6, 8, 8, 8}`，`max(a, 16)` 将返回 4（最长段为“5, 5, 5, 5”）。

一、单选题（每题 1.5 分，共 6 分）

- (1) 给定一个单向链表，若要删除一个指针 p 所指的节点，下列哪一个操作是正确的。
- ① $p \rightarrow next = p \rightarrow next \rightarrow next$ ② $p = p \rightarrow next$
③ $p \rightarrow next \rightarrow next = p \rightarrow next$ ④ $p = p \rightarrow next \rightarrow next$
- (2) 从理论上讲，将数据以下哪种结构存放，则查找一个数据所用时间不依赖于数据个数 N。
- ① 二叉树查找 ② 链表 ③ 二叉树 ④ 哈希表
- (3) 有 n 个结点的无向图的边数最多为下列哪一种？
- ① $n+1$ ② $n(n-1)/2$ ③ $n(n+1)$ ④ $2n(n+1)$
- (4) 某二叉树的前序遍历序列为 IJKLMNO，中序遍历序列为 JLKINMO，则后序遍历序列为下列哪个选项。
- ① JLKMNOI ② LKNJOMI ③ LKJNOMI ④ LKNOJMI

二、简答题（共 20 分）

- (1) 二分查找算法的时间复杂度为（填空 A）（请统一写在答题纸上）（2 分）
- (2) 给定一个整数序列：{25、84、21、47、15、27、68、35、20}，请写出采用堆排序算法（Heapsort）初始建立的一个最小堆的整数序列（按照数组元素下标从小到大）。
(5 分)
- (3) 若文本中文字仅由 5 种字符 a, b, c, d, e 组成，它们出现的频率分别为 21 次、3 次、9 次、12 次、55 次，画出 Huffman 树，并列出 Huffman 编码。（5 分）
- (4) 在用于表示有向图的邻接矩阵中，对第 I 行的元素进行累加，可得到第 I 个定点的（填空 B）度，而对第 J 列的元素进行累加，可得到第 J 个顶点的（填空 C）度。（请统一写在答题纸上）（3 分）
- (5) 将整数序列{12、25、80、99、90、85、15} 按序插入一个初始为空的 AVL 树，画出插入完毕后的 AVL 树（至少要写其中的三个步骤）。（5 分）

三、已知二叉树中的节点类型用 BinTreeNode 表示，被定义为：

```
struct BinTreeNode{  
    char data;  
    BinTreeNode *leftChild, *rightChild;  
};
```

其中 data 为节点值域； leftChild 和 rightChild 分别为指向左、右孩子的指针域，根据下

面函数声明写出求一棵二叉树高度的算法，该高度由函数返回。参数 BT 初始指向这棵二叉树的根节点。（8 分）

```
int BtreeHeight (BinTreeNode *BT);
```

四、下面算法主要完成任务如下：给定一个单向链表，将其反序。请将该算法补充完整。（6 分）

```
/*Assuming no header and L is not empty.*/  
  
List ReverseList(List L)  
{  
    PositionCurrent,NextPos,PrewiousPos;  
    PreviousPos=NULL;  
    _____(填空 E)_____; (请统一写在答题纸上)  
    NextPos=L->Next;  
    while(NextPos!=NULL)  
    {  
        CurrentPos->Next=PreviousPos;  
        PreviousPos=CurrentPos;  
        _____(填空 F)_____; (请统一写在答题纸上)  
        NextPos=NextPos->Next;  
    }  
    _____(填空 G)_____; (请统一写在答题纸上)  
    return CurrentPos;  
}
```



《浙大软件考研 878》微信公众号，扫描二维码关注，获取更多考研资讯
微信：tqh4546

浙江大学

二〇一〇年攻读硕士研究生入学考试试题

考试科目 计算机专业基础 (A 卷) 编号 878

注意：答案必须写在答题纸上，写在答题卷上或草稿纸上无效。

第一部分：数据结构（共 60 分）

一、单项选择题（每题 2 分，24 分）

1、 h 为不带头结点的单向链表，在 h 的头上插入一个新结点 t 的语句是：

- A. $t \rightarrow \text{next} = h; h = t;$
- B. $h = t; t \rightarrow \text{next} = h;$
- C. $t \rightarrow \text{next} = h \rightarrow \text{next}; h = t;$
- D. $h = t; t \rightarrow \text{next} = h \rightarrow \text{next};$

2. 设一个栈的输入顺序是 1、2、3、4、5，若第一个输出的元素是 4，则最后一个输出的元素是：

- A. 1
- B. 5
- C. 1 或 5
- D. 以上都不是

3、某一义树的前序遍历序列和中序遍历序列好相反，则该义树一定是：

- A. 任一节点无左孩子
- B. 任一节点无右孩子
- C. 只有一个根节点
- D. 不存在

4、在一个用数组表示的完全叉树中，根节点下标为 1，那么下标为 17 和 19 节点的最近公共祖先的下标是：

- A. 1
- B. 2
- C. 4
- D. 8

5、如果森林 F 采用“儿子 - 兄弟”表示法对应的二叉树是 16 个节点的完全二叉树，问：森林 F 中树的数目和最大树的节点个数分别是多少？

- A. 2 和 8
- B. 2 和 9
- C. 4 和 8
- D. 4 和 9

6、下面关于 m 阶 B 树说法不正确的是：

- A. 最低一层节点数一定大于其它各层节点数的总和；
- B. 非叶结点均有不小于 $m/2$ 个儿子；
- C. 最大值一定在叶节点上；
- D. 当插入一个数据项引起 B 树节点分裂后，树并不一定会长高一层。

7、若 AVL 树的节点数是 21，则该树的高度至多是多少？（只有一个根节点的树高度为 0）

- A. 4
- B. 5
- C. 6
- D. 7

8、设一段文本中包含对象 $\{a, b, c, d, e\}$ ，其出现次数相应为 $\{3, 2, 5, 1, 1\}$ ，则经过哈夫曼编码后，该文本所占总位数为：

- A. 40
- B. 36
- C. 25
- D. 12

9、已知有向图 $G = \{V, E\}$ ，其中 $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ ， $E = \{\langle v_1, v_2 \rangle, \langle v_1, v_3 \rangle, \langle v_2, v_5 \rangle, \langle v_2, v_6 \rangle, \langle v_3, v_4 \rangle, \langle v_3, v_5 \rangle, \langle v_4, v_5 \rangle, \langle v_5, v_6 \rangle\}$ 。下面哪个序列是 G 的拓扑序列：

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

10、在求解有向图的关键路径问题时，如果该有向图用邻接矩阵表示且第 i 列值全为 ∞ ，则：

- A. 如果关键路径存在，第 i 个顶点一定是起点
- B. 如果关键路径不存在，第 i 个顶点一定是终点
- C. 关键路径不存在
- D. 该有向图对应的无向图存在多个连通分量

11、一个待排序序列 $(46, 84, 56, 40, 38, 79)$ 经过第 1 轮处理后得到的结果是 $(40, 38, 46, 56, 84, 79)$ 。则采用的排序算法是：

- A. 前单选择排序
- B. 简单插入排序
- C. 快速排序
- D. 堆排序

12、一个待排序序列为 $(46, 84, 56, 40, 38, 79, 13, 27)$ 按由小到大进行排序。下面哪个是初始步长 $d=4$ 的希尔排序法第一趟的结果

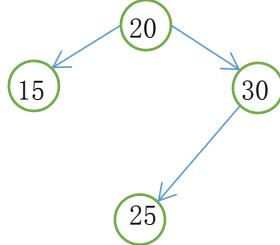
- A. $13, 27, 38, 40, 46, 56, 79, 84$

- B. 46, 27, 56, 40, 38, 79, 13, 84
- C. 46, 84, 40, 56, 38, 79, 13, 27
- D. 38, 79, 13, 27, 46, 84, 56, 40

二、分析题(共 36 分)

1、(8 分) 对于如图所示 AVL 树，请分别画出按顺序插入 27, 26, 14 后的结果。

注意：画出插入每个节点后的结果；后一个插入节点基于前一个插入的结果。



2、(8 分) 现有一棵节点值各不相同的完全叉树，有人给出了判别该完全二叉树是否是查找树(排序树)或最小堆的函数(如下)。请问：

- (1) 查找树判别函数 IsSearchTree 是否正确？若不正确，说明理由不用编写正确的函数；
- (2) 最小堆判别函数 IsMinHeap 是否正确？若不正确，说明理由(不用编写正确的函数)。

//树用数组 a 表示，根为 a[1]树总节点数为 n - 1。

//若要判别整个树是否是查找树或是最小堆，可分别调用 IsSearchTree(a, n, 1) 或

//IsMinHeap(int a[n], int n, int i);若是返回 1，否则返回 0。

```
int IsSearchTree(int a[ ], int n, int i)//该函数判别以 i 为根的子树是否是查找树
{
    if(i >= n) return 0;
    if(2*i+1 < n) return (IsSearchTree(a, n2*i) && IsSearchTree(a, n, 2*i+1)
        && (a[ i ] > a[2*i] && (a[ i ] < a[2*i+1])));
    else if(2*i+1 == m) return (a[ i ] > a[2*i]);
    else return 1;
}
```

```
int IsMinHeap(int a[ ], int n, int i)//该函数判别以 i 为根的子树是否是最小堆
{
    if(i >= n) return 0;
    if(2*i+1 < n) return (IsMinHeap(a, n2*i) && IsMinHeap(n, 2*i+1)
```

```
&& (a[ i ]<a[2*i])&&(a[ i ]<a[2*i+1]))  
else if(2*i+1== n) return (a[ i ]<a[2*i]);  
else return 1;  
}
```

3、(10 分)已知无向图 G 的邻接矩阵如图,请

- (1) 求 G 的最小生成树代价值,并画出相应的最小生成树;
- (2) 写出从顶点 0 出发的深度遍历结果;
- (3) 若用 Dijkstra 算法求顶点 0 到所有其他顶点的最短路径,则节点扩展的顺序是么?

	0	1	2	3	4	5	6
0	0	17	∞	∞	∞	10	8
1	17	0	16	∞	∞	∞	14
2	∞	16	0	12	∞	∞	∞
3	∞	∞	12	0	22	∞	18
4	∞	∞	∞	22	0	25	24
5	10	∞	∞	∞	25	0	∞
6	8	14	∞	18	24	∞	0

4、(10 分)对下列关键字序列(33, 25, 48, 59, 36, 72, 46, 7)构造表长为 13 的哈希表。假设哈希函数为 $h(key)=key \% 13$, 用二次探测法解决冲突, 探测序列为 $h_i(key)=(h(key) +/- i^2) \% 13$, 即 $h_0, h_0+1, h_0-1, h_0+4, h_0-4, h_0+9, h_0-9 \dots$ 等等。

- (1) 画出按序列顺序插入各关键字后的哈希表;
- (2) 针对插入上述序列后的哈希表, 求等概率情况下查找成功的平均查找长度 ASL;
- (3) 若要查找关键字 8, 请给出相应的探测序列并说明在哪个位置上发现查找不成功。



《浙大软件考研 878》微信公众号，扫描二维码关注，获取更多考研资讯
微信：tqh4546

浙江大学

二〇一一年攻读硕士研究生入学考试试题

考试科目 计算机专业基础 (A 卷) 编号 878

注意：答案必须写在答题纸上，写在答题卷上或草稿纸上无效。

第一部分：数据结构（共 60 分）

一、单项选择题(每题 2 分, 24 分)

1. 设某二维数组 $A[1 \dots n, 1 \dots n]$ ，则在该数组中用顺序查找法查找一个元素的时间复杂度为：

- A. $O(\log n)$
- B. $O(n)$
- C. $O(n \log n)$
- D. $O(n)$

2 若某堆栈的输入序列为 1, 2, 3..n-1, n, 输出列为递减序列，则为了满足这样的序列输出，该栈的容量至少必须是：

- A. $n/2$
- B. n
- C. n^2
- D. 无法判断

3、若采用带头、尾指针的单向链表表示个堆栈，那么该堆栈的栈顶指针 top 应该如何设置：

- A. 将表项设置为 top
- B. 将链表尾设置为 top
- C. 随便哪端作为 top 都可以
- D. 链表头、尾都不适合作为 top

4 已知 10 个数据元素为 (55, 18, 10, 34, 73, 62, 85, 60, 26, 43)，对该数列按从小到大排序，经过一趟冒泡排序后的序列为：

- A. 10, 18, 34, 55, 73, 62, 60, 26, 43, 85
- B. 18, 10, 34, 55, 62, 73, 60, 26, 43, 85
- C. 18, 10, 34, 55, 62, 60, 73, 26, 43, 85
- D. 10, 18, 34, 55, 62, 60, 73, 26, 43, 85

5、用某种排序方法对数列 (25, 84, 21, 47, 15, 27, 68, 35, 20) 进行排序时，元素序列变化情况如下，

- (1) 23, 15, 20, 47, 35, 21, 68, 84, 27
- (2) 20, 15, 25, 21, 27, 47, 35, 84, 68

(3) 15, 20, 21, 25, 27, 35, 47, 68, 84

则所采用的排序方法是：

- A. 简单选择排序
- B. 分组为 3、2、1 的希尔排序
- C. 冒泡排序
- D. 以第一个元素为支点的快速排序

6. 下列几种排序法中，要求空间最大的是

- A. 插入排序
- B. 选择排序
- C. 快速排序
- D. 归并排序

7. 对相同的 n 个整数构成的二叉排序树和最小堆，下面哪个说法是不正确的：

- A. 二叉排序树高度大于等于最小堆高度
- B. 对该叉排序树进行中序遍历可得到从小到大的序列
- C. 从最小堆根节点到其任何叶节点的路径上的节点值构成从小到大的序列
- D. 对该最小堆进行按层次 (Levelorder) 遍历可得到从小到大的序列

8. 在为 n 个对象设计最优编码时，如果相应的哈夫曼树有 67 个节点，则可知 n 的值为：

- A. 22
- B. 33
- C. 34
- D. 不确定

9. 某二义树的前序遍历序列和中序遍历序列一样，则该二义树一定是：

- A. 任一结点无左孩子
- B. 任结点无右孩子
- C. 只有一个根结点
- D. 不存在

10. 所谓“好的”散列 (hash) 函数应该有一个重要的性质，即两数值应当尽量以什么方式取其值域的每个值？

- A. 最大概率
- B. 最小概率
- C. 平均概率
- D. 同等概率

11. 给定一个无向有权图 G，下列哪些说法是正确的？

- A. 设 T 为 G 的最小生成树，那 T 中任何两个顶点之间的路径就是图 G 中这两个顶点的最

短路径；

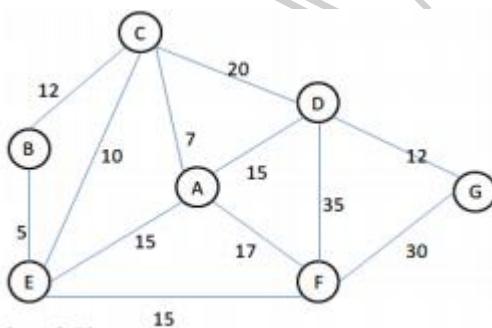
- B. 设 p 是 v 到 u 的最短路径，如果将图 G 中的每条边长度均加 1 后，那么 P 仍然是从 v 到 u 的最短路径；
C. 如果该图有 n 个顶点且正好有 $n-1$ 条边，那么该图一定没有回路；
D. 以上说法都不对。

12、为了测试一有向图中是否有回路，可用的方法是：

- A. Dijkstra 算法
B. Kruskal 算法
C. 拓扑排序方法
D. 关键路径算法

二、分析题(共 36 分)

1、(10 分) 已知无向图如下。试给出以 A 为顶点访问序列起点(同顶点的多个邻接点，按字母顺序访问)的：



(1) 深度优先(depth-first)遍历序列：

- (2) 宽度优先(breadth-first)遍历序列；
(3) 最小代价生成树及最小代价。

2、(10 分) 已知线性表 {42, 26, 8, 70, 6, 56, 2, 80, 35}

- (1) 按表中元素顺序依次插入一棵初始为空的二叉排序树，请画出插入完成后的排序树；
(2) 按表中元素顺序构造一颗平衡二叉树(AVL 树)。

3. (8 分) 下列函数 sort 是对数组 $\{r[0, 1, \dots, r[n-1]\}$ 进行从小到大排序。与一般的简单选择排序方法不同，该算法是在每一轮循环中同时查找当前未排序序列中的最大和最小值，将它们放到正确的位置，请将缺少的语句填上，其中，函数 $swap(r, i, j)$ 为交换 $r[i]$ 、

r[j]两个数组分量的值。

```
voidsort(intr[ ], int n )  
{  
    int i, min, max ;  
    for(i=1;i<n-i+1;i++) {  
        min= max= i;  
        for(j=i+1;_____1_____;++j)  
        {  
            if(_____2_____) min=j;  
            else if(r[j]>r[max]) max=j;  
        }  
        if(_____3_____)  
            swap(r, min, i);  
        if(max!=n-i+1){  
            if(_____4_____) swap(r, min, n-i+1);  
            else swap(r, max, n- i+1);  
        }  
    }  
}
```

4、如果分别用 S 和 X 代表栈的 push 和 pop 操作。那么 S 和 X 组成的字符序列将可以表示堆栈的操作的序列。如果该字符序列对应一个可行的堆栈操作、我们称该序列是合法序列(因为在 push 一个元素之后，不可能连续 pop 出两个元素)。请编写一个函数 legal (用于判断一个由“S”和‘X’组成的字符序列是否合法。Int legal(char st1[])//函数返回 1\(合法)



《浙大软件考研 878》微信公众号，扫描二维码关注，获取更多考研资讯
微信：tqh4546

浙江大学

二〇一二年攻读硕士研究生入学考试试题

考试科目 计算机专业基础 (A 卷) 编号 878

注意：答案必须写在答题纸上，写在答题卷上或草稿纸上无效。

第一部分：数据结构（共 60 分）

一、单项选择题(每题 2 分, 共 24 分)

1、下列函数的时间复杂性是：

```
int f(int n){  
    if(n<=0) return 1;  
    else return n+f(n-1);  
  
    A. O(log2n)          B. O(n)  
    C. O(nlog2n)        D. O(n2)
```

2、在单链表中，指针 p 指着结点 A，若要删除 A 之后的结点(假设存在)，则语句可以为：

- A. p=p->next;
- B. p->next=p->next->next;
- C. p=p->next->next, ;
- D. p->next=p;

3、在一个链式队列中，f 和 r 分别指向队列的头和尾。如果将 s 指向的结点插入队列中，正确的操作是：

- A. f->next=s; f=s
- B. r->next=s; r=s
- C. s->next=r; r=s;
- D. s->next=f; f=s;

4、如果某非空二叉树的前序序列和后序序列正好相反，则该又树一定具有的特点是：

- A. 任一结点无左孩子
- B. 任一结点无右孩子
- C. 空或只有一个结点
- D. 不存在度为 2 的结点

5、从权值分别为 9, 2, 5, 7 的四个叶子结点构造一棵哈夫曼 (Huffman) 树，则该树的带权路径长度 WPL 为：

6、若想查找 63，哪个最有可能是在二叉排序树上进行的查找

- A. 2 25 101 39 80 70 59 63
B. 39 101 25 80 70 59 63
C. 101 70 2 39 25 59 80 63
D. 59 2 80 70 39 25 63

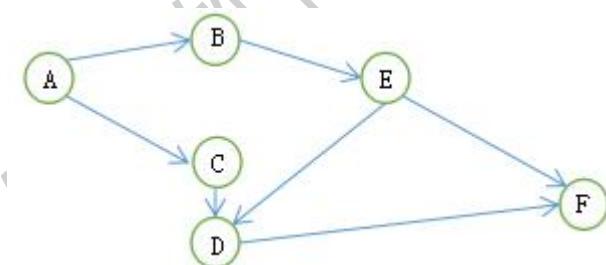
7、对于一个共有 n 个结点， K 条边的森林，共有几棵树？

- A. $n-k$
 - B. $n-k+1$
 - C. $n-k-1$
 - D. 不能确定

8、对于一个 n 个顶点的有向无环图，如果它的拓扑排序是唯一的，那么下列哪句话是不对的

- A. 该图的最长路径是 $n - 1$
 - B. 该图不是一个双连通图
 - C. 至少存在一个顶点它的出度大于 1
 - D. 当从入度为 0 的顶点开始分别进行深度和宽度遍历时，遍历结果是一样的

9、下图为AOV网，其可能的拓扑有序序列为



- A. ACBDEF B. ACBEDF
C. ABCEFD D. ABCDFE

10、在下列排序算法中，时间复杂度不受数据初始特性影响、恒为 $O(n^2)$ 的是

- A. 插入排序
 - B. 冒泡排序

C. 希尔排序

D. 堆排序

11、已知 10 个数据元素为(54, 28, 16, 34, 73, 62, 95, 60, 26, 43)，对该数列按从小到大排序，经过一趟冒泡排序后的序列为：

A. 16, 28, 34, 54, 73, 62, 60, 26, 43, 95

B. 28, 16, 34, 54, 62, 60, 73, 26, 43, 95

C. 28, 16, 34, 54, 62, 73, 60, 26, 43, 95

D. 16, 28, 34, 54, 62, 60, 73, 26, 43, 95

12、如果 n 个对象具有相同的散列(hash)值，并且采用线性探测冲突解决策略，那么要查找这个对象，所需要的最少探测次数是：

A. $n-1$

B. n

C. $n+1$

D. $n(n+1)/2$

二、分析题(共 36 分)

1、(7 分) 请按表 {42, 26, 8, 70, 102, 6, 56, 2, 88, 80, 35} 中元素顺序构造一棵二叉平衡树(AVL 树)。

2、(9 分) 下列函数从最大堆中删除最大值元素，请将缺少的语句填上。

```
ElementType DeleteMax( PriorityQueue H )  
/*H 为包含数组 Elements、当前堆大小 Size 和堆最大容量 Cap 的结构指针*/  
{  
    int i, Child;  
    ElementType MaxElement, LastElement;  
    MaxElement=_____1_____;  
    LastElement = H->Elements[H->size--];  
    for( i= 1; i *2 <= H->Size; i= Child)  
    {  
        Child = i *2;  
        if(_____2_____)  
            Child++;  
        if(LastElement < H -> Elements[Child])  
            _____3_____;
```

```
        else          break;  
    }  
  
    H->Elements[ i ] = LastElement;  
  
    return MaxElement;  
}
```

3、(10 分) 给出有向图的邻接矩阵如下：

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
V1	0	1	1	1	0	0	0	0	0	0
V2	0	0	0	1	1	0	0	0	0	0
V3	0	0	0	1	0	1	0	0	0	0
V4	0	0	0	0	0	1	1	0	1	0
V5	0	0	0	0	0	0	1	0	0	0
V6	0	0	0	0	0	0	0	1	1	0
V7	0	0	0	0	0	0	0	0	1	0
V8	0	0	0	0	0	0	0	0	0	1
V9	0	0	0	0	0	0	0	0	0	1
V10	0	0	0	0	0	0	0	0	0	0

- (a) 请分别写出 V1 开始的深度遍历和宽度遍历的结果。根据图邻接矩阵，结点访问顺序按列从左向右，因而该深度遍历和宽度遍历结果都是唯一的。
- (b) 请给出拓扑排序序列，要求在寻找拓扑排序的各结点时，如果有多种选择那么优先选择结点序号小的结点。根据上述要求，这样的拓扑序列是唯一的。

4、(10 分)

- (1) 如果二叉树的前序遍历结果为 ABDEHCFIJGK, 中序遍历结果为 DBHEAIFJCKG, 问在这棵二叉树中，距离结点 I 和 G 最近的共同祖先是哪个结点？

《浙大软件考研 878》微信公众号，扫描二维码关注，获取更多考研资讯

微信：tqh4546

- (2) 如果已知二叉树的前序遍历结果和中序遍历结果分别用字符串 s1 和 s2 表示，其中的字符代表树的结点。请设计算法求距离指定的两个结点(字符)最近的共同祖先。

《浙大软件考研878》微信公众号



浙江大学

二〇一三年攻读硕士研究生入学考试试题

考试科目 计算机专业基础 (A 卷) 编号 878

注意：答案必须写在答题纸上，写在答题卷上或草稿纸上无效。

第一部分：数据结构（共 60 分）

一、单项选择题（每题两分，共 24 分）

- 1、将中缀表达式 $A - (B - C / D) * E$ 转化为后缀表达式，该堆栈的大小至少为：()
- A. 2 B. 3 C. 4 D. 5
- 2、若有一棵二叉树的总结点数为 98，只有一个儿子结点数为 48，该树的叶子结点数是：()
- A. 25 B. 50 C. 不确定 D. 这样的树不存在
- 3、已知一二叉树的前序遍历的结果是 ABC，则以下哪个序列是不可能的中序遍历结果：()
- A. ABC B. BAC C. CAB D. 以上三种均有可能
- 4、设高度为 h (只有根节点时 h 为 1) 的二叉树只有度为 0 和 2 的结点，则该二叉树的结点至少为：()
- A. $2h - 1$ B. 2^{h-1} C. $2^{h-1} + 1$ D. $2^h - 1$
- 5、有 6 个字符出现的频率分别是 2、1、4、5、7、3，如果采用哈夫曼编码方式进行最优编码，那么字符编码长度最长的位数是：()
- A. 3 B. 4 C. 5 D. 6
- 6、设一个二叉树的前序遍历和中序遍历分别为 ABCDFE 和 CBAFDE，那么在树对应的二叉中序线索树中，下列哪些结点的右链都是线索：()
- A. C 和 B B. B 和 A C. C 和 D D. D 和 F
- 7、对于一个有 6 个结点、十条边的无向图，包含两个不连通的子图，那么这两个子图的结点分别是：()
- A. 1 和 5 B. 2 和 4 C. 3 和 3 D. 不确定
- 8、哈希表的地址区间为 0~10，哈希函数为 $H(K) = K \% 11$ 。采用线性探测法处理冲突 $h_i(k) = H(K+i) \% 11$ ，并已经将关键字序列 6、25、39、61 依次存储到初始为空的哈希表中。接下来存放哪个元素不会发生冲突：()
- A. 3 B. 6 C. 8 D. 10
- 9、将 1、2、3、4、5、6 依次存入到初始为空的 AVL 树中，当完成这六个元素插入后，该

AVL 树共有多少层（结点层数）：（ ）

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

10、直接插入排序在最好的情况下的时间复杂度为：（ ）

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

11、对某个初始序列为进行从小到达排序，经过若干趟处理后的状态为（8、9、10、4、5、6、20、1、2）。可以知道的排序方式可能为：（ ）

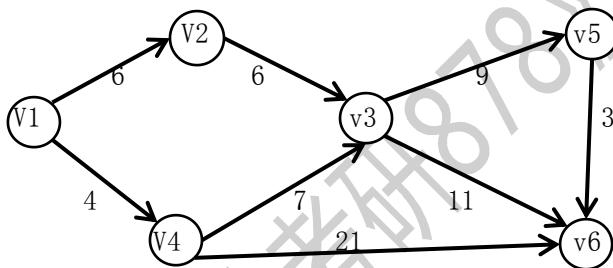
- A. 堆排序 B. 插入排序 C. 选择排序 D. 冒泡排序

12、下列哪个序列是堆（ ）

- A. 49, 76, 65, 13, 27, 50, 97, 38 B. 50, 65, 76, 97, 13, 27, 38, 49
C. 97, 76, 65, 50, 49, 13, 27, 38, D. 97, 49, 27, 50, 76, 38, 65, 13

二、分析题（共 36 分）

1、（10 分）已知一个工程活动的有向图如下，边的权值代表完成活动所需要时间，V1 为活动起点。



(1) 给出 V1 结点开始的深度优先 (depth-first) 遍历序列。如果某个结点有多个邻接点，这结点数字递增的顺序访问。

- (2) 给出 V1 开始的拓扑排序序列。如果某个结点有多个邻接点选择，结点编号小的优先；
(3) 给出 V6 的最早完成时间。

2、（10 分）已知线性表 {26、8、70、6、56、2、80、30}

(1) 按表中元素顺序依次插入一个初始为空的哈希表。哈希表的地址区间为 0-12，哈希函数为 $H(K) = K \% 13$ 。采用二次探测法处理冲突 $h_i(k) = (H(k) + i^2) \% 13$ ，请画出插入完成后的哈希表状态；

- (2) 针对该序列，计算平均（成功）查找次数；
(3) 请比较分析用哈希方法和 AVL 树方法解决动态查找问题的特点。

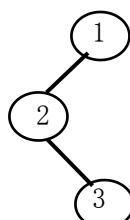
3、（8 分）下列函数从 N 个整数的数组 A 中找出其中的第 k (< N) 大整数，其中函数 BuildMinHeap(H, R) 是将整数 H[1]……H[6]建成最小堆。

```
int FindKthLargest ( int A[ ], int N, int K ) {
    int *H, i, next child;
    H= ( int )malloc ( ++N, sizeof ( int ) );
    for ( i=1; i<=N; i++ ) H[i]= A[i-1];
    RmitdMinHeap ( H, K );
    for ( next=K+1; next < =N; next++ ) {
        H[0]= H[next];
        if ( H[0]> H[1] ) {
            for ( i=1; i*2 <=K; i=child ) {
                child= i*2;
                if ( child!=K && _____ 1 _____ ) child--;
                if ( _____ 2 _____ )
                    H[i]=H[child];
                Else      break;
                _____ 3 _____ ;
            }
        }
        Return _____ 4 _____ ;
    }
}
```

3、(8 分) 对二叉查找树(排序树) 的中序遍历过程实际上是将树的结点进行一系列堆校操作的过程

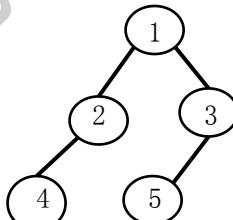
程。如果分别用 S(e) 和 X(e) 代表 e 入栈和栈顶 e 出栈操作，如下左图所示的二义树的中降遍历过程

可以用堆栈操作序列表示： S(1)、S(2)、X2)、S(3)、X3)、X(I)。



左图

右图



- (1) 请写出右图中二叉树的中序遍历堆栈操作序列；
- (2) 请说明求解以下问题的思路：根据二叉树的中序遍历堆栈操作序列，求任何一个结 i 的左子树结点总数。例如，左图中，结点 1 和结点 2 的左子树结点总数分别为 2 和 0。

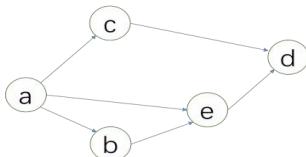


《浙大软件考研 878》微信公众号，扫描二维码关注，获取更多考研资讯
微信：tqh4546

17 年浙大软院 878 试题

一、选择题（20 题，每题 2 分，共 40 分）

1. 判断 $O(N^3)$ 、 $O(N \log N)$ 、 $O(N(\log N)^3)$ 、 $O(N^2 \log N)$ 的增速哪个最快。
2. 若有一颗完美二叉排序树，问选项中错误的性质描述。
3. 给了一个 AVL 树的建树序列，判断选项中错误的是
4. 11 年真题的 11 题，关于无向有权图的性质



5. ,此图的拓扑序列有几个？
6. 若有两个叶子结点是 A 和 B，则问前中后序序列里，A 和 B 的相对位置。
7. 堆排序、快速排序、归并排序的需要的辅助空间大小排序？
8. 给出原始序列，第一趟序列，第二趟序列。问是什么排序
9. 一个有 N 个结点的无向图，所有结点的度数为节点数的多少倍？
10. 一个三叉树，N1=5, N2=3, N3=2。问 N0=?
11. 给了一个最小堆，问删了两个元素之后，堆的层序遍历。

二、大题（50 分）

- 1、给出一个有权无向图，问深度遍历序列、广度遍历序列、最小生成树及其权值 (18 分)
- 2、哈希动态排序题，给出一个序列，将其放入容量为 13 的哈希表中，
 $H(key) = key \% 13$ ，用线性探测的方式处理冲突。
(1). 填表，将所给序列填入表中。
(2). 在第一次发生冲突的时候，表的装填因子是什么。
3. 程序填空题，要求利用堆排序将一个数列 $A[N]$ 排列成非递减序列。(12 分，共空)
(已经在历年真题中出现不少于 3 次了)
4. 编程题，编一个程序 `void (tree T, int X)`，作用是在二叉排序树 T 中找到比 X 大的数，并以非递增的方式打印出来。 (8 分)

C 语言（共 60 分）

一、选择题（15 道，每题 2 分，共 30 分）

二、分析题（5 道，每题 6 分，共 30 分）

此部分的前 4 题是读程序，写结果。第 5 题是程序填空，3 空，每空 2 分。

- 1、逆置链表
- 2、排序
- 3、数字符串中出现最多的元素
- 4、字符串，大小写转换
- 5、排序的一道程序填空



浙江大学

二〇一八年攻读硕士研究生入学考试试题

考试科目 计算机专业基础 编号 878

注意：答案必须写在答题纸上，写在答题卷上或草稿纸上无效。

第一部分：程序设计基础 C (共 60 分)

一、单项选择题（每题两分，共 30 分）

1、下列表达式不能表示 x 在区间 $[0, 5]$ 的是（）

- A. $x >= 0 \&\& x <= 5$ B. $x <= 5 \&\& x >= 0$
C. $!(x < 0 \mid x > 5)$ D. $0 <= x <= 5$

2、下列表达式与循环语句中 $x \% 3$ 等价的是（）

- While ($x \% 3$) i++;
A. $x \% 3 != 0$ B. $x \% 3 == 0$
C. $x \% 3 != 1$ D. $x \% 3 != 2$

4、已知二维数组 int a[2][3]={1, 2, 3, 4, 5}，则 a[1][1] 为：（）

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

5、有程序段

```
Int n=10, m=4;  
d1=(double) (m/n);  
d2=(double)m/n;
```

则执行程序后 d1 与 d2 的关系为（）

- A. $d1 > d2$ B. $d1 < d2$ C. $d1 = d2$ D. 无法判断

6、有程序段

```
int i, j, s=1;  
for(i=1, j=4; ++i != ++j; )  
    s=s*2;
```

则执行程序后 s 为（）

- A. 2 B. 4 C. 8 D. 不确定

7、有程序段

```
int s=15;  
for(i=s-1; s%i; i--)  
    Return i;
```



《浙大软件考研 878》微信公众号，扫描二维码关注，获取更多考研资讯
微信：tqh4546

则执行程序后 i 为（）

- A. 15 B. 5 C. 3 D. 10

8、下列表达式无意义的是（）

- A. “Flower” - “Flay” B. “Flower” +1
C. ‘1’ +2 D. ‘z’ -3

8、定义 A[3][2]，则下列哪个正确（）

- A. A[-1][1] B. A[2][2] C. A[3][1] D. A[2][-1]

9、一个文件复制题目，具体想不起来了

10、Char *p={ “Happy” , “Live” } , Char *s;
s=p;

则与 s+3 等价的为（）

具体选项记不清啦。

11、已知一个函数 f (3) ，求 f (f (3)) 的值。 (简单递归)

其余的几个没印象了。

二、综合应用题（共 30 分）

1、写出下面程序执行后的结果：

```
#include<stdio.h>
void sort(int a[], int m);
main() {
    int a[]={5, 3, 9, 1, 14, 2, 12, 6, 25, 15, 17, 35, 14, 22, 16, 4}, n;
    n=sizeof(a)/sizeof(int);
    sort(a[], n)
    for(int i=0;i<n;i++)
        printf("%d", a[i])
}
void sort(int a[], int m) {
    int i, j, t;
    for(i=1;i<n;i++)
        for(j=0;j<n-i;j++)
            if(a[j]>a[j+1]) {
                t=a[j];
                a[j]=a[j+1];
                a[j+1]=t;
            }
}
```



```
a[j+1]=t;  
}  
}
```

2、写出下列程序执行后的结果：

3、写出下列程序执行后的结果：

4、写出下列程序执行后的结果：

5、程序填空题。下列程序目的是为了删除链表中的逆序结点，如链表 4、2、5、3、6、7 经过执行后变为 4、5、6、7。

第二部分：数据结构（共 90 分）

一、单项选择题（每题两分，共 40 分）

1、已知每维均为 N 三维数组，在不改变元素顺序的情况下，则要找其最大元素的时间复杂度为（）

- A. N B. N^2 C. N^3 D. $N \log N$

2、下列关于线性表、栈、队列的主要区别描述正确的是（）

- A. 线性表和队列为线性结构，栈是非线性结构
B. 栈和队列是输入输出受限的线性表
C. 队列和链表可以有循环结构，而栈没有循环结构
D. 链表是线性结构，栈和队列是非线性结构

3、元素 1、2、3、4、5、6 按顺序进栈，如果输出的第一个元素为 5，则最后一个输出的元素为（）

- A. 1 B. 2 C. 6 D. 1 或 6

4、在二叉树的前、中、后遍历中，一个结点的左右孩子结点的顺序（）

- A. 前序和后序不同 B. 前序和中序不同
C. 都相同 D. 与树的形状有关

5、已知一个四叉树中，有 3 个度为 4 的结点、2 个度为 3 的结点、3 个度为 2 的结点，则该四叉树的叶子结点数为（）

- A. 15 B. 16 C. 17 D. 18

6、下面关于排序树的说法错误的是（）

- A. 最小结点一定是叶子结点
B. 将树进行中序遍历可以得到有序序列
C. 在二叉排序树查找一个元素，时间复杂度总是 $\log n$ (n 为结点数)

- D. 最小关键值结点右指针为空
- 7、78、70、61、96、120、90 构成的平衡二叉树的前序遍历序列为（）
- A. 70, 61, 96, 78, 120, 90
- B. 78, 70, 61, 96, 90, 120
- C. 78, 70, 61, 90, 96, 120
- D. 70, 61, 96, 78, 90, 120
- 8、已知堆 {5, 8, 12, 19, 28, 20, 15, 22} 插入 3 后的堆为（）
- A. 3, 5, 12, 8, 28, 20, 15, 22, 19
- B. 3, 5, 12, 19, 20, 15, 22, 8, 28
- C. 3, 8, 12, 5, 20, 15, 22, 28, 19
- D. 3, 12, 5, 8, 28, 20, 15, 22, 19
- 9、下列关于堆与二叉树的描述错误的是（）
- A. 堆是一棵完全二叉树
- B. 堆进行中序遍历不一定能得到有序序列
- C. 对二叉排序树进行中序遍历可以得到有序序列
- D. 在一棵二叉排序树中查找一个元素的时间复杂度为 $\log n$
- 10、用哈夫曼编码对 a、b、c、d 四个字母进行编码，已知 a、b 出现的频率最小，则哈夫曼编码可能为（）
- A. a:001 B. b:101 C. c:10 D. d:00
- 11、已知图用邻接矩阵存储，则第 i 个结点的出度为（）
- A. 第 i 行非零元素个数 B. 第 i 列非零元素个数
- C. 第 i 行零元素个数 D. 第 i 列零元素个数
- 12、用深度优先搜索遍历一个无环有向图，并在深度优先搜索算法退栈时打印相应的顶点，则输出定点的序列是（）
- A、逆拓扑有序 B、拓扑有序 C 无序的 D 以上都不对
- 13、已知邻接矩阵，求最小生成树的权值
- 14、用 Dijkstra 求最短路径，目标顶点依次为（以上两题王道有相似习题）
- 15、在基于排序码比较的排序算法中，哪种排序算法在最坏的情况下时间复杂度不高于 $O(N \log N)$ （）
- A 冒泡排序 B 希尔排序 C 归并排序 D 快速排序
- 16、已知有 n 个顶点的无向连通图，则其边数最少为（）
- A、n B、n+1 C、n-1 D、 n^2
- 17、已知序列 {3, 2, 5, 4, 7, 9, 6} 为某排序算法两趟排序后的结果，则所用的排序算

法为（）

- A 冒泡排序 B 插入排序 C 堆排序 D 快速排序

18、如果 n 个对象具有相同的散列(hash)值，并且采用线性探测冲突解决策略，那么要查找这个对象，所需要的最少探测次数是：

- A. $n-1$ B. n
C. $n+1$ D. $n(n+1)/2$

19、给出图的广度遍历序列，和部分邻接表，填入邻接表的空缺（虽然新颖，但是很简单）

20、实在想不起来了

二、综合应用题（50 分）

1、已知二叉树的后序遍历序列为 6, 5, 8, 2, 9, 1, 7, 3, 4，中序遍历序列为 6, 5, 2, 8, 9, 1, 4, 3, 7

- (1) 画出此二叉树
(2) 写出此二叉树的前序遍历序列

2、哈希表插入问题，哈希函数为 $H(Key)=H \% 11$ （具体数据忘记了，参见 13 年）

- (1) 画出插入后的哈希表；
(2) 当第一次发生冲突时的装填因子

3、程序填空

```
bool Floyd(MGraph Graph, WeightType D[][MaxVertexNum], Vertex
path[][MaxVertex-Num])
{
    vertex i, j, k;
    /*初始化*/
    for(i=0;i<Graph->Nv;i++)
        for(j=0;j<Graph->Nv;j++){
            D[i][j]=Graph->G[i][j];
            path[i][j]=-1;
        }
    for(k=0;k<Graph->Nv;k++)
        for(i=0;i<Graph->Nv;i++)
            for(j=0;j<Graph->Nv;j++)
                if(______){
```



《浙大软件考研 878》微信公众号，扫描二维码关注，获取更多考研资讯
微信：tqh4546

```
D[i][j]=_____;  
if(i==j&&D[1][j]<0) /* 若发现负值圈 */  
    return false; /* 不能正确解决，返回错误标记。 */  
path[i][j]=_____;  
}  
return true;  
/*算法执行完毕，返回正确标记*/  
}
```

D[i][k] + D[k][j] < D[i][j]

D[i][k] + D[k][j]

K

4、程序设计题

设计一个程序，在二叉树中寻找两个结点的公共结点。



2001 年计算机科学基础答案

一、

- 1、 A
- 2、 B
- 3、 3ello
- 4、 2
- 5、 printf((i+1)%5==0?"%d\n": "%d",a[i]);
- 6、 i<10&&a[i]>=0

二、

1、 $x - x^2 + x^3 - \dots$

2、

1) 、 TSC

2) 、 将 head 为头指针的单链表中的结点数据域不是大写字母 ('A'~'Z') 的结点删除，并释放结点的空间。

三、 程序如下：

1)、

```
#include <stdio.h>
Int atoi(char *s)
/*函数返回可能包含符号的数字序列 s 相应的整数*/
{
    Char c;
    Int result,sign;
    Result=0,sign=1;
    While(c=*s++)
        Switch(c){
            Case '-': sign=-1;break;
            Default: result=result*10+(c-48);break;
        }
    Return sign==? (0-sign):result;
}
```

```
}

2),
Int atoi(char*);  
Main()  
{  
FILE *ifp,*ofp;  
Char *str:  
  
Ifp=fopen("test.txt","r");  
Ofp=fopen("test.dat","rb");  
  
While(fscanf(ifp,"%s",str)!=EOF) /*从输入文件中依次读入数字序列 str*/  
    Fprintf(ofp,"%d ",atoi(str));  
/*str 转换为相应的整数后，写入输出文件中*/  
  
Fclose(ifp);  
Fclose(ofp);  
}
```

注：该程序为参考答案，并非标准答案。



2002 年计算机科学基础答案

一、

1~5、 ACCCD

6、 n

7、 5, 7, 7

8、 c<'a'||c>'z'?1:0;

二、

S[i]-'a'+10 f(s)

三、

1) 、输出为:

1+1+1

1+1+2

2), 输出为:

2+2+2

2+2+4

2+3+3

四、程序清单如下：

1) 、

Double myexp(double x)

/*该函数利用 e^x 的级数展开式求其近似值，精确到 10^{-4} */

{

 Double result,temp;

 Double a;

 Int b,i; /*级数中的每项由 temp=a/b 来表示*/

 Result=1.0,a=x,b=1,i=1;

 Temp=a/b;

 While(temp>=1.0E-4){

 Result+=temp;

 A*=x;

 B*=(++i);

```
Temp=a/b;  
}
```

```
Return result;  
}
```

2)、

Main()

/*该程序将 FIBONACCI 数列的前 n 项 F_i 及 e^{F_i} , 写入文件 out.txt 中*/

```
{
```

```
Int *a,n,l;
```

```
FILE *fp;
```

```
Scarf("%d",&n);
```

```
A=(int*)malloc(sizeof(int)*n);
```

```
A[0]=a[1]=1;
```

```
For(i=2;i<n;i++){
```

```
    A[i]=a[i-1]+a[i-2];
```

```
}/*构造 FIBONACCI 数列的前 n 项*/
```

```
Fp=fopen("out.txt","w");
```

```
For(i=0;i<n;i++){
```

```
    Fprintf(fp,"%d,%f\n",a[i],myexp(a[i]));
```

```
}/*分别将偶对 ( $F_i, e^{F_i}$ ) 写入输出文件中*/
```

```
Fclose(fp);
```

```
}
```



2003 年计算机科学基础答案

试题七

1~3、CDB

4、ava2

5、 $(-1)^{*}x^{*}x/i$

试题八：

1、1)、7654321 2)、交换 a,b

2、1)、AHelloWorld 2)、去掉串 s 中的空白符和转义字符'\t'和'\n'，并向左对齐。

试题九：

程序清单如下：

1)、

```
#include <iostd.h>
Int search(int *v,int n, int v)
/*在 v[n]中二分搜索 v，成功返回相应下标，不成功返回-1*/
{
    Int low,high,mid;
    Low=0,high=n-1;
    While(low<=high){
        Mid=(low+high)/2;
        If(v[mid]<v) low=mid+1;
        Else if(v[mid]>v) high=mid-1;
        Else return mid; /*搜索成功返回相应下表标*/
    }
    Return -1; /*搜索不成功返回-1*/
}
```

2)、

```
Int search(int *,int ,int);

Main()
/*该程序从 input.txt 文件中分别读入 2 个有 100 个整数的序列，并利用二分搜索
2 个序列中共同出现的整数输出到文件 output.txt 中*/
{
    Int *a,*b; /*分别用来存放 100 个整数序列*/将
    FILE *ifp,*ofp;
    Int i;

    A=(int*)malloc(100*sizeof(int));
    B=(int*)malloc(100*sizeof(int));
    Ifp=fopen("input.txt","r");
    Ofp=fopen("output.txt","w");

    Fread(a,sizeof(int),100,ifp);
    Fread(b,sizeof(int),100,ifp);

    For(i=0;i<100;i++){
        If(search(b,100,a[i])!=-1)
            Fprintf(ofp,"%d";a[i]); /*共同出现的整数则写入输出文件中*/
    }
    Fclose(ifp);
    Fclose(ofp);

}
```



2004 年计算机科学基础答案

程序设计部分

一

1. B 0x是十六进制的意思
2. C 0为f, 其余为t
3. B 不是很确定。*(a+i)[j]==((a+i)[j])??
4. D c中给字符串赋值要用strcpy () ; c#, java中是这样写的; c++中忘了
5. A 同scanf("...",&a), x[i]不是地址
6. if(i==n/10) convert(a+1, i);
 *a = i%10+'0'

二

1. 789 注意c是static
2. 不知道 我这里没有c的编译器。感觉题有问题, i没有初始化, 我记得c中没有初始化的变量是随机值, 而java中i会是0。如果i初始等于0, 结果是第i个单词的第i个字母经处理后输出, 即DCLJE

三

```
float calc(int n)
{
    float a = 1, b = 2, s = 0;
    int i = 0;
    for(i=0; i<n; i++)
    {
        s += a/b;
        b = a+b;
        a = b-a;
    }
    return s;
}
```



《浙大软件考研 878》微信公众号，扫描二维码关注获取更多资讯！

2005 年计算机科学基础答案

一、 1、 A 2、 A 3、 A 4、 5, 10

5、 ?

二、

1、 97531 2、 1248833



2006 年计算机科学基础答案

一、CBBB

二、1、abcdef

2、

6:3, 3

8:5, 3

10:7, 3

10:5, 5

3、1 3 5 7 9



《浙大软件考研 878》微信公众号，扫描二维码关注获取更多资讯！

2007 年计算机科学基础答案

一、 1、 A 2、 A 3、 5 4、 C

二、 程序填空

P2!=NULL

P1

三、 程序理解

1、 23

2、 abc123abcd4ef

3、 13045

四、 C语言编程

```
1 #include<iostream>
2 #include<cstdio>
3 #include<cstring>
4 #include<string>
5 #include<cmath>
6 #include<algorithm>
7
8 using namespace std;
9
10 int main()
11 {
12     int N,f,pre = 0,ans = 0;
13     cin>>N;
14     for(int i = 0;i < N;i++)
15     {
16         cin>>f;
17         if(f > pre) ans += 6 * (f - pre) + 5;
18         else ans += 4 *(pre - f) + 5;
19         pre = f;
20     }
21     cout<<ans<<endl;
22     return 0;
23 }
```



数据结构答案

1、 C

2、 B

3、 D

4、 D

5、 25, 15, 20, 47, 35, 21, 68, 84, 21

6、 (1)

结点	0	1	2	3	4
入度	2	2	2	1	1
出度	1	2	0	4	1

(2)

	0	1	2	3	4
0	0	∞	1	∞	∞
1	∞	0	1	1	∞
2	∞	∞	0	∞	∞
3	1	1	1	0	1
4	1	∞	∞	∞	0

(3) $0 \rightarrow 3 \rightarrow 4$

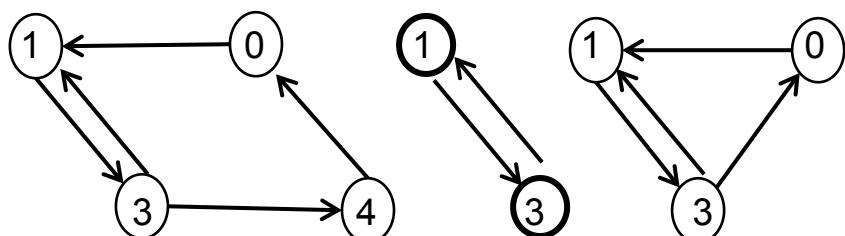
$1 \rightarrow 0 \rightarrow 3$

$2 \rightarrow 1 \rightarrow 3$

$3 \rightarrow 1$

$4 \rightarrow 3$

(4)



7、 ++child

root

List[child]

8、删除x结点，只能删除其左右子树，然后才能将x删除。

```
void DeleteXTree(BiTree bt){
```

```
    if(bt){
```

```
        DeleteXTree(bt->lchild);
```

```
        DeleteXTree(bt->rchild);
```

```
        free(bt);
```

```
    }
```

```
}
```

```
void Search(BiTree bt,ElemType x){
```

```
    BiTree Q[];
```

```
    if(bt){
```

```
        if(bt->data==x){
```

```
            DeleteXTree(bt);
```

```
            exit(0)
```

```
        }
```

```
}
```

```
Init Queue(Q);
```

```
EnQueue(Q,bt);
```

```
while(!IsEmpty(Q)){
```

```
DeQueue(Q,p);

if(p->lchild)

    if(p->lchild->data==x){

        DeleteXTree(p->lchild);

        p->lchild==NULL;

    }

else

    EnQueue(Q,p->lchild);

if(p->lchild)

    if(p->rchild->data==x){

        DeleteXTree(p->rchild);

        p->rchild=NULL;

    }

else

    EnQueue(Q,p->rchild);

}
```



《浙大软件考研 878》微信公众号，扫描二维码关注获取更多资讯！

2008 年计算机科学基础答案

一、 1、 C 2、 C 3、 e 4、 $3*i - 1$

5、 argv[argc]

c

$c - 'A' + 'Q'$

三、 程序理解

1、 2

2、 252

3、 6421357

数据结构

一、 选择题

1、 (1) 2、 (4) 3、 (2) 4、 (3)

二简答题

(1) $0(\log_2 n)$

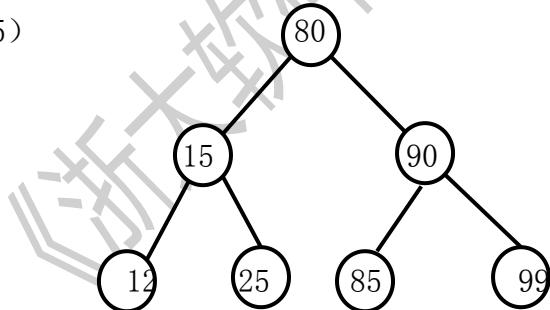
(2) 15, 20, 21, 25, 84, 27, 68, 35, 47

(3) a: 01 b: 0000 c: 0001 d: 001 e: 1

注：（哈夫曼树不是唯一的，编码也不唯一，但是权值之和一定是最小的）

(4) 出；入

(5)



具体步骤详解类比2010年。

三、 用递归算法

```
int BtreeHeight(BinTreeNode *BT) {
```

```
Int lh, rh;  
If (BT==NULL)  
    return 0;  
lh=BtreeHeight(BT->leftChild);  
rh=BtreeHeight(BT->rightChild);  
return (lh>rh?lh:rh);  
}
```

四、currentPos=L;
currentPos=NextPos;
CurrentPos=PreviousPos;

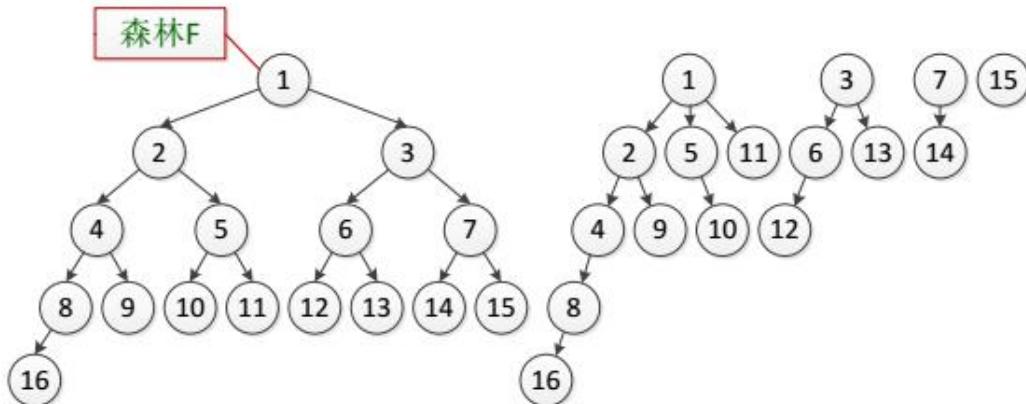


二零一〇年计算机专业基础（A卷）参考答案

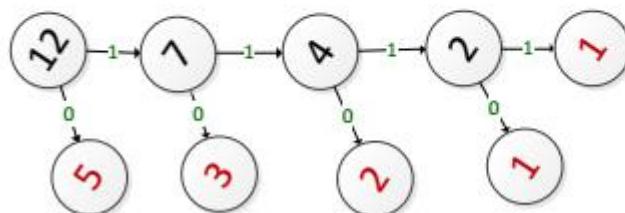
第一部分：数据结构

一、单项选择题

- 1、A 由于链表不带头结点，所以 A 即可完成在第一个元素之前插入新节点，并更改链表指针。B、D 选项先进行 $h=t$ 操作，结果链表和 h 、 t 都丧失联系；C 选项其实是用新节点替换链表首节点。（==应为=）
- 2、C 第一个出栈元素为 4，则可断定在 4 之前入栈的所有元素，在 4 入栈后都还按后进先出的原则压在栈中，且在 4 之后入栈的所有元素，在 4 出栈之前尚未入栈。所以在栈顶元素 4 出栈后，栈中由栈底到栈顶依次为 1、2、3，因此在 1、2、3、4 四个元素中 1 最后出栈。由于元素 5 可能在 1 出栈之前入栈，也可能在 1 出栈之后入栈，所以最后出栈的元素可能是 1，也可能是 5。
- 3、B 假设二叉树根节点为 P，左子树根节点为 L，右子树根节点为 R，则前序遍历序列为 $P(L)_前(R)_后$ ，中序遍历序列为 $(L)_中 P(R)_中$ ，按题意 $P(L)_前(R)_后 == (R)_中递 P(L)_中递$ ，故 $(R)_中递$ 为空（即根节点无右子树）且 $(L)_前 == (L)_中递$ ，同理 P 的左子树 L 无右子树。以 L 为根节点递推下去可知任一节点无右子树。
- 4、C 用数组存储完全二叉树时，节点 $i(1 < i \leq n)$ 的双亲节点为 $[i/2]$ ，故下标为 17 的节点的祖先节点下标为 8、4、2、1；下标为 19 的节点其祖先下标依次为 9、4、2、1，因此最近的公共祖先下标为 4。
- 5、D 如下图所示，森林中树的个数为 4 个，最大的树含有节点数目为 9 个



- 6、B 一棵 m 阶 B 树的根节点可以有 2 到 m 棵子树。
- 7、B 设高度为 h 的 AVL 节点数至少为 $N(h)$ ，则 $N(h+2) = N(h+1) + N(h) + 1$ ；初始时 $N(0)=1$ ， $N(1)=2$ ，故 $N(2)=4, N(3)=7, N(4)=12, N(5)=20, N(6)=33$ 。故该 AVL 树的高度至多为 5。
- 8、C 经过编码，a 的编码为 10 (2bit), b 的编码为 110 (3bit), c 编码为 0 (1bit) d、e 编码分别为 1110、1111 (各 4bit)；故总位数为 $2*3+3*2+1*5+4*1+4*1=25$ bit



9、C 边<v3,v4>存在，故 A 错；<v2,v5>存在故 B 错；<v3,v5>存在故 D 错。

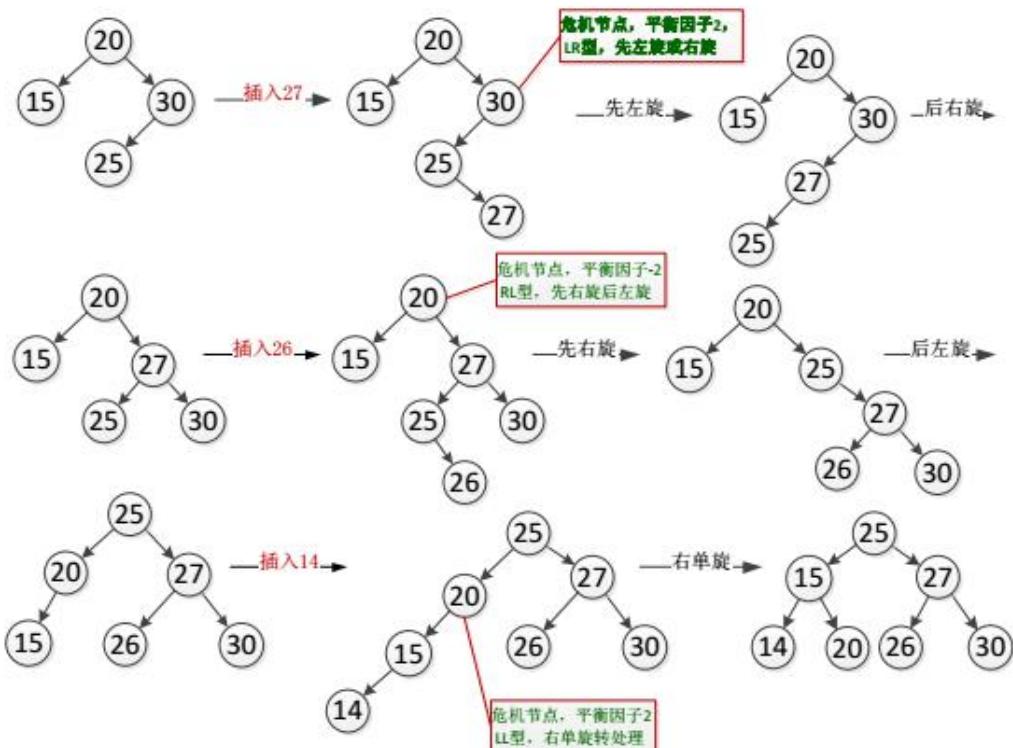
10、A 第 i 列值全为 ∞ ，则第 i 个顶点的入度为 0。在 AOE 网中，只有一个顶点入度为 0，即为源点。表示实际工程计划的 AOE 网应该是无环的，并且存在唯一的入度为 0 的开始顶点和唯一的出度为 0 的完成顶点。

11、C 简单选择排序和冒泡排序都会首先使待排序序列的最值就位，经过 i 轮简单插入排序后，序列的前 $i+1$ 个元素(或后 $i+1$ 个)将有序，而其它 $n-i-1$ 个均未移动。堆排序经过第一轮排序后， $[n/2]$ 和 $[n]$ 是有序的，即 $r[3]$ 和 $r[6]$ 应有序，而其它元素均未移动。题中的处理结果直接看好像是以 46 为枢轴进行快速排序的。但如果以 46 为枢轴进行一轮快速排序，结果是 (38, 40, 46, 56, 84, 79)，只能推测是以 56 为枢轴记录进行一轮快速排序的，但究竟具体排过程如何进行才能得到题目中的结果，我现在还没搞清楚，望高手指点。(题我没有抄错)

12、D 第一个元素 46 和第五个元素 38 交换，84 和 79、56 和 13、40 和 27 交换。

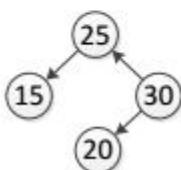
二、分析题

1、二叉排序树的插入过程如下图所示：



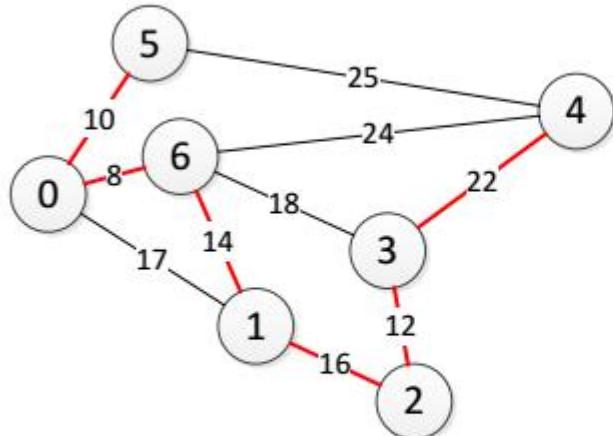
2、(1) 该查找树判别函数是不正确的。此函数的判别依据是：根节点的数值大于其左子树根节点的数值而小于其右子树的数值。而符合此条件的二叉树并不一定是二叉排序树，如下图。正确的判别方法是根据二叉排序树的定义或其中序遍历有序的特性来进行的，即根节点的数值大于其左子树的最大值（即其中序遍历前驱节点的数值）而又小于其右子树的最小值（即其中序遍历后继结点的数值）。

(下图表示的二叉树能通过函数 IsSearchTree 的测试，但它不是二叉排序树。)



(2) 该最小堆判别函数是正确的。

3、该邻接矩阵所表示的无向图入下所示。



(1) G 的最小生成树代价值为 82，其最小生成树的 n-1 条边为红线所示。

(2) 从顶点 0 深度优先遍历的结果序列为：0、1、2、3、4、5、6；图的存储结构一旦确定，其遍历序列也就唯一。

(3) 节点的扩展顺序为 0、6、5、1、2、3、4。

4、(1) 按序列顺序插入关键字之后的哈希表如下图所示：

No	0	1	2	3	4	5	6	7	8	9	10	11	12
Key				7			72	33	59	48	36	46	25
collision count				4			2	0	1	0	0	3	0

(2) 在查找成功的情况下，查找过程中比较的次数为冲突次数加 1。

$$ASL_{succ} = (5+3+1+2+1+1+4+1) / 8 = 18/8$$

(3) 对于关键字 8，其探测序列为：8、9、7、12、4；当探测到 4 时，由于该位置上关键字为空，故断定查找不成功。



二零一一年计算机专业基础（A 卷）参考答案

一、单项选择题（每题 2 分，共 24 分）

1. D

[分析]当顺序查找二维数组中一个元素时，需要逐一遍历数组中每个元素与所查找元素比较，故时间复杂度为 $O(n^2)$ 。

2. B

[分析]输出序列为递减序列则第一个输出元素为 n，元素需要全部入栈，故栈的容量为 n。

3. A

4. B

5. B

[分析]简单选择排序、冒泡排序、快速排序经过一趟排序均有一个元素被放到最终位置，观察（1）知没有一个元素在最终位置，只能选 B，是分组为 3、2、1 的希尔排序。

6. D

7. D

[分析]堆中的子结点大小不定，通过层次遍历不一定能得到递增序列。

8. C

[分析]n 个对象所组成的哈夫曼树的结点数为 $2n-1$ 。

9. A

10. D

11. B

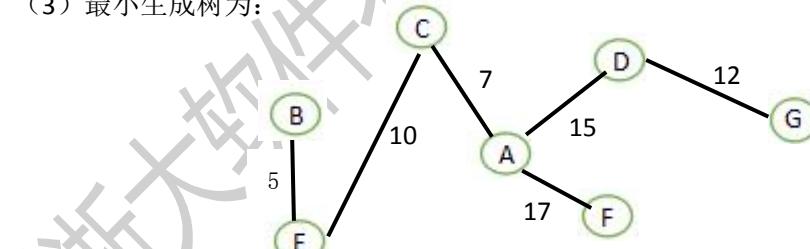
12. C

二、分析题（共 36 分）

1、（1）深度优先遍历序列为：ACBEFDG；

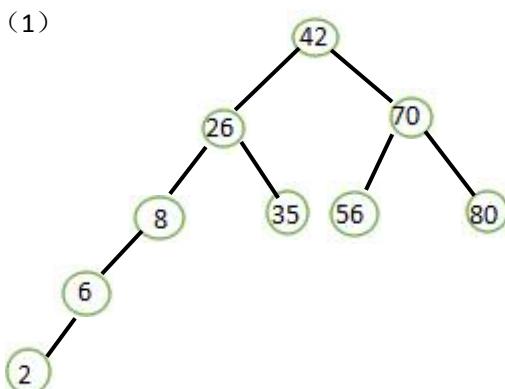
（2）宽度优先遍历序列为：ACDEFBG；

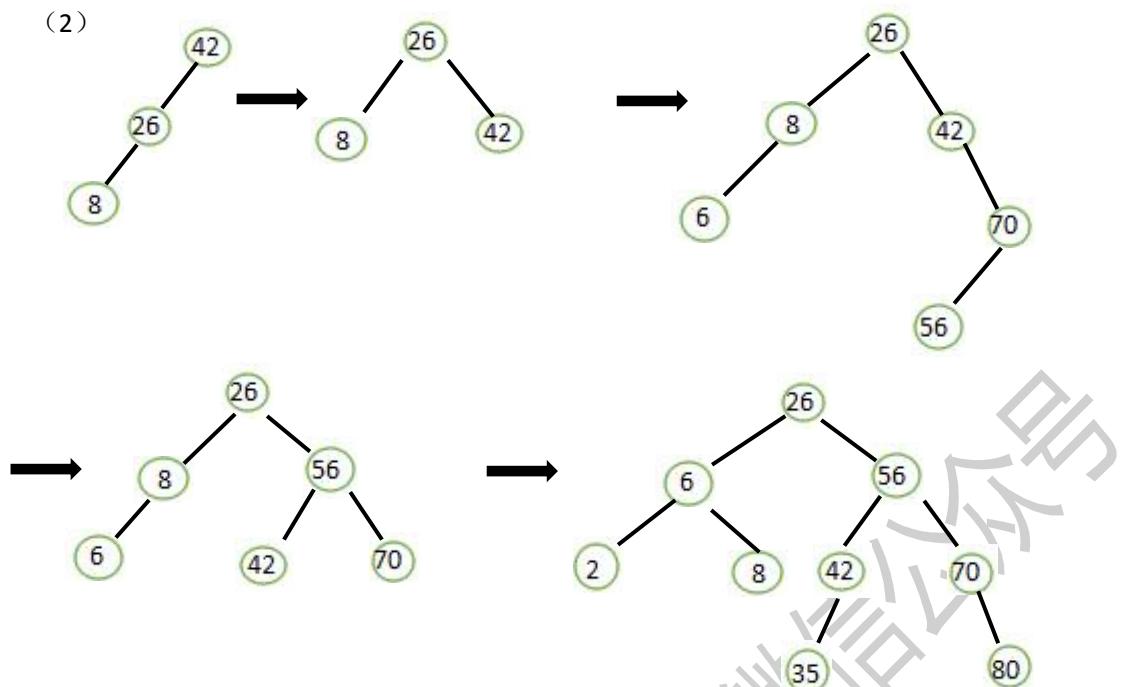
（3）最小生成树为：



故最小生成树的代价为：66。

2、（1）





- 3、 (1) $j \leq n-j+1$;
- (2) $r[j] < r[min]$;
- (3) $min \neq i$;
- (4) $Max == i$;

[分析]本题为简单选择排序的优化，减少了循环次数。每次找两个元素一个最小一个最大，分别放到第一、最后一位。整个程序中都是对数组下标进行操作，完成排序任务。

4、算法的设计思想：依次逐一扫描入栈出栈序列（即由‘S’和‘P’组成的字符串），每次扫描至任一位置均需检查出栈次数（‘X’个数）是否小于入栈次数（即‘S’的个数），若大于则为非法序列。扫描结束后，再判断入栈和出栈次数是否相等，若相等，则序列合法；否则为非法序列。

```
Int Judge(Char A[]){
    //判断字符数组 A 的序列是否合法，合法返回 1，不合法返回 0
    Int i=0;
    Int j=k=0;           //i 为下标，j、k 分别为字母 S 和 X 的个数
    While(A[i]!='\0'){  //未到字符串尾
        Switch(A[i]){
            Case 'S': j++;break;          //入栈次数加一
            Case 'X': k++;;
            If(k>j){
                return 0;
                exit(0);
            }
        }
        i++;      //指针向后移动
    }
}
```

《浙大软件考研 878》微信公众号，扫描二维码关注获取更多资讯！

```
If(j!=k)  
    return 0;  
Else return 1;  
}
```

另解：入栈后，栈内元素个数加 1，出栈后，栈内元素减 1，因此，可以将判定一组出入栈序列是否合法，转化为一组+1、-1 组成的序列，它的任意前缀子序列的累加和不小于 0，则合法；否则非法。



二零一二年计算机专业基础（A 卷）参考答案

一、单项选择题（每题 2 分，共 24 分）

13. B

14. B

15. B

16. D

[分析]前序序列和后序序列相反，则该树为单支树，故不存在度为2的结点。

17. C

[分析]由 9、2、5、7、构成的哈夫曼树的权值为 44。

18. A

19. A

[分析]6个元素组成的平衡二叉树的高度为3。

20. C

[分析] 的拓扑序列唯一但是其没有出度为 2 的结点。

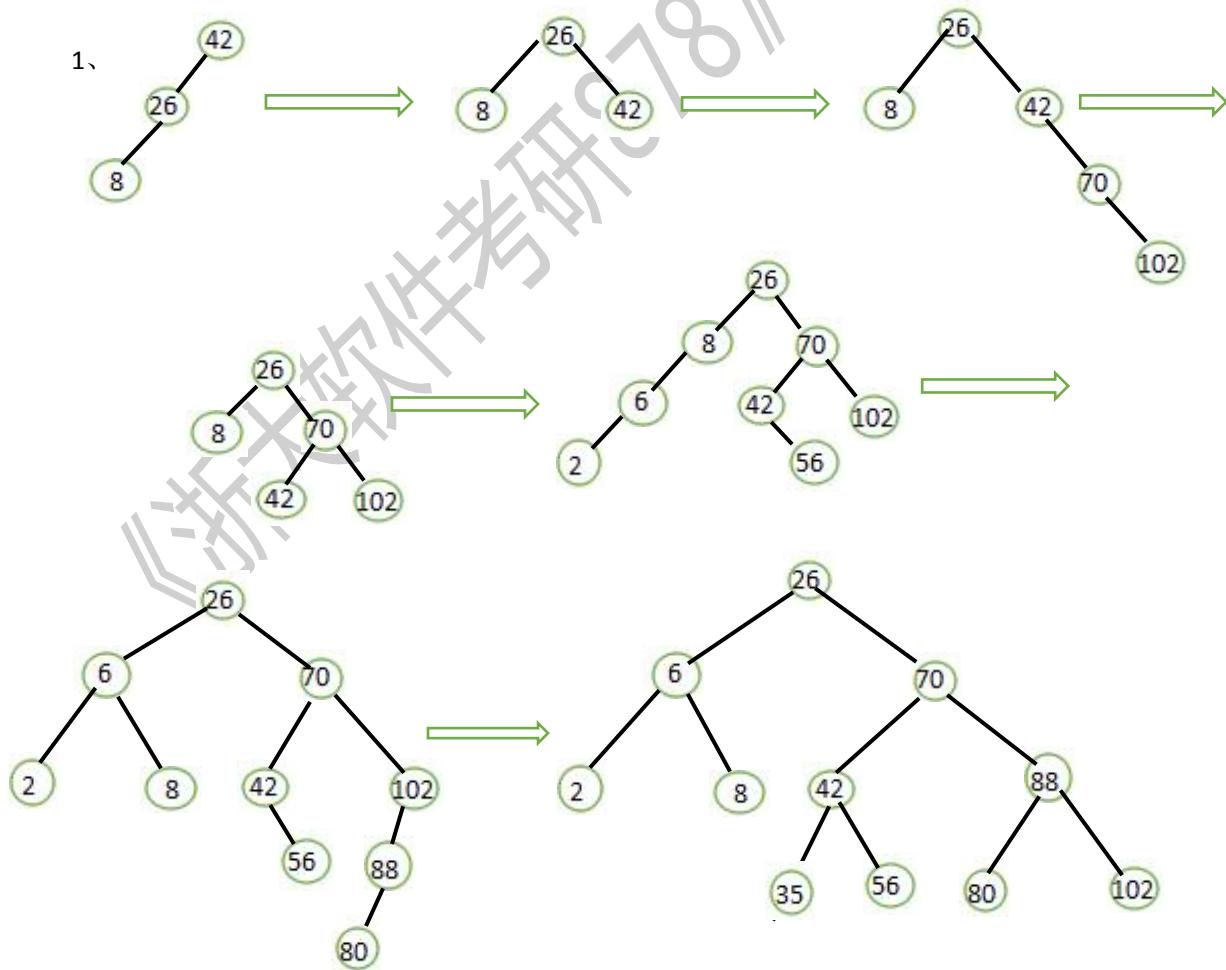
21. B

22. B

23. C

24. D

二、分析题（共 36 分）

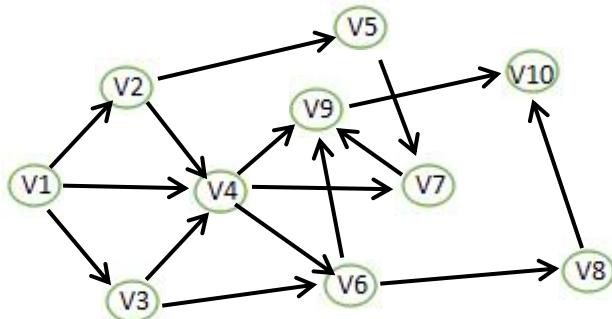


5、 (1) $H \rightarrow \text{Elements}[1]$

(2) $\text{Child} \neq \text{Size} \& H \rightarrow \text{Elements}[\text{Child}+1] > H \rightarrow \text{Elements}[\text{Child}]$

(3) $H \rightarrow \text{Elements}[i] = H \rightarrow \text{Elements}[\text{Child}]$

6、 根据邻接矩阵画出该有向图：



(a) 深度遍历序列为：v1, v2, v4, v6, v8, v10, v9, v7, v5, v3;

广度遍历序列为：v1, v2, v3, v4, v5, v6, v7, v9, v8, v10。

(b) 拓扑序列为：v1, v2, v3, v4, v5, v6, v7, v8, v9, v10。

7、 (1) 距离 l 和 G 的最近公共祖先结点为 C。

(2) 后序遍历最后访问根结点，即在递归算法中，根是压在栈底的。本题要找 p 和 q 的最近公共祖先结点下不失一般性，设 p 在 q 的左边。算法思想采用后序非通归算法，栈中存放二叉树结点的指针，当访问到某结点时，栈中所有元素均为该结点的祖先。后序遍历必然先遍历到结点 p，栈中元素均为 p 的祖先。先将栈复制到另一辅助栈中。继续遍历到结点 q 时，将栈中元素从栈顶开始逐个到辅助栈中去匹配，第一个匹配(即相等)的元素就是结点 p 和 q 的最近公共祖先。

算法实现如下：

```

typedef struct{
    BiTree t;
    int tag; //tag=0 表示左子女已被访问, tag=1 表示右子女已被访问
}stack; //栈, 容量足够大
stack s[ ],s1[ ];
BiTree Ancestor(BiTree ROOT,BiTNode *p,BiTNode *q) {
    //本算法求二叉树中 p 和 g 指向结点的最近公共结点
    top=0 ;bt =ROOT ;
    while(bt!=NULL||top>0) {
        while (bt !=NULL&&bt !=p&&bt !=q)//结点入栈
            while (bt !=NULL){
                s[++top].t=bt;
                s[top].tag=0;
                bt=bt->lchild;
            } //沿左分支向下
        while(top!=0&&s[top].tag==1){
            if(s[top].t==p){
                //处理逻辑
            }
        }
    }
}
  
```

/假定 P 在 q 的左侧，遇到 p 时，栈中元素为 p 的祖先

if(s[top].t==p){

```
for(i=1; i<=top;i++)
    s1[i]=s[i];
    top1=top;
} //将栈中的元素转入辅助栈 s1 保存

if(s[top]).t==q)           //找到 q 结点

for(i=top;i>0; i--){ 将栈中元素的树结点到 s1 中去匹配
    for(j=top1;j>0;j--)
        if(s1.[j].t==s[i].t)//p 和 q 的最近公共祖先已找到
            return s[i].t;
}
top--;                      //退栈

}//while
if(top!=0) {
    s[top].tag=1;
    bt=s[top].t->rchiid
}
//沿右分支向下遍历
}//while
return NULL;
}

本答案为王道《数据结构联考复习指导》上的答案，与真题要求稍微有点不同本参考答案为用栈，而题要求用数组。只需再用两个数组存储栈中元素即可。
```



二零一三年计算机专业基础（A 卷）参考答案

一、单项选择题（每题 2 分，共 24 分）

25. C

[分析] 栈的容量至少为 4，具体过程参见《栈和队列的应用》。

26. D

[分析] 假设度为 2 的结点数为 N_2 ，度为 1 的结点数为 N_1 ，叶子结点数为 N_0 ，由度的总和加一等于结点数。即： $N_2+N_1+N_0=2N_2+N_1+1$ ；度为二的结点数等于叶子结点数减一，即 $N_2+1=N_0$ ；结合上式得： $2N_2=49$ 。故这样的树不存在。

27. C

[分析] 由前序遍历的特点知 A 为根结点，C 选项不可能。

28. D

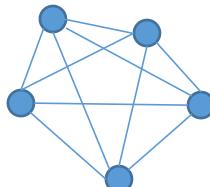
[分析] 高度为 h 的满二叉树的结点数为 2^h-1 。

29. B

[分析] 由 2、1、4、5、7、3 构成的哈夫曼树的高度为 4，故最长编码为 4。

30. A (该知识点已经从考纲中删除)

31. A



32. D

[分析] $H(6)=6\%11=6$; $H(25)=25\%11=3$; $H(39)=39\%11=6$, 发生冲突, 用线性探测法处理冲突 $hi(39)=(H(39)+1)\%11=7$; $H(61)=61\%11=6$, 用线性探测法处理冲突 $hi(61)=(H(61)+2)\%11=8$; 当存放 3、6、8 时均会发生冲突, 只有存入 10 时不会发生冲突。

33. B

[分析] 6 个元素组成的平衡二叉树的高度为 3。

34. B

[分析] 在最好的情况下, 表中元素已经有序, 此时每插入一个元素, 都只需比较一次而不需要移动元素, 因而时间复杂度为 $O(n)$ 。

35. B

[分析] 若干趟排序结果元素没有一个在最终位置, 只能选 B。堆排序、选择排序、冒泡排序每趟排序后都会有一个元素被放到最终位置。

36. C

[分析] 由堆的性质知 C 为大顶堆。

二、分析题（36 分）

1、

(1) $v1 \rightarrow v2 \rightarrow v3 \rightarrow v5 \rightarrow v6 \rightarrow v4$

(2) $v1 \rightarrow v2 \rightarrow v4 \rightarrow v3 \rightarrow v5 \rightarrow v6$

(3) 关键路径为： $v1 \rightarrow v4 \rightarrow v6$ ，其权值为 25。

2、

(1) $H(26)=26\%13=0$; 未发生冲突, 26 的最终位置下标为 0;

$H(8)=8\%13=8$; 未发生冲突, 8 的最终位置下标为 8;

$H(70)=70\%13=5$; 未发生冲突, 70 的最终位置下标为 5;

$H(6)=6\%13=6$; 未发生冲突, 6 的最终位置下标为 6;
 $H(56)=56\%13=4$; 未发生冲突, 56 的最终位置下标为 4;
 $H(2)=2\%13=2$; 未发生冲突, 2 的最终位置下标为 2;
 $H(80)=80\%13=2$; 发生冲突, 用二次探测法处理冲突得 80 的最终位置下标为 3;
 $H(30)=30\%13=4$; 发生冲突, 用二次探测法处理冲突得 30 的最终位置为下标为 10。

故插入完成后的哈希表为:

散列地址	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字	26		80	2	56	70	6		8		30		
冲突次数	0		0	1	0	0	0		0		3		

(2) 平均 (成功) 查找次数为:

$$ASL_{\text{成功}} = (1+1+2+1+1+1+1+4) / 8 = 1.5$$

(3) 哈希表主要可以在 $O(1)$ 时间内对查找对象定位, 但是事实上, 如果输入集合不确定的情况下, 可能出现大量的冲突, 虽然有很多好的哈希函数, 但是随着随机输入, 大量冲突还是不可避免, 可能出现最差情况。所以, 哈希表如果用在输入集合确定 (即以后只会做查询操作) 的情况下, 选择合适的函数函数和解决冲突的方法 (perfect hash) 可以在 $O(1)$ 时间内完成查找。

二叉树支持动态的插入和查找, 保证操作在 $O(\text{height})$ 时间, 这就是完成了哈希表不便完成的工作, 动态性。但是二叉树有可能出现 worst-case, 如果输入序列已经排序, 则时间复杂度为 $O(N)$ 。

2、

- (1) $H[\text{child}] < H[\text{child}+1]$
- (2) $H[i] > H[\text{child}]$
- (3) $H[i] = H[0]$
- (4) $H[1]$

3、

- (1) $4 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 3$
- (2)

2000~2001 学年《程序设计基础-C》试卷

考试时间：2000 年 12 月 31 日上午 8:30-10:30

注：请将答案写在答卷上

一、单项选择题（每小题 2 分，共 30 分）

1. C 语言程序中可以对程序进行注释，注释部分必须用符号_____括起来。
A、‘{’和‘}’ B、‘[’和‘]’ C、“/*”和“*/” D、“*”和“/*”
2. 下列运算符中，优先级最低的是_____。
A、* B、!= C、+ D、=
3. 若变量已正确定义，表达式 (j=3, j++) 的值是_____。
A、3 B、4 C、5 D、0
4. 已知字符 ‘a’ 的 ASCII 码为 97，执行下列语句的输出是_____。

```
printf ("%d, %c", 'b', 'b'+1);
```

A、98, b B、语句不合法 C、98, 99 D、98, c
5. 表达式 strcmp("box", "boss") 的值是一个_____。
A、正数 B、负数 C、0 D、不确定的数
6. 数组定义为 int a[3][2]={1, 2, 3, 4, 5, 6}，数组元素_____的值为 6。
A、a[3][2] B、a[2][1] C、a[1][2] D、a[2][3]
7. 要调用数学函数时，在#include 命令行中应包含_____。
A、"stdio.h" B、"string.h" C、"math.h" D、"ctype.h"
8. 判断 i 和 j 至少有一个值为非 0 的表达式是_____。
A、i!=0 && j!=0 B、i*j!=0 C、!(i==0 || j==0) D、i&j
9. 若 a 是基本整型变量，c 是单精度实型变量，输入语句_____是错误的。
A、scanf("%d,%f", &a, &c); B、scanf("d=%d, c=%f", &a, &c);
C、scanf("%d%f", &a, &c); D、scanf("%d%f", a, c);
10. 若变量已正确定义并且指针 p 已经指向某个变量 x，则(*p)++相当于_____。
A、p++ B、x++ C、*(p++) D、&x++
11. 若 p1、p2 都是整型指针，p1 已经指向变量 x，要使 p2 也指向 x，_____是正确的。
A、p2=p1; B、p2=**p1; C、p2=&p1; D、p2=*p1;
12. 下列程序段的输出是_____。

```
int c[]={1, 3, 5};  
int *k=c+1;  
printf("%d", *++k);
```

A、3 B、5 C、4 D、6
13. 不正确的赋值或赋初值的方式是_____。
A、char str[]="string";
B、char str[7]={'s', 't', 'r', 'i', 'n', 'g'};
C、char str[10]; str="string";
D、char str[7]={'s', 't', 'r', 'i', 'n', 'g', '\0'};
14. 对于以下结构定义，++p->str 中的++加在_____。

```
struct {  
    int len;
```

- ```
char *str;
} *p;
A、指针 str 上 B、指针 p 上 C、str 指的内容上 D、以上均不是
15. 对于如下说明，语法和语义都正确的赋值是_____。
int c, *s, a[]={1, 3, 5};
A、c=*s; B、s[0]=a[0]; C、s=&a[1]; D、c=a;
```

## 二、填空题（每小题 2 分，共 20 分）

1. 写出下列程序段的输出结果。

```
float x1, x2;
x1=3/2;
x2=x1/2;
printf("%d, %.1f", (int)x1, x2);
```

2. 表达式  $(7 << 1) >> 2^2$  的值是\_\_\_\_\_。

3. 写出下列程序段的输出结果。

```
#define A 10
#define B (A<A+2)-2
printf("%d", B*2);
```

4. 写出判断字符变量 c 是英文字母的表达式\_\_\_\_\_。

5. 写出下列程序段的输出结果。

```
int k, x;
for (k=0, x=0; k<=9&&x!=10; k++)
 x+=2;
```

```
printf ("%d, %d", k, x);
```

6. 写出下列程序段的输出结果。

```
printf("%d, %d", NULL, EOF);
```

7. 对于以下递归函数 f，调用 f(3) 的返回值是\_\_\_\_\_。

```
f (int n)
{ return ((n>0) ? 2*f(n-1)+f(n-2) : -1); }
```

8. 写出下列程序段的输出结果。

```
char str[]="hello\eworld\n";
printf("%d, %c\n", sizeof(str), *(str+10));
```

9. 输入 12345#后，写出下列程序的输出结果。

```
void main()
{ char c;
 for(c=getchar(); getchar()!='#'; c=getchar())
 putchar(c);
}
```

10. 执行程序 `find -nx ould text.txt` 时，`*++argv[1]` 的值是\_\_\_\_\_。

## 三、程序阅读题（每题5分，共20分）

1. 若输入 3 1 2 3 <回车>，则以下程序的输出结果是\_\_\_\_\_。

```
#include "stdio.h"
void main()
{ int i, j, n, a[10];
```

```
scanf("%d", &n);
for (i=0; i<n; i++)
 scanf("%d",&a[i]);
for (i=0; i<n; i++)
 for (j=0; j<n; j++)
 printf("%d, ", a[(i+j)%n]);
}
```

2. 若输入89 76 24 25 9 8 11 16 35 4 <回车>， 则以下程序的输出结果是\_\_\_\_\_。

```
#include "stdio.h"
#define N 10
main()
{
 int x[N], y1[N], y2[N];
 int i, j, n1, n2, t, p;
 n1=n2=0;
 for(i=0;i<N;i++){
 scanf("%d", &x[i]);
 if(x[i] % 2==0) y1[n1++]=x[i];
 else y2[n2++]=x[i];
 }
 for(i=1; i<n1; i++)
 for(j=0; j<n1-1; j++)
 if(y1[j]>y1[j+1]){
 t=y1[j]; y1[j]=y1[j+1]; y1[j+1]=t;
 }
 for(i=0; i<n2-1; i++){
 p=i;
 for(j=i+1; j<n2; j++)
 if(y2[p]<y2[j]) p=j;
 if(p!=i){
 t=y2[i]; y2[i]=y2[p]; y2[p]=t;
 }
 }
 for(i=0; i<n1; i++) printf("%d, ", y1[i]);
 for(i=0; i<n2; i++) printf("%d, ", y2[i]);
}
```

3. 以下程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
int z;
void p(int *x, int y)
{
 ++*x;
 y--;
 z=*x+y+z;
```

```
 printf("%d, %d, %d#", *x, y, z);
}
void main()
{
 int x=1, y=5, z=9;
 p(&x, y);
 printf("%d, %d, %d#", x, y, z);
}
```

4. 若输入 -6+15\*3/5=<回车>， 则以下程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
void main()
{
 int m=0, sum=0;
 char c, oldc='+';
 do {
 c=getchar();
 if(c<='9'&&c>='0') m=10*m+c - '0';
 else {
 if(oldc=='+') sum += m;
 else sum -= m;
 m=0;
 oldc=c;
 printf("%d, ", sum);
 }
 } while(c!='=');
}
```

#### 四、程序填空题（每空 2 分，共 20 分）

1. 下列程序的功能是创建单向链表。

```
#include <stdio.h>
#include <alloc.h>
struct link {
 char name[10];
 int mark;
 struct link * next;
};

void insert(char *, int);
struct link * head = NULL;
main()
{
 char name[10];
 int mark;
 struct link * t;
 while (1) {
 scanf("%s %d", name, &mark);
```

```
 if (strcmp(name, "#") == 0) break;
 _____(1)_____;
 }
 for (t=head; _____(2)_____)
 printf("<%s>: %d\n", t->name, t->mark);
}
void insert(char * name, int mark)
{
 struct link * p;
 p = _____(3)_____ ;
 strcpy(p->name, name);
 p->mark = mark;
 _____(4)_____;
 if (head != NULL) _____(5)_____;
 head = p;
}
```

2. prog.c 是带命令行参数的程序。运行 prog filename 时，将文本文件 filename 中的内容原样输出；运行 prog -u filename 时，将文本文件 filename 中的所有字母以大写方式输出（其余字符原样输出）；运行 prog -d filename 时，只将文本文件 filename 中的数字输出。

```
include <stdio.h>
include <ctype.h>
void main(int argc, char * argv[])
{
 FILE *fp;
 char * filename, ch;
 int tag=0;
 if (argc <2 || argc>3) { printf("Argument error\n"); return; }
 if (argc == 3) {
 if (strcmp(argv[1], "-u") == 0) tag=1;
 else if (strcmp(argv[1], "-d") == 0) tag=2;
 else {printf("Option error\n"); return;}
 _____(6)_____;
 }
 else filename = argv[1];
 if (_____7_____) {
 printf("Cannot open file\n"); return;
 }
 while (_____8_____) {
 ch = fgetc(fp);
 if (tag == 0) putchar(ch);
 else if (tag == 1) _____(9)_____;
 else if (_____(10)_____) putchar(ch);
 }
 fclose(fp);
}
```

五、编程题（共 10 分，混合班学生做第2题，其他学生做第1题）

1. [程序说明] (10 分)

输入 m 和 n ( $m \geq n \geq 0$ ) 后，计算下列表达式的值并输出。

$$\frac{m!}{n!*(m-n)!}$$

要求将计算阶乘的运算写成函数 fact(n)，函数返回值的类型为 float。

2. [程序说明] (10 分)

编写函数 strcmp(char \* s1, char \* s2)。若 s1 和 s2 均是数字串（包括+/-号），则按照十进制整数大小进行比较；否则按照 Ascii 序进行比较。s1 大于/等于/小于 s2 时，分别输出 1/0/-1。可以直接调用 atoi 函数。

**2000~2001 学年《程序设计基础 C》参考答案**

**一、单项选择题（每小题 2 分，共 30 分）**

| 题号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 选择 | C | D | A | D | A | B | C | B | D | B  | A  | B  | C  | A  | C  |

**二、填空题（每小题 2 分，共 20 分）**

1 1, 0.5                          2 1

3 -3                                  4 c>='a'&&c<='z'||c>='A'&&c<='Z'

5 5, 10                                  6 0, -1

7 -17                                          8 13, d

9 13                                          10 'n'

**三、程序阅读题（每题 5 分，共 20 分）**

1 1, 2, 3, 2, 3, 1, 3, 1, 2,                          2 4, 8, 16, 24, 76, 89, 35, 25, 11, 9,

3 2, 4, 6#2, 5, 9#                                  4 0, -6, 9, 6, 1,

**四、程序填空题（每空 2 分，共 20 分）**

(1) insert(name, mark)

(2) t!=NULL; t=t->next

(3) (struct link \*)malloc(sizeof(struct link))

(4) p->next = NULL

- 
- (5) p->next = head
- 
- (6) filename = argv[2]
- 
- (7) (fp=fopen(filename, "r")) == NULL
- 
- (8) !feof(fp)
- 
- (9) putchar(toupper(ch))
- 
- (10) tag== 2&& isdigit(ch)

五、编程题（共 10 分，混合班学生做第 2 题，其他学生做第 1 题）

1、

```
#include <stdio.h>
float fact(int);
main()
{
 int m, n;
 float val;
 scanf("%d%d", &m, &n);
 val = fact(m) / (fact(n) * fact(m-n));
 printf("Value: %f\n", val);
}
float fact(int n)
{
 if (n <= 0) return 1;
 return(fact(n-1) * n);
}
```

float fact(int n)  
{ int i; float f=1;  
for(i=1; i<=n; i++)  
 f=f\*i;  
return f;  
}

2、

```
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
int strcmp(char * s1, char * s2)
{
 int token =0, val1, val2, n, i;

 for (i=0; s1[i]!='\0'; i++)
 if (s1[i]!='+' && s1[i] !='-' && !isdigit(s1[i])) { token = 1; break; }
 for (i=0; !token && s2[i]!='\0'; i++)
 if (s2[i]!='+' && s2[i] !='-' && !isdigit(s2[i])) { token = 1; break; }
 if (token == 0)
 {
 val1 = atoi(s1);
 val2 = atoi(s2);
```

```
if (val1 > val2) n = 1;
else if (val1 < val2) n = -1;
else n = 0;
return n;
}
else
{
 for (i=0; s1[i]!='\0' && s2[i]!='\0'; i++)
 if (s1[i] > s2[i]) return 1;
 else if (s1[i] < s2[i]) return -1;
 if (strlen(s1) == strlen(s2)) return 0;
 if (s[i] == '\0') return -1;
 else return 1;
}
}
```

## 2000~2001 学年“程序设计基础--C”试卷

考试时间: 2001 年 6 月 27 日下午 2:00-4:00

注: 请将答案写在答题卷上

### 一、单项选择题(每题 2 分, 共 30 分)

1. \_\_\_\_\_ 是合法的用户自定义标识符。  
A、b-b      B、float      C、<fr>      D、\_isw
2. 若变量已正确定义并赋值, 表达式 \_\_\_\_\_ 不符合 C 语言语法。  
A、a\*b/c;      B、3.14%2      C、2, b      D、a/b/c
3. \_\_\_\_\_ 是不正确的字符常量。  
A、'\n'      B、'1'      C、"a"      D、'\101'
4. 在 C 程序中, 用 \_\_\_\_\_ 表示逻辑值“真”。  
A、1      B、非 0 的数      C、非 1 的数      D、大于 0 的数
5. \_\_\_\_\_ 把 x、y 定义成 float 类型变量, 并赋同一初值 3.14。  
A、float x, y=3.14;      B、float x, y=2\*3.14;  
C、float x=3.14, y=x=3.14;      D、float x=y=3.14;
6. 若变量已正确定义, 表达式 \_\_\_\_\_ 的值不是 2。  
A、2&3      B、1<<1      C、a==2      D、1^3
7. 若变量已正确定义, 语句 “if(a>b) k=0; else k=1;” 和 \_\_\_\_\_ 等价。  
A、k=(a>b)?1:0;      B、k=a>b;  
C、k=a<=b;      D、a<=b ? 0 : 1;
8. 设变量定义为 “int a, b;” , 执行下列语句时, 输入 \_\_\_\_\_, 则 a 和 b 的值都是 10。  
scanf("a=%d, b=%d",&a, &b);  
A、10 10      B、10, 10      C、a=10 b=10      D、a=10, b=10
9. 下列程序的输出结果是 \_\_\_\_\_。  
fun(int a, int b, int c)  
{ c =a\*b; }  
main()  
{ int c;

- ```
fun(2,3,c);
printf("%d\n",c);
}
A、0          B、1          C、6          D、无法确定
10. 在 C 程序中，若对函数类型未加说明，则函数的隐含类型为_____。
    A、int        B、double      C、void       D、char
11. 表达式 strlen("hello") 的值是_____。
    A、4          B、5          C、6          D、7
12. 设变量定义为 “int a[4];” ，则表达式_____不符合 C 语言语法。
    A、*a         B、a[0]        C、a          D、a++
13. 设变量定义为 “int x, *p=&x;” ，则&p 相当于_____。
    A、p          B、*p         C、x          D、*&x
14. 若变量已正确定义，_____不能使指针 p 成为空指针。
    A、p=EOF     B、p=0        C、p='\0'     D、p=NULL
15. 对于以下的变量定义，表达式_____不符合 C 语言语法。
    struct node {
        int len;
        char *pk;
    } x = {2, "right"}, *p = &x;
    A、p->pk      B、*p.pk       C、*p->pk      D、*x.pk
```

二、填空题（每题 2 分，共 20 分）

11. 表示 “ $x \geq y \geq z$ ” 的 C 表达式是_____。
12. 下列程序段的输出是_____。
printf("%o\n", 010<<1^1);
13. 下列程序段的输出是_____。
int k;
float s;
for (k=0, s=0; k < 7; k++)
 s += k/2;
printf("%d, %f\n", k, s);
14. 下列程序段的输出是_____。

```
#define f(a, b, x)  a*x+b
printf("%d, %d\n", f(3,2,1), f(6, 5, f(3, 2, 1)));
```
15. 下列程序段的输出是_____。
main()
{
 int k;
 for(k=1; k<5; k++) {
 if(k % 2) printf("*");
 else continue;
 printf("#");
 }
}
16. 下列程序段的输出是_____。

```
# include <stdio.h>
int f(int x)
{
    static y=1;
```

```
y++;
x += y;
return x;
}
void main()
{ int k;
k=f(3);
printf("%d %d\n", k, f(k));
}
```

17. 下列程序段的输出是_____。

```
int *p, *q, k = 5, j = 6 ;
q = &k ; p = q ; q = &j ;
printf("%d", *p);
```

18. 函数

```
void f(char s[ ], char t[ ])
{ int k=0;
while (s[k]==t[k]) k++;
}
```

等价于

```
void f(char *s, char *t)
{ while (____);
}
```

19. 下列程序段的输出是_____。

```
char *a[2]={"one", "two"}, **p=a;
printf("%s,", *(p++)+1);
printf("%c\n", **p-1);
```

20. 用 `typedef` 定义一个文件指针 FP。

三、程序阅读题（每题4分，共16分）

1. 输入 elephant 后，下列程序的输出结果是_____。

```
# include <stdio.h>
void main()
{ int i=0, k, ch;
static int num[5];
char alpha[ ]={'a', 'e', 'i', 'o', 'u'}, in[80];
gets(in);
while(in[i]){
    for(k=0; k<5; k++)
        if(in[i] == alpha[k]) {
            num[k]++;
            break;
        }
    i++;
}
for(k=0; k<5; k++)
    if(num[k]) printf("%c%d", alpha[k], num[k]);
printf("\n");
}
```

2. 下列程序的输出结果是_____。

```
#include <stdio.h>
void f(int n)
{
    if(n<0) {
        putchar('-');
        n = -n;
    }
    if(n/10)  f(n/10);
    putchar(n%10+'0');
}
main()
{
    f(-610);
}
```

3. 输入 0 10 2 7 后，下列程序的输出结果是_____。

```
#include <stdio.h>
void p(int v[])
{
    int i, k, temp;
    for (i=1; i<4; i++)
        for(k=i-1; k>=0 && v[k]<v[k+1]; k--) {
            temp=v[k];
            v[k]=v[k+1];
            v[k+1]=temp;
        }
}
main()
{
    int k, a[4];
    for(k=0; k<4; k++)
        scanf("%d", &a[k]);
    p(a);
    for(k=0; k<4; k++)
        printf("%d ", a[k]);
}
```

4. 下列程序编译后生成 t.exe，运行 t input output zoo oil 的输出结果是_____。

```
void main(int argc, char * argv[])
{
    int k;
    char *p=argv[1];
    for(k=2; k<argc; k++)
        if(strcmp(argv[k], p)>0)  p=argv[k];
    printf("%s\n", p);
}
```

四、程序填空题（每空2分，共20分）

1. 验证哥德巴赫猜想：任何一个大于 6 的偶数均可表示为两个素数之和。要求将 6~100

之间的偶数都表示成两个素数之和。素数指只能被 1 和自身整除的正整数，1 不是素数，2 是素数。

```
#include <stdio.h>
int prime(int n)      /* 判断 n 是否为素数 */
{
    int k;
    for (k=2; k<=n/2; k++)
        if (n%k==0) return 0;
    return ____(1____);
}
main()
{
    int i, k;
    for (i=6 ; i<=100; i+=2)
        for (k=2; k<=i/2; k++)
            if (__(2)____) {
                printf("%d = %d + %d\n", i, k, i-k);
                break;
            }
}
```

2. 下列程序调用 swap 函数交换 a 和 b 的值，并输出交换后的 a 和 b 的值。程序的输出结果是：After swap 5, 3。

```
void main()
{
    int a, b;
    a = 3, b = 5;
    swap(___(3)____);
    printf("After swap %d, %d", a, b);
}
swap(___(4)____)
{
    int t;
    _____(5)____;
    _____(6)____;
    _____(7)____;
}
```

3. 函数 del 将头指针为 head 的链表中所有的节点全部删除，并释放相应的内存空间。

```
struct node{
    int k;
    struct node *next;
};
struct node *del(struct node *head)
{
    struct node *p, *q ;
    p=head;
    while(___(8)____){
        _____(9)____;
        free(p);
        _____(10)____;
    }
    return NULL;
}
```

五、编程题（第1题4分，第2题10分，共14分）

1. 对任意输入的 x ，用下式计算并打印出 y 的值。

$$y = \begin{cases} e^x & x > 10 \\ 0 & x = 10 \\ 3x + 5 & x < 10 \end{cases}$$

2. 输出文本文件 input.txt 中的非空格字符。

2000~2001 学年下《程序设计基础 C》答案

一、单项选择题（每题2分，共30分）

1 D

6 C

11 B

2 B

7 C

12 D

3 C

8 D

13 A

4 B

9 D

14 A

5 C

10 A

15 B

二、填空题（每题2分，共20分）

1 $x \geq y \quad \&\& \quad y \geq z$ (错1项扣1分)

3 7, 9.0 (错1个数字扣1分)

5 *#*# (错2个字符扣1分)

7 5

9 ne, s (ne给1分, s给1分)

2 21

4 5, 25 (错1个数字扣1分)

6 5, 8 (错1个数字扣1分)

8 $*s = *t, s++, t++$ 或 $*s++ = *t++$

$*s = *t,$ (1分) $s++, t++$ (1分)

10 typedef FILE* FP (错1项扣1分)

三、程序阅读题（每题4分，共16分）

1 a1e2 (错1个扣1分) 2 -610 (错1个扣1分)

3 10, 7, 2, 0 (错1个扣1分) 4 zoo

四、程序填空题（每空2分，共20分）

(1) 1 或 $k > n/2$ 或 $k \geq n/2 + 1$ (2) $\text{prime}(k) \quad \&\& \quad \text{prime}(i-k)$ (错1项扣1分)

(3) $\&a, \&b$ 或 $\&b, \&a$ (错1项扣1分) (4) $\text{int } *x, \text{ int } *y$ 或其它变量名(错1项扣1分)

(5) $t = *x$ (6) $*x = *y$

(7) $*y = t$ (8) $p \neq \text{NULL}$ 或 $p \neq 0$ 或 p

(9) q=p->next

(10) p=q

注: (4)-(7)作为一个整体评分, 实现*x 和*y 交换

五、编程题 (第1题4分, 第2题10分, 共14分)

1、

```
# include "math.h"          头文件、变量说明、输入、输出 1 分
main( )
{ float x, y;
  scanf("%f", &x);
  if(x>10) y=exp(x);      3分
  else if(x==10) y=0;
  else y=3*x+5;
  printf("%f\n", y)
}
```

2、

```
# include <stdio.h>        头文件、变量说明 1 分
main()
{
  FILE *fpin;
  char ch;
  if (( fpin = fopen("input.txt", "r" )) == NULL){    2 分
    printf( "Can't open file input.txt\n" );
    exit(0);
  }
  while ( !feof( fpin ) ){                      2 分
    ch=fgetc(fpin);                            2 分
    if(ch!=' ') putchar(ch);                   2 分
  }
  fclose( fpin );                           关闭文件 1 分
}
```

或 while ((ch=fgetc(fpin))!=EOF) 4 分(读2分, 判断2分)
 if(ch!=' ') putchar(ch); 2 分

浙江大学“程序设计基础-C”试题卷

2001-2002 学年秋季学期

考试时间: 2002 年 1 月 20 日上午 8:30-10:30

注意: 答题内容必须写在答题卷上, 写在本试题卷上无效

一. 单项选择题 (每题 2 分, 共 30 分)

1. C 语言源程序文件经过编译连接之后生成一个后缀为_____的文件。
A. ".c" B. ".obj" C. ".exe" D. ".bas"
2. 以下选项中不正确的整型常量是_____。
A. 12L B. -10 C. 1,900 D. 123U
3. 表达式_____的值是0。
A. 3%5 B. 3/5.0 C. 3/5 D. 3<5
4. 表达式 $!(x>0||y>0)$ 等价于_____。
A. $!x>0||!y>0$ B. $!(x>0)||!(y>0)$ C. $!x>0&&!y>0$ D. $!(x>0)&&!(y>0)$
5. 若变量已正确定义并赋值, 表达式_____不符合C语言语法。
A. 4&&3 B. +a C. a=b=5 D. int(3.14)
6. 若变量已正确定义, 执行语句 `scanf("%d%d%d", &k1, &k2, &k3);` 时, _____是正确的输入。
A. 203040 B. 20 30 40 C. 20, 30 40 D. 20, 30,40
7. 执行语句 `printf("_____", 2);` 将得到出错信息。
A. %d B. %o C. %x D. %f
8. 设变量定义为 `char s[]="hello\nworld\n";`, 则数组s中有_____个元素。
A. 12 B. 13 C. 14 D. 15
9. 若x是整型变量, 表达式 $(x=10.0/4.0)$ 的值是_____。
A. 2.5 B. 2.0 C. 3 D. 2
10. 若变量已正确定义并赋值, 表达式 $-k++$ 等价于_____。
A. $-(k++)$ B. $(-k)++$ C. $-(k+1)$ D. $-(++k)$
11. 在C 语言程序中, 在函数内部定义的变量称为_____。
A. 全局变量 B. 外部变量 C. 局部变量 D. 内部变量
12. 设变量定义为 `int a[2]={1,3}, *p=&a[0]+1;`, 则*p的值是_____。
A. 2 B. 3 C. 4 D. &a[0]+1
13. 对于以下的变量定义, 表达式_____是正确的。

```
struct node {  
    char s[10];  
    int k;  
} p[4];
```

A. $p->k=2$ B. $p[0].s="abc"$ C. $p[0]->k=2$ D. $p->s='a'$
14. 下列运算符中, 优先级从高到低依次排列为_____。
A. !, &&, || B. ||, &&, ! C. &&, ||, ! D. &&, !, ||

15. 下列程序段的输出结果是_____。

```
int *p, *q, k = 1, j=10;  
p=&j; q = &k ; p = q ; (*p)++;  
printf("%d", *q);
```

- A. 1 B. 2 C. 10 D. 11

二、填空题（每题 2 分，共 20 分）

21. 设 ch 是字符型变量，判断 ch 为英文字母的表达式是_____。

22. 计算下列分段函数的 if 语句是_____。

$$y = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

23. 表达式 $!!5$ 的值是_____。

24. 下列程序段的输出结果是_____。

```
printf("%x\n", (0x19 << 1) & 0x17);
```

25. 下列程序段的输出结果是_____。

```
int k, s;  
for(k=0, s=0; k<10; k++)  
    if (s>k) break;  
    else s += k;  
printf("k=%d s=%d", k, s);
```

26. 下列程序段的输出结果是_____。

```
#define MIN(x,y) (x)<(y)?(x):(y)  
printf("%d", 10*MIN(10,15));
```

27. 下列程序在运行时可能会出错，原因是_____。

```
# include <stdio.h>  
void main()  
{  char *s;  
    gets(s);  
    puts(s);  
}
```

28. 下列程序的输出结果是_____。

```
# include <stdio.h>  
int f(int m)  
{  static int k=0;  
    int s=0;  
    for(; k<=m; k++) s++;  
    return s;  
}  
void main()  
{  int s=1;  
    s=f(5);  
    printf("%d %d\n", s, f(3));  
}
```

29. 下列程序段的输出结果是_____。

```
char *st[ ]={"abcd","efgh","ijkl","mnop"};
printf("%s", *(st+1)+1);
```

30. 用 `typedef` 定义一个 10 个元素的整型数组类型 ARR，再用 ARR 定义一个 10 个元素的整型数组 a。

三、程序阅读题（每题5分，共15分）

5. 输入 123+234=357Yes?<回车>后，下列程序的输出结果是_____。

```
# include <stdio.h>
void main()
{ int k, n1, n3[10];
  char c;
  n1=0;
  for(k=0; k<10; k++)
    n3[k]=0;
  while((c=getchar( ))!="\n"){
    switch(c){
      case'0': case '1': case '2': case '3': case '4':
      case'5': case '6': case '7': case '8': case '9':
        n3[c-'0']++;
        break;
      default:
        n1++;
        break;
    }
  }
  if(n1) printf("%d ", n1);
  for(k=0; k<10; k++)
    if(n3[k]) printf("%d ", n3[k]);
}
```

2. 输入 3<回车>后，下列程序的输出结果是_____。

```
#include "stdio.h"
void main()
{ int j,k,n,temp, a[6][6];
  scanf("%d",&n);
  for(k=0;k<n;k++)
    for(j=0;j<n;j++)
      a[k][j]=j+k*n;
  for (k=0;k<n;k++)
    for(j=0;j<n/2;j++){
      temp=a[n-j-1][k];
      a[n-j-1][k]=a[k][j];
      a[k][j]=temp;
```

```
    }
    for(k=0;k<n;k++)
    { for(j=0;j<n;j++)
        printf("%d ",a[k][j]);
        printf("#");
    }
}
```

3. 下列程序的输出结果是_____。

```
# include <stdio.h>
#define LEN sizeof(struct line)
struct line
{ int num ;
    struct line *next;
};
void main( )
{ int k ;
    struct line *p , *head , *tail;
    void f(struct line *p);
    head=tail=NULL;
    for(k=1; k<6; k++){
        p=(struct line *) malloc (LEN) ;
        p->num=k;
        p->next=NULL;
        if(head==NULL)      head=p;
        else          tail->next=p;
        tail=p;
    }
    f(head);
}void f(struct line *p)
{
    if(p!=NULL){
        f(p->next);
        if(p!=NULL)  printf("%d ", p->num) ;
    }
}
```

四、程序(段)填空题 (每空2分, 共20分)

1. 输入 1 个正整数 n, 计算并输出 s 的前 n 项的和。

$$S = 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + \dots$$

```
#include <stdio.h>
void main( )
{ int k, flag, n;
    float s;
```

```
flag=1; s=0;
scanf("%d", &n);
for (k=1 ; k<=n; k++){
    s=s+ (1);
    (2);
}
printf("sum=%f\n", s);
}
```

2. 输入以 -1 结束的一批整数，输出其中的最大值。

例如：输入 -3 -8 -10 -1 后，输出：max=-3。

```
#include <stdio.h>
void main( )
{   int max, x;
    scanf("%d", &x);
    if(x!= -1){
        (3);
        while((4)){
            if(max < x) max=x;
            (5);
        }
        printf("max=%d\n", max);
    }
}
```

3. 设变量 x, y, z 已正确定义和赋值，请仔细阅读程序段 A，然后在程序段 B 中填入正确的内容，使程序段 A 和程序段 B 的输出结果一样（每空只能写一个赋值表达式）。

程序段 A

```
x=x+y;
y=x-y;
x=x-y;
z=x+y;
printf("%d,%d,%d",x,y,z);
```

程序段 B

```
(6);
(7);
y=z;
z=x+y;
printf("%d,%d,%d",x,y,z);
```

4. 请仔细阅读函数 f1，然后在函数 f2 中填入正确的内容，使函数 f1 和函数 f2 有相同的功能。

```
char *f1(char s[ ], char t[ ])
{   int k;
    for(k=0; t[k]!='\0'; k++)
        s[k]=t[k];
    s[k]='\0';
    return s;
}
```

```
char *f2(char *s, char *t)
{   (8);
    while(((9) = *t++) != '\0')
        ;
    return (10);
}
```

五、编程题（共 15 分，竺可桢学院混合班学生做第 3、4 题，其他学生做第 1、2 题）

1. 设 a 是一个整型数组， n 和 x 都是整数，数组 a 中各元素的值互异。请编写函数 $\text{find}(a, n, x)$ ，在数组 a 的前 n 个元素中查找 x ，如果找到，返回 x 在数组 a 中的位置；如果没有找到，返回 0。

例如：数组 a 中有元素 {1, 4, 3, 2, 5}，调用 $\text{find}(a, 5, 2)$ 后返回 4，因为在 a 的前 5 个元素中找到了 2，它在 a 中的位置是 4；而调用 $\text{find}(a, 5, 10)$ 后返回 0，因为在 a 的前 5 个元素中没有找到 10。

2. 文本文件 input.txt 中存放着一批互异的整数(数量不多于 100)。请编写一个程序，当运行该程序时，首先从键盘输入一个整数 x ，如果文本文件 input.txt 中包含了 x ，则输出 x 在文件中的位置，否则，输出"Not found" (必须调用第1题定义的函数 find 查找 x)。

例如：如果文件 input.txt 中存放了 1, 4, 3, 2, 5，运行程序，并输入 2 后，输出 4。

3. 设 a 是一个整型数组， n 和 x 都是整数，数组 a 中各元素的值互异，并按升序排列。请编写函数 $\text{find}(a, n, x)$ ，采用二分法在数组 a 的前 n 个元素中查找 x ，如果找到，返回 x 在数组 a 中的位置；如果没有找到，返回 0。

例如：数组 a 中有元素 {1, 3, 5, 7, 8}，调用 $\text{find}(a, 5, 3)$ 后返回 2，因为在 a 的前 5 个元素中找到了 3，它在 a 中的位置是 2；而调用 $\text{find}(a, 5, 10)$ 后返回 0，因为在 a 的前 5 个元素中没有找到 10。

4. 文本文件 filename 中存放着一批已按升序排列的互异的整数(数量不多于 100)。请编写一个带命令行参数的程序 prog.c ，当运行 $\text{prog filename number}$ 时，如果文本文件 filename 中包含了数 number ，输出该数在文件中的位置，否则，输出"Not found"，其它的运行方式显示出错信息。(程序中可以调用函数 atoi ，必须调用第3题定义的函数 find 查找 number)。

浙江大学“程序设计基础-C”参考答案

2001-2002 学年秋季学期(2002 年 1 月 20 日)

一、单项选择题（每题 2 分，共 30 分）

1 C
6 B
11 C

2 C
7 D
12 B

3 C
8 B
13 A

4 D
9 D
14 A

5 D
10 A
15 B

二、填空题（每题 2 分，共 20 分）

1 ch>='A'&&ch<='Z' ||ch>='a'&&ch<='z'
或 isalpha(ch)

3 1

5 k=4 s=6 (错 1 个数字扣 1 分)

7 指针 s 没有赋值

9 fgh (错 1 个字母扣 1 分)

2 if(x>0) y=1;
else if(x==0) y=0;
else y=-1;

4 12

6 15

8 6 0 (错 1 个数字扣 1 分)

10 typedef int ARR[10];
ARR a;

三、程序阅读题（每题 5 分，共 15 分）

1 6 1 2 3 1 1 1 错 1 个数字扣 1 分

3 6 1 2 # 7 4 5 # 8 3 0 # 错 1 个数字扣 1 分

3 5 4 3 2 1 错 1 个数字扣 1 分

四、程序填空题（每空 2 分，共 20 分）

- | | |
|---------------------|----------------|
| (1) 1.0/k*flag | (2) flag=-flag |
| (3) max=x | (4) x!=-1 |
| (5) scanf("%d", &x) | (6) z=x |
| (7) x=y | (8) char *ss=s |
| (9) *s++ | (10) ss |

五、编程题（共 15 分，竺可桢学院混合班学生做第 3、4 题，其他学生做第 1、2 题）

1、

```
int find(int *a, int n, int x)      /* 说明1 分 */
{
    int k=0;
    for(k=0; k<n; k++)            /* 循环、查找2 分 */
        if (a[k]==x)  return k+1;   /* 找到返回位置 1 分 */
    return 0;                      /* 没有找到返回0 1 分 */
}
```

2、

```
# include <stdio.h>          /*头文件、变量说明、打开关闭文件、输入x共 2
分 */
void main()
{
    FILE *fp;
    int k=0, res, x;
    int a[100];
    if (( fp = fopen("input.txt","r" ) ) == NULL){
        printf( "Can't open file input.txt\n" );
        exit(0);
    }
    scanf("%d", &x);
    while ( !feof( fp ) ){           /*判断 2 分*/
        fscanf(fp,"%d",&a[k]);      /* 读入数组3 分*/
        k++;
    }
    res=find(a, k, x);             /* 调用 2 分*/
    if(res)  printf("%d", res);     /* 条件判断 1 分*/
    else printf("not found");
    fclose( fp );
}
```

浙江大学“程序设计基础-C”试题卷

2001-2002 学年春季学期

考试时间: 2002 年 6 月 30 日下午 2:00-4:00

注意: 答题内容必须写在答题卷上, 写在本试题卷上无效

一、单项选择题 (每题 2 分, 共 20 分)

16. C 语言程序的三种基本结构是顺序结构、选择结构和_____结构。
A. 递归 B. 转移 C. 循环 D. 嵌套
17. 下列运算符中, 优先级最高的是_____。
A. -> B. ++ C. && D. =
18. 表达式 $!(x>0 \&\& y>0)$ 等价于_____。
A. $!(x>0) || !(y>0)$ B. $!x>0 || !y>0$ C. $!x>0 \&\& !y>0$ D. $!(x>0) \&\& !(y>0)$
19. 表达式 `sizeof("\nsum=%d\n")` 的值是_____。
A. 8 B. 9 C. 10 D. 11
20. 以下选项中不正确的实型常量是_____。
A. 0.23E B. 2.3e-1 C. 1E3.2 D. 2.3e0
21. 表达式_____的值不是1。
A. 0?0:1 B. 5%4 C. !EOF D. !NULL
22. 数组定义为 `int a[11][11]`, 则数组a 有_____个数组元素。
A. 12 B. 144 C. 100 D. 121
23. 对于以下的变量定义, 表达式_____是不正确的。
`struct node {
 int j, k;
} x, *p=&x;`
A. `p->k=2` B. `(*p).k=2` C. `x.k=2` D. `x->k=2`
24. 不正确的赋值或赋初值的方式是_____。
A. `char str[]="string";` B. `char str[10]; str="string";`
C. `char *p="string";` D. `char *p; p="string";`
25. 设程序中不再定义新的变量, 写出在函数main中可以使用的所有变量_____。

```
#include <stdio.h>  
int z;  
int p(int x)  
{    static int y;  
    return(x+y);  
}  
void main()  
{    int a,b;  
    printf("%d\n", p(a));  
    .....  
}
```


A. a, b B. a,b,z C. a,b,y,z D. a,b,x,y,z

二、填空题 (每题 2 分, 共 30 分)

31. 设字符型变量 ch 的值是小写英文字母, 将它转换为相应大写字母的表达式是_____。

32. 表达式 $1 < 0 < 5$ 的值是_____。
33. 表达式 $\sim(10 >> 1 \wedge \sim 5)$ 的值是_____。
34. 下列程序段的输出结果是_____。

```
int a=1;
long b=a*0x12345678;
printf("%x\n", b);
```

35. 设变量已正确定义，在程序段 B 中填入正确的内容，使程序段 A 和程序段 B 等价。

程序段 A

```
for(k=s=0; k<=10; s+=a[k++])
    if (a[k]<0) break;
```

程序段 B

```
for(k=0, s=0; ; s+=a[k++])
    if (_____) break;
```

36. 下列程序段的输出结果是_____。

```
#define T 10
#define MD 3*T
printf("%d", 30/MD);
```

37. 下列程序段的输出结果是_____。

```
int k, s;
for(k=s=0; k<10&&s<=10; s+=k)
    k++;
printf("k=%d, s=%d", k, s);
```

38. 下列程序的输出结果是_____。

```
#include <stdio.h>
int f(int m)
{
    static int k=2;
    k++;
    return k+m;
}
void main()
{
    int s;
    s=f(2);
    printf("%d, %d\n", s, f(s));
}
```

39. 对于以下递归函数 f，调用 f(3) 的返回值是_____。

```
f (int n)
{
    return ((n<=0) ? n: f(n-1)+f(n-2));
}
```

40. 下列程序段的输出结果是_____。

```
int c[ ]={10, 0, -10};
int *k=c+2;
printf("%d", *k--);
```

41. 下列程序段的输出结果是_____。

```
int k=1, j=2, *p=&k, *q=p;
```

```
p=&j;  
printf("%d, %d", *p, *q);
```

42. 下列程序的输出结果是_____。

```
# include <stdio.h>  
char * f(char *str)  
{    return str="fun";  
}  
void main( )  
{    char *str="one";  
    printf("%s, %c\n", f(str)+1, *f(str)+1);  
}
```

43. 下列程序的输出结果是_____。

```
#include <stdio.h>  
void p(int *x,int y)  
{    ++ *x;  
    y --;  
}  
void main()  
{    int x=0, y=3;  
    p(&x, y);  
    printf("%d, %d",x, y);  
}
```

44. 下列程序的输出结果是_____。

```
# include "stdio.h"  
void main()  
{    int j, k, a[2][2];  
    for(k=1; k<3; k++)  
        for(j=1; j<3; j++)  
            a[k-1][j-1]=(k/j)*(j/k);  
    for(k=0; k<2; k++)  
        for(j=0; j<2; j++)  
            printf("%d,", a[k][j]);  
}
```

45. 用 `typedef` 定义一个指向函数的指针类型 FP，该函数的返回值为整型。

三、程序阅读题（每题5分，共15分）

6. 输入 `c2470f ?<回车>` 后，下列程序的输出结果是_____，该程序的主要功能是_____。

```
# include "stdio.h"  
int main( )  
{    char ch;  
    long number=0;  
    while((ch=getchar())<'0'||ch>'6')  
        ;  
    while(ch!='?'&&ch>='0'&&ch<='6')  
        number=number*7+ch-'0';
```

```
    printf("%ld#",number);
    ch=getchar();
}
}
```

2. 输入 4<回车>后，下列程序的输出结果是_____。

```
# include "stdio.h"
int main( )
{ int j,k,n;
  float f,s;

  scanf("%d", &n);
  s=0; f=1;
  for(k=1;k<=n;k++){
    for(j=1;j<k;j++)
      f=f*k;
    s=s+f;
    printf("%.0f#",s);
  }
}
```

3. 输入 55 92 63 69 -1<回车>后，下列程序的输出结果是_____。

```
#include <stdio.h>
#include <alloc.h>
struct link {
  int mark;
  struct link * next;
};

void f(struct link **);
main( )
{
  struct link * head, *p;
  head=(struct link *)malloc(sizeof(struct link));
  head->mark = 0;
  head->next=NULL;
  f(&head);
  for (p=head; p!=NULL; p=p->next)
    printf("%d#", p->mark);
}

void f(struct link ** head)
{
  int mark;
  struct link *p;
  scanf("%d", &mark);
  if ( mark<=0){
    (*head)->mark++;
    return ;
  }
}
```

```
    else{
        p=(struct link *)malloc(sizeof(struct link));
        p->mark = mark;
        p->next = *head;
        *head = p;
        f(head);
    }
}
```

四、程序填空题（每空2分，共20分）

1. 输入 10 个浮点数，调用函数 $f(x)$ 计算并输出相应的函数值。

$f(x)$ 的功能是计算分段函数：

$$y = \begin{cases} x^2 + 2x + 1 & x < 0 \\ 0 & x = 0 \\ 3 & x > 0 \end{cases}$$

```
#include <stdio.h>

(1)
{
    if (x<0) _____(2)_____;
    else _____(3)_____;
    else return 3;
}

void main()
{
    int k;
    float x;
    for(k=1; k<=10; k++){
        scanf("%f", &x);
        printf("y=%f\n", f(x));
    }
}
```

2. 函数 $sort$ 使用冒泡法将一维整型数组中各元素按值从大到小排序。

```
void sort( int a[ ], int n)
{
    int k, j, t;
    for(k=1; k<n; k++)
        for(_____(4)_____; j < n-k; j++)
            if(_____(5)_____) _____(6)_____
}
```

3. 输入 1 个长整数，求各位数字的平方和。

例如：输入 -123，输出 14。

```
#include <stdio.h>
void main()
{
    int digit;
    long in,s;

    scanf("%ld", &in);
```

```
_____(7)_____  
_____(8)_____  
while(in>0){  
    _____(9)_____;  
    s=s+digit*digit;  
    _____(10)_____;  
}  
printf("sum=%ld\n", s);  
}
```

五、编程题（共 15 分）

1. 函数 `ff(str,ch)` 的功能是统计并返回字符 `ch` 在字符串 `str` 中出现的次数。

例如：调用 `ff("every", 'e')` 后返回 2，因为在字符串 "every" 中字符 'e' 出现了 2 次。

2. 从键盘输入一个字符 `ch`，输出该字符在文本文件 `input.txt` 的每一行中出现的次数。（必须调用第 1 题定义的函数 `ff` 统计 `ch` 的出现次数）。

例如：如果文件 `input.txt` 中存放了下列数据：

`every`

`121`

`I am a student.`

运行程序，并输入'e'后，输出

`2`

`0`

`1`

浙江大学“程序设计基础-C”参考答案

2001-2002 学年春季学期(2002 年 6 月 30 日)

一、单项选择题（每题 2 分，共 20 分）

1 C
6 C

2 A
7 D

3 A
8 D

4 B
9 B

5 AC
10 B

二、填空题（每题 2 分，共 30 分）

1	ch-'a'+'A'或 toupper(ch)
3	0
5	a[k]<0 k>10 (错 1 项扣 1 分)
7	k=5, s=15 (错 1 项扣 1 分)
9	-2
11	2, 1 (错 1 个数字扣 1 分)
13	1, 3 (错 1 个数字扣 1 分)
15	typedef int (*FP)(); (错 1 项扣 1 分)

2	1
4	5678 或 12345678
6	100
8	5, 9 (错 1 个数字扣 1 分)
10	-10
12	un, g (un1 分, g1 分)
14	1,0,0,1 (错 1 个数字扣 1 分)

三、程序阅读题（每题 5 分，共 15 分）

1	2#18# 七进制转换十进制	错 1 项扣 2 分
3	1#3# 21#1173#	错 1 项扣 2 分
3	70#63#92#55#0#	错 1 项扣 1 分

四、程序填空题（每空 2 分，共 20 分）

(1)	float f(float x)	(2)	return x*x+2*x+1
(3)	if(x==0) return 0	(4)	j=0
(5)	a[j]<a[j+1]	(6)	{ t = a[j+1]; a[j+1] = a[j]; a[j] = t; }
(7)	if (in<0) in=-in	(8)	s=0 (7)、(8)可以互换
(9)	digit=in%10	(10)	in=in/10

五、编程题（共 15 分）

1、int ff(char *str, char ch) /* 说明1 分 */

```
{ int count=0;
    while(*str){ /* 循环3分 条件、查找、str递增各1 分 */
        if(*str==ch) count++;
        str++;
    }
    return count; /* 返回 1 分 */
}
```

2、#include <stdio.h> /*头文件、变量说明、打开关闭文件、输入ch 共 2 */

```
#include <stdlib.h>
void main()
{ char ch,c,s[80];
    int k;
    FILE *fp;
    if((fp=fopen("input.txt","r"))==NULL) {
        printf("error\n");
        exit(0);
    }
    ch=getchar();

    k=0;
    while(1){ /*循环+break 1 分*/
        c=fgetc(fp); /* 读入 1 分*/
        if(c=='\n'||c==EOF){ /* 共 6 分 */
            s[k]='\0';
            k=0;
            printf("%d\n",ff(s,ch));
            if(c==EOF) break;
        } else s[k++]=c;
    }

    fclose(fp);
}
```

或

```
k=0;
while(!feof(fp)){ /* 循环判断 1 分 */
    c=fgetc(fp); /* 读入 1 分 */
    if(c=='\n') /* 共5分 */
        s[k++]=c;
    else{
        s[k]='\0';
        k=0;
        printf("%d\n",ff(s,ch));
    }
}
if(c!='\n'){ /* 处理最后1行 1 分*/
    s[k]='\0';
    printf("%d\n", ff(s,ch));
}
```

浙江大学“C 程序设计及实验”试题卷

2002-2003 学年春季学期

考试时间: 2003 年 6 月 20 日上午 8:30-10:30

注意: 答题内容必须写在答题卷上, 写在本试题卷上无效

一、单项选择题（每题 1 分，共 10 分）

26. 下列运算符中，优先级最低的是_____。
A. << B. != C. && D. =
27. 以下选项中不正确的整型常量是_____。
A. 0xa B. 010L C. 10 D. 1.0
28. 表达式 !(x>0||y>0) 等价于_____。
A. !(x>0)&&!(y>0) B. !x>0&&!y>0 C. !x>0||!y>0 D. !(x>0)||!(y>0)
29. 表达式_____的值不是0。
A. 1/2 B. !'\0' C. !EOF D. NULL
30. 若 x 是浮点型变量，表达式 (x=10/4) 的值是_____。
A. 2.5 B. 2.0 C. 3 D. 2
31. 若变量已正确定义并赋值，表达式_____不正确。
A. a&b B. a^b C. &&x D. a, b
32. 对于变量定义 int a[10], *p=a;, 表达式_____是不正确的。
A. a[9] B. p[5] C. *p++ D. a++
33. 不正确的赋值或赋初值的方式是_____。
A. char str[10]; str="string"; B. char str[]="string";
C. char *p="string"; D. char *p; p="string";
34. 设程序中不再定义和声明变量，写出在函数fun中可以使用的所有变量_____。
- ```
#include <stdio.h>
void fun(int x)
{ static int y;

 return;
}
int z;
void main()
{ int a,b;
 fun(a);

}
```
- A. x, y      B. x, y, z      C. a,b,y,z      D. a,b,x,y,z
35. 对于变量定义 int p[5], \*a[5];, 表达式\_\_\_\_\_是正确的。  
A. p=a      B. p[0]=a      C. \*(a+1)=p      D. a[0]=2

### 二、填空题（每题 2 分，共 30 分）

46. 数组定义为 int a[2][3][4], 则数组a 中有\_\_\_\_\_个数组元素。  
47. 计算下列分段函数的条件表达式是\_\_\_\_\_。

$$y = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

48. 表达式  $1 < 10 < 5$  的值是\_\_\_\_\_。
49. 表达式  $\sim(10 \ll 1) \& 4$  的值是\_\_\_\_\_。
50. 表达式 `sizeof("hello")` 的值是\_\_\_\_\_。
51. 下列程序段的输出结果是\_\_\_\_\_。

```
int k, s;
for(k=1, s=0; k<10; k++){
 if (k%2==0) continue;
 s += k;
}
printf("k=%d s=%d", k, s);
```

52. 下列程序段的输出结果是\_\_\_\_\_。
- ```
#define MM(x,y) (x*y)  
printf("%d", MM(2+3,15));
```
53. 下列程序段的输出结果是_____。
- ```
int k=1, s=0;
switch (k) {
 case 1: s+=10;
 case 2: s+=20; break;
 default: s+=3;
}
printf("k=%d s=%d", k, s);
```

54. 下列程序的输出结果是\_\_\_\_\_。
- ```
# include <stdio.h>  
int f()  
{    static int k;  
    return ++k;  
}  
void main()  
{    int k;  
    for(k=0;k<3;k++)  
        printf("%d#", f());  
}
```

55. 下列程序的输出结果是_____。
- ```
f (int x)
{
 if(x<=1) return 1;
 else return f(x-1)+f(x-2);
}
void main()
{ printf("%d", f(4));
}
```

56. 下列程序段的输出结果是\_\_\_\_\_。

```
int k=1, j=2, *p, *q, *t;
p=&k; q=&j;
t=p; p=q; q=t;
printf("%d, %d", *p, k);
```

57. 下列程序段的输出结果是\_\_\_\_\_。

```
int c[]={10, 30, 5};
int *pc;
for(pc=c; pc<c+2; pc++)
 printf("%d#", *pc);
```

58. 下列程序段的输出结果是\_\_\_\_\_。

```
char *st[]={"ONE", "TWO", "FOUR", "K"};
printf("%s, %c\n", *(st+2), **st+1);
```

59. 下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
void p(int *x,int y)
{ ++ *x;
 y=y+2;
}
void main()
{ int x=0, y=3;
 p(&y, y);
 printf("%d, %d", x, y);
}
```

60. 用 `typedef` 定义一个 10 个元素的字符指针数组类型 CP。

### 三、程序阅读题（每题5分，共30分）

1. 下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
void main()
{
 int k, x, s, t;

 x=3;
 s=0; t=x;
 for(k=1; k<=3; k++){
 t=t*10+x;
 s=s+t;
 printf("%d#", s);
 }
}
```

2. 输入 7 3 0 3 0 3 1 2 9 7 6 0 <回车>后，下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
void main()
{
```

```
int j, k, sub, x;
int a[5];

for(j=1; j<=2; j++){
 for(k=0; k<5; k++)
 scanf("%d", &a[k]);
 scanf("%d", &x);
 sub=-1;
 for(k=0; k<5; k++)
 if(a[k]==x) sub=k;
 printf("%d#", sub);
}
}
```

3. 下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
void main()
{
 long number, wt, x;

 x=number=10230;
 wt=1;
 while(x!=0){
 wt=wt*10;
 x=x/10;
 }
 wt=wt/10;
 while(number!=0){
 printf("%d#", number/wt);
 number=number%wt;
 wt=wt/10;
 }
}
```

4. 输入 3 9 8 6 5 3 7 1 2 4 2 1 2 3 4<回车>后，下列程序的输出结果是\_\_\_\_\_。

```
#include "stdio.h"
void main()
{
 int flag,i,j,k,col,n,ri,a[6][6];
 for(ri=1; ri<=2; ri++){
 scanf("%d",&n);
 for(i=0; i<n; i++)
 for(j=0; j<n; j++)
 scanf("%d",&a[i][j]);
 flag=0;
 for(i=0; i<n; i++){
```

```
col=0;
for(j=0; j<n; j++)
 if (a[i][col]<a[i][j]) col=j;
for (k=0; k<n; k++)
 if(a[i][col]<a[k][col]) break;
if(k>=n){
 printf("a[%d][%d]=%d#", i, col,a[i][col]);
 flag=1;
}
}
if(!flag) printf("NO#");
}
```

5. 输入 how are you?<回车>后，下列程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
void main()
{
 int word;
 char ch;

 word=0;
 while((ch=getchar())!='?'){
 if(ch==' ') word=0;
 else if(word==0){
 word=1;
 if(ch<='z'&&ch>='a')
 ch=ch-'a'+'A';
 }
 putchar(ch);
 }
}
```

6. 假设文本文件 a.txt 中存放了下列数据：

one?two?1234?output?

文本文件 b.txt 中存放了下列数据：

one?two?1204?input?

下列程序的输出结果是\_\_\_\_\_。

```
include <stdio.h>
include <stdlib.h>
void main()
{ int count;
 char ch1,ch2;
 FILE *f1, *f2;

 if ((f1 = fopen("a.txt","r")) == NULL){
```

```
 printf("Can't open file : %s\n", "a.txt");
 exit(0);
}
if ((f2 = fopen("b.txt", "r")) == NULL){
 printf("Can't open file : %s\n", "b.txt");
 exit(0);
}
count=0;
while (!feof(f1)||!feof(f2)){
 ch1=fgetc(f1);
 ch2=fgetc(f2);
 if(ch1!=ch2){
 printf("%c#%c#",ch1,ch2);
 printf("%d#", count);
 break;
 }
 if(ch1=='?') count++;
}
fclose(f1);
fclose(f2);
}
```

#### 四、程序(段)填空题 (每空2分, 共30分)

1. 输入 1 个正整数 n, 计算并输出 s 的前 n 项的和。

$$s = 1/1! + 1/2! + 1/3! + \dots + 1/n!$$

```
#include <stdio.h>
void main()
{
 int j, k, n;
 float f, s;
 scanf("%d", &n);
 (1) ;
 for (k=1 ; k<=n; k++){
 (2) ;
 for(j=1; (3); j++)
 (4) ;
 s=s+1.0/f;
 }
 printf("sum=%f\n", s);
}
```

2. 输入一个正整数 n, 如果它是素数, 输出"YES", 否则, 输出"NO" (素数就是只能被 1 和自身整除的正整数, 1 不是素数, 2 是素数)。

```
#include <stdio.h>
#include <math.h>
void main()
```

```
{
 int n;
 int prime(int m);

 scanf("%d", &n);
 if(__5__)
 printf("YES\n");
 else
 printf("NO\n");
}

int prime(int m)
{ int i,n;

 if(m==1) return 0;
 n=sqrt(m);
 for(i=2; i<=n; i++)
 if(__6__) return __7__ ;
 __8__ ;
}
```

3. 请仔细阅读函数 f1，然后在函数 f2 中填入正确的内容，使函数 f1 和函数 f2 有相同的功能。

```
int f1(char s[])
{ int k=0;
 while(s[k]!='\0')
 k++;
 return k;
}

int f2(char *s)
{ char *ss;
 __9__ ;
 while(*s++!=__10__)
 ;
 return __11__ ;
}
```

4. 输入 30 位学生的姓名和成绩，输出成绩最高的学生的姓名及其成绩。

```
#include <stdio.h>
struct student{
 char name[10];
 int score;
};
void main()
{
 int k;
 struct student s[30], *ptr;
 struct student *f(struct student *stud, int n);

 for(k=0; k<30; k++)
```

```
scanf("%s%d", s[k].name, &s[k].score);
(12);
printf("%s %d\n", ptr->name, ptr->score);
}
struct student *f(struct student *stud, int n)
{
 struct student * pmax, *p;
 (13) ;
 for(p=stud; p<stud+n; p++)
 if(p->score > pmax->score) (14);
 return (15);
}
```

## 浙江大学“C 程序设计及实验” 参考答案

2002-2003 学年春季学期(2003 年 6 月 20 日)

### 一、单项选择题（每题 1 分，共 10 分）

1 D  
6 C

2 D  
7 D

3 A  
8 A

4 B  
9 A

5 B  
10 C

### 二、填空题（每题 2 分，共 30 分）

1 24  
2 \_\_\_\_\_  
3 1  
4 \_\_\_\_\_  
5 6  
6 \_\_\_\_\_  
7 47  
8 \_\_\_\_\_  
9 1#2#3# (错 1 个数字扣 1 分)  
10 \_\_\_\_\_  
11 2, 1 (错 1 个数字扣 1 分)  
12 \_\_\_\_\_  
13 FOUR, P (错 1 个项扣 1 分)  
14 \_\_\_\_\_  
15 typedef char \* CP[10]; 错 1 项扣 1 分  
16 \_\_\_\_\_

2 x>0? 1 : (x==0) ? 0 : -1  
3 \_\_\_\_\_  
4 0  
5 \_\_\_\_\_  
6 k=10 s=25 (错 1 项扣 1 分)  
7 \_\_\_\_\_  
8 k=1 s=30 (错 1 项扣 1 分)  
9 \_\_\_\_\_  
10 5  
11 \_\_\_\_\_  
12 10#30# (错 1 个数字扣 1 分)  
13 \_\_\_\_\_  
14 0,4 (错 1 个数字扣 1 分)  
15 \_\_\_\_\_

### 三、程序阅读题（每题 5 分，共 30 分）

- 1 33#366#3699# 错 1 项扣 2 分
- 3 3#-1# 错 1 项扣 2 分
- 3 1#0#2#3# 错 1 项扣 2 分
- 4 a[0][0]=9#a[1][2]=7#a[1][1]=4 错 1 项扣 2 分
- 5 How Are You 错 1 项扣 2 分
- 6 3#0#2# 错 1 项扣 2 分

## 浙江大学“C 程序设计及实验”试题卷

2003-2004 学年春季学期

考试时间: 2004 年 6 月 25 日上午 8:30-10:30

**注意: 答题内容必须写在答题卷上, 写在本试题卷上无效**

### 一. 单项选择题 (每题 1 分, 共 10 分)

36. C 语言程序的三种基本结构是顺序结构、选择结构和\_\_\_\_\_结构。  
A. 递归      B. 转移      C. 循环      D. 嵌套
37. 下列运算符中, 优先级最高的是\_\_\_\_\_。  
A. +=      B. [ ]      C. ?:      D. &
38. 运算符\_\_\_\_\_不能做为单目运算符。  
A. /      B. &      C. \*      D. +
39. 以下选项中不正确的字符常量是\_\_\_\_\_。  
A. '1'      B. '\0'      C. '%d'      D. 10
40. 表达式 x&&1 等价于\_\_\_\_\_。  
A. x==0      B. x==1      C. x != 0      D. x != 1
41. 已知字符 'a' 的ASCII码为 97 , 执行下列语句的输出是\_\_\_\_\_。  
`printf ("%c%d", 'b', 'b'+1);`  
A. b99      B. 98c      C. 9899      D. bc
42. 在C 语言程序中, 在函数内部定义的变量称为\_\_\_\_\_。  
A. 全局变量      B. 外部变量      C. 局部变量      D. 内部变量
43. 执行语句 `printf("_____ ", "hello");` 将得到出错信息。

- A. %u      B. %s      C. %x      D. %f
44. 对于以下的变量定义，表达式\_\_\_\_\_是不正确的。
- ```
struct node {  
    float x, y;  
} point, *p=&point;
```
- A. p->x=2.0 B. (*p).y=3.0 C. point.x=2.0 D. *p->y=3.0
45. 对于变量定义：int **p[10], p 是一个_____。
A. 指针 B. 数组 C. 函数 D. 数组元素

二、填空题（每题 2 分，共 30 分）

61. 写出计算 $s(s-a)(s-b)(s-c)$ 的 C 语言表达式____。
62. 如果运算符 * 和 / 都是右结合的，则表达式 $7 * 2 / 3$ 的值是____。
63. 表达式 $!!10$ 的值是____。
64. 表达式 $\sim(\sim 1 \ll 1)$ 的值是____。
65. 下列程序段的输出结果是____。

```
#define MM(x,y) (x)+(y)  
printf("%d", MM(2*3,15)*2);
```

66. 对于变量定义：int a[10], *p=a+3; 数组 a 的最后一个元素是 p[6]。
67. 下列程序段的输出结果是____。

```
int k=1, j=2;  
int *p=&k, *q=&j;  
(*p)++;  
k=j;  
(*p)++;  
printf("%d, %d", k, j);
```

68. 下列程序段的输出结果是____。
char *st[]={"ONE","TWO","FOUR","K"};
printf("%s, %c\n", *(st+1)+1, *(*(st+2)+2)-1);

69. 下列程序的输出结果是____。

```
#include <stdio.h>  
void p(int *x,int *y)  
{ int *p;  
    p=x; x=y; y=p;  
}  
void main()  
{ int x=0, y=3;  
    p(&y, &x);  
    printf("%d, %d", x, y);  
}
```

70. 用 **typedef** 定义一个整型指针类型 PA。 **typedef int *PA**

三、程序阅读题（每题5分，共30分）

1. 输入 2 59 60 <回车>后，下列程序的输出结果是____。

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

```
{    int ri, repeat;
    int mark;
    scanf("%d", &repeat);
    for(ri=1; ri<=repeat; ri++)
        scanf("%d", &mark);
    if(mark >= 60)
        printf("Pass");
    else
        printf("Fail");
}
```

2. 输入 $2+5*3/2\$$ <回车>后，下列程序的输出结果是_____。

```
#include <stdio.h>
void main( )
{    int op1, op2, res;
    char operator;
    scanf("%d", &op1);
    operator = getchar();
    while(operator!='='){
        scanf("%d", &op2);
        switch(operator){
            case '+': res = op1+op2; break;
            case '-': res = op1-op2; break;
            case '*': res = op1*op2; break;
            case '/': res = op1/op2; break;
            default: res = 0;
        }
        op1 = res;
        printf("%d#", res);
        operator = getchar();
    }
}
```

3. 调用函数 $f(2004, 4, 10)$ 的返回值是_____。

```
int f(int x, int y, int z)
{
    int k, flag;
    int tab[2][13]={
        {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},
        {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}
    };
    flag=x%4==0 && x%100!=0 || x%400==0;
    for(k=1; k<y; k++)
        z += tab[flag][k];
    return z;
}
```

4. 调用函数 f(15)的输出结果是_____。

```
void f(int n)
{
    if(n<3)
        printf("%d", n);
    else{
        f(n/3);
        printf("%d", n%3);
    }
    return;
}
```

5. 输入 How are you?<回车>后，下列程序的输出结果是_____。

(函数 strcmp(s, t)的功能：若 s 和 t 相等,返回 0; 若 s 大于 t,返回一个正数; 若 s 小于 t,返回一个负数)

```
#include <stdio.h>
#include <string.h>
void main()
{
    int i;
    char s[80], ss[80];

    scanf("%s", s);
    strcpy(ss,s);
    for(i=1; i<3; i++){
        scanf("%s", s);
        if(strcmp(s, ss) > 0)
            strcpy(ss,s);
    }
    printf("%s\n", ss);
}
```

6. 下列程序的功能是_____。

```
#include <stdio.h>
# include <stdlib.h>
void main()
{
    FILE *fpa, *fpb;

    if((fpa=fopen("t.c","r")) == NULL){
        printf("can not open file a.txt !\n");
        exit(0);
    }
    if((fpb=fopen("b.txt", "w")) == NULL){
        printf("can not open file b.txt!\\n");
        exit(0);
    }
    while(!feof(fpa))
        fputc(fgetc(fpa), fpb);
```

```
fclose(fpa);
fclose(fpb);
}
```

四、程序(段)填空题（每空2分，共40分）

1. 读入 1 个正实数 eps, 计算并输出 $1 - 1/3 + 1/5 - 1/7 + \dots$, 直到最后一项的绝对值小于 eps。

```
#include <stdio.h>
#include <math.h>
void main( )
{
    int temp, flag;
    double eps, item, sum;

    scanf("%le", &eps);
    sum = 0; item = flag = temp = 1;
    while(fabs(item)>=eps){
        sum = sum + item;
        temp = (1);
        (2);
        item = (3)*flag/temp;
    }
    printf("%f\n", sum);
}
```

2. 输入一行字符，分别统计出其中的英文字母、空格、数字和其他字符的个数。

```
#include <stdio.h>
void main( )
{
    int blank=0, digit=0, letter=0, other=0;
    char c;

    c = getchar();
    while((4)){
        if((5)) /* 判断英文字母 */
            letter++;
        else if((6)) /* 判断数字字符 */
            digit++;
        else if((7)) /* 判断空格 */
            blank++;
        else
            other++;
    }
    printf("letter=%d, blank=%d, digit=%d, other=%d\n", letter, blank, digit, other);
}
```

3. a 是一个 3×3 的矩阵，输入 a 的元素，如果 a 是下三角矩阵，输出"YES"，否则，输出"NO"。(下三角矩阵，即主对角线以上的元素都为 0，主对角线为从矩阵的左上角至右下角的连线)

```
#include "stdio.h"
void main( )
{ int a[3][3], flag, i, k;

    for (i=0;i<3;i++)
        for (k=0;k<3;k++)
            scanf("%d",&a[i][k]);
    (9);
    for (i=0;i<3;i++)
        for ((10); k<3;k++)
            if((11)){
                (12);
                break;
            }
    if(flag) printf("YES\n");
    else   printf("NO\n");
}
```

4. 定义函数 f(number)，它的功能是返回 number 的逆序数。例如 f(-123)的返回值是-321。

```
long f(long number)
{ int flag, digit;
long res = 0;

flag = number<0 ? -1 : 1;
if(number<0)  number = - number;
while(number!=0){
    (13);
    number /= 10;
}
(14);
```

5. 编写一个函数 delchar(s, c)，该函数将字符串 s 中出现的所有 c 字符删除。

```
void delchar(char s[ ],char c)
{ int j=0, k=0;

    while(s[j] != '\0'){
        if(s[j] != c){
            s[k] = s[j];
            (15);
        }
        (16);
    }
```

____(17)____;
}

6. 请仔细阅读函数 f1, 然后在函数 f2 中填入正确的内容, 使函数 f1 和函数 f2 有相同的功能。

```
void f1(char s[ ], char t[ ])          void f2(char *s, char *t)  
{  int j, k;  {  
    j=k=0;    while(____(18)____)  
    while(s[j]!='\0')      ____(19)____++;  
        j++;    while(____(20)____)  
    while((s[j++]=t[k++])!='\0')      ;  
        ;  
}
```

浙江大学“C 程序设计及实验” 参考答案

2003-2004 学年春季学期(2004 年 6 月 25 日)

一、单项选择题（每题 1 分，共 10 分）

1 C
6 A

2 B
7 C

3 A
8 D

4 C
9 D

5 C
10 B

二、填空题（每题 2 分，共 20 分）

1 s*(s-a)*(s-b)*(s-c)

2 0

3 1

4 3

5 36

6 6

7 3, 2

(错 1 个数字扣 1 分)

8

WO,T

(错 1 项扣 1 分)

9 0,3

(错 1 个数字扣 1 分)

10

typedef int * PA;

错 1 项扣 1 分)

三、程序阅读题（每题 5 分，共 30 分）

1 Pass (Fail Pass 扣 3 分)

2 7#21#10#0 (错 1 项扣 2 分)

3 101

4

120

(错 1 个数字扣 2 分)

5 you?

少 ? 扣 1 分

6

文件复制 a.txt=>b.txt

四、程序填空题（每空 2 分，共 40 分）

(1) temp + 2

(2) flag = -flag

(3) 1.0 或 (float) 或 (double)

(4) c != '\n'

(5) c>='a' && c <= 'z' || c>='A' && c <= 'Z' (6) c>='0' && c <= '9'

(7) c==' ' (8) c=getchar()

(9) flag=1 (10) k=i+1

(11) a[i][k]!=0 (12) flag=0

(13) res = res*10 + number%10 (14) return flag*res

(15) k++ (16) j++

(17) s[k] = '\0' (18) *s != '\0'

(19) s (20) *s++ = *t++

2004~2005 学年“C 程序设计基础”试卷

考试时间: 2005 年 7 月 5 日上午 10:30-12:30

姓名_____ 学号_____ 成绩_____

一、单项选择题（每题 2 分，共 20 分）

46. 下面 C 语言的标识符中, 哪个不合法_____。
A. _total B. Large tall C. int_ D. _while
2. C 语言源程序经过编译连接最后生成的文件, 其后缀名是_____。
A. ".c" B. ".obj" C. ".exe" D. ".bas"
3. 表达式 $x \& \& 1$ 等价于_____。
A. $x != 0$ B. $x == 0$ C. $x != -1$ D. $x == -1$
4. 若变量 k 已正确定义且 k 的值是 4, 计算表达式 ($j = k++$) 后, _____。
A. $k=4, j=4$ B. $j=4, k=5$ C. $j=5, k=4$ D. $j=5, k=5$
5. C 语言中, 函数若没有声明返回值的类型, 默认的类型是_____。
A. int B. long C. float D. double
6. 若 p1 和 p2 都是指向 int 类型变量的指针, p1 已经指向变量 x, 要使 p2 也指向 x, 则_____是正确的。
A. $p2 = \&p1;$ B. $p2 = \&\&p1;$ C. $p2 = \&p1;$ D. $p2 = *p1;$
7. 下列程序段输出是_____。
int a[]={1,3,5,7,9};
int *b=a+2;
printf("%d",*b++);
A. 3 B. 5 C. 7 D. 9
8. 执行下列程序:

```
#define FB(a,b) (a*b+1)
int k=3;
k=FB(FB(k,k+1),k)-7;
后变量 k 的值是_____。
```


A. 7 B. 24 C. 27 D. 33
9. 在某 C 程序中调用了 puts 函数。在程序的编译预处理部分应用 #include 包含
_____。
A. "math.h" B. "string.h" C. "stdio.h" D. "ctype.h"
10. 设有结构定义如下:

```
struct student{
    int num;
    char name[30];
    float score;
}stud1;
```


则标准的 ANSI C(如 TC)下, 计算 sizeof(stud1) 的值为_____。
A. 9 B. 16 C. 25 D. 36

二、填空题（每题 3 分，共 24 分）

1. 写出语句 `printf("%d,%o,%x",'a'/4,0x14,014);` 的输出结果_____。
2. 写出计算 $\sqrt{s(s-a)(s-b)(s-c)}$ 的 C 语言表达式_____。

3. 写出下列程序段的输出结果是_____。

```
int m;char ss[10]={"12345"},*s=ss;
for(m=10;m!=0;m--);
    printf("%d",m);
while(*s++)
    putchar(*++s);
```

4. 下列程序段的输出结果是_____。

```
#include "stdio.h"
int x;
void f(int x)
{static int a=1;
printf("%d,",a);
a+=x+2;
}
void main()
{int k;
for(k=1;k<=3;k++)
f(k);
}
```

5. 对于以下递归函数 f, 调用 f(4) 的返回值是_____。

```
f(int n)
{return((n>0)?2*f(n-1)+f(n-2):1);}
```

6. 下列程序的输出结果是_____。

```
#include "stdio.h"
void main()
{int x=3,y=6,z=0;
while(x++)!=(y-=1))
{z++;
if(x<y)break;
}
printf("x=%d,y=%d,z=%d",x,y,z);
}
```

7. 下列程序的输出结果是_____。

```
int p,q,k=3,j=4;
q=&k;p=q;q=&j;
printf("%d", *p);
```

8. 下面函数定义中的错误是_____。

```
double sum(double x,y)
{return(x*x+y*y);
}
```

三、程序阅读题（每题5分，共20分）

7. 下列程序的输出结果是_____。

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

```
void main( )
{ int x,k;
x=12;
for(k=0;x!=0;x=x+2)
    ++k;
printf("k=%d,x=%d\n",k,x);
}
```

8. 下列程序的输出结果是_____。

```
# include <stdio.h>
void sort(int b[],int n,int x)
{ int k;
for(k=n-1;k>=0;k--)
    if(b[k]<x)
        b[k+1]=b[k];
    else
        break;
    b[k+1]=x;
}
void main()
{int i,j;
static int a[5]={8,5,7,2,6};
int b[8];
b[0]=a[0];
for(i=1;i<5;i++)
{sort(b,i,a[i]);
    for(j=0;j<=i;j++)printf("%d ",b[j]);
    printf("\n");
}
}
```

9. 输入12345后，下列程序的输出结果是_____。

```
#include "stdio.h"
int fun();
void main()
{ fun();}
int fun()
{ char c;
if((c=getchar())!='\n') fun();
putchar(c);
}
```

4. 下列程序的输出结果是_____。

```
# include <stdio.h>
void main()
{ int i,n;
char *pa,*s;
pa="Happy";
```

```
for(s=pa,n=0;8S1="\0';++s,++n);
for(s=pa,j=0;j<n;putchar(*s++),++i);
}
```

四、程序填空题（每空2分，共20分）

1. 读入1个正整数n，计算并输出s的前n项的和。

```
s=1-1/2+1/3-1/4+.....
# include <stdio.h>
# include <math.h>
void main()
{ int k,flag,n;
float s;
flag=1;s=0;
scanf("%d",&n);
for(k=1;k<=n;k++){
s=s+_____ (1) _____;
_____ (2) _____;
}
printf("sum=%f\n",s);
}
```

2. 在C盘一级目录TC(c:\tc)下创建文本文档k2.txt。用键盘键入一个字符串，以‘@’作为结束标记，并写入该文件，然后重新从中取出字符并显示在屏幕上。

```
# include <stdio.h>
# include <stdlib.h>
void main()
{ _____ (3) _____ ;
if((fp=fopen("_____ (4) _____ ","w"))==NULL){
printf("Can't open file\n");
exit(0);
}
while(_____ (5) _____ )
fputc(ch,fp);
fclose(fp);
if(_____ (6) _____ ){
printf("Can't open file\n");
exit(0);
}
while(!feof(fp)){
ch=fgetc(fp);
printf(_____ (7) _____ );
}
fclose(fp);
}
```

3. 下列程序用于实现从键盘上输入一个正整数，左移三位，并将移位后的数以二进制数输

出。

```
# include <stdio.h>
void main()
{ int m,n,i=0,j;
int bb[16];
scanf("%d",&n);
m=_____ (8) _____;
while(m>0)
{bb[i]=m%2;
 _____ (9) _____;
i++;
}
printf("binary number is:");
for (_____ (10) _____)
printf("%d",bb[i]);
}
```

五、编程题（16分）

(略)

2004~2005 学年“C 程序设计基础”试卷参考答案

考试时间: 2005 年 7 月 5 日上午 10:30-12:30

一、单项选择题（每题 2 分，共 20 分）

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

二、填空题（每题 3 分，共 24 分）

1. 24,24,c
2. $\sqrt{s^*(s-a)^*(s-b)^*(s-c)}$
3. 0.35
4. 1,4,8
5. 41
6. $x=4,y=5,z=1$
7. 3
8. 对形参 y 必须作类型说明

三、程序阅读题（每题 5 分，共 20 分）

1. $k=6,x=0$
2. 85
875
8752
87652
3. \n
54321
4. Happy

四、程序(段)填空题（每空 2 分，共 20 分）

- (1) $flag*1./k$
- (2) $flag=-flag$
- (3) FILE *fp;
- (4) c:\\tc\\k2.txt
- (5) (ch=getchar()!='@')
- (6) fp(fopen("c:\\tc\\k2.txt","r"))==NULL
- (7) "%c",ch
- (8) n<<3
- (9) m/=2
- (10) j=i-1;j>=0;j--

浙江大学 2005-2006 学年冬季学期

《C Programming》课程期末考试试卷

开课学院：计算机学院， 考试形式：闭卷

考试时间：2006 年 1 月 11 日，所需时间：120 分钟，任课教师_____

考生姓名：_____ 学号：_____ 专业：_____

(注意：答题内容必须写在答题卷上，写在本试题卷上无效)

Section 1: Single Choice (2 mark for each item, total 20 marks)

- 1、If x is a float variable, the value of expression $(x = 7/2, x+1)$ is _____.
A. 3 B. 3.5 C. 4 D. 4.5
- 2、The value of expression `sizeof("\total=%lf\\n")` is _____.
A. 13 B. 14 C. 15 D. 16
- 3、The statement _____ is correct if a is an integer variable and b is a double precision float variable .
A. `scanf("%d,%f", &a, &b);` B. `scanf("a=%d b=%*f", &a, &b);`
C. `scanf("%d%*f", &a, &b);` D. `scanf("%d%*f", &a);`
- 4、The value of expression `strcmp("internet", "intranet")` is _____.
A. positive B. negative C. zero D. uncertain
- 5、The expression `!(x>0&&y<0)` is equivalent to _____.
A. $x \leq 0 \mid y \geq 0$ B. $!x > 0 \&\&!y < 0$ C. $!x > 0 \mid !y < 0$ D. $!(x > 0) \&\& !(y < 0)$
- 6、If variables are defined, the value of expression _____ is not 1.
A. 2^3 B. $1 << 1/2$ C. $(a=1,b=2,a>b)$ D. $!\backslash 0'$
- 7、If variables are defined and assigned correctly, the expression _____ is wrong.
A. `float(256)` B. `a>b>5` C. `a,2,b` D. `5&&6`
- 8、The following code fragment prints out _____.

```
int i=5;
while (i-->1){
    if(i%2) printf("*");
    else continue;
    printf("#");
}
printf("$\\n");
```


A. *#*#*#\$ B. #*#*#*\$ C. #*#*\$ D. *#*#\$
- 9、With regard to the following definition:

```
struct {
    int len;
    char *str;
}s[2]={{10,"hello"},{20,"world"}},*p=s;
```

operator `++` in the expression `*p++->str` is applied on _____.
A. s[1].str B. s[2].str C. s[1].len D. s[2].len

Section 2: Fill in the blanks (2 mark for each item, total 30 marks)

- 1、If ch is an uppercase character variable, the expression can be written as _____ according to the changing rule: 'A'->'Z', 'B'->'Y', 'C'->'X'.....

2、The value of expression `strlen("http:\\www.zju.edu.cn")` is _____.

3、The following code fragment prints out _____.
`printf("%o\n", (010<<1^1) &(0xe<<1));`

4、Write a conditional expression_____ to calculate the value of y.
$$y = \begin{cases} 1/x & x>0 \\ 0 & x=0 \\ -2/x & x<0 \end{cases}$$

5、The following code fragment prints out _____.
`#define FB(a,b) (a*b+1)
int k=3;
k=FB(FB(k+1,k-1),k)-7;
printf("%d",k);`

71. After running the following code fragment, the value of a is _____.
`int a=-1,x=10,y=20,ok1=5,ok2=0;
if(x<y)
 if(y!=10)
 if(!ok1)
 a=1;
 else
 if(ok2) a=10;`

72. The following code fragment prints out _____.
`int i,j,k=0;
for (i=0, j=10; i<=j; i++, j--)
 k+=i+j;
printf ("%d,%d,%d", i,j,k);`

73. The following code fragment prints out _____.
`int k=2, a=1;
while(k--!=1) {
 do {
 a*=k;
 a++;
 } while (k--);
}
printf("%d,%d",k,a);`

74. The following code fragment prints out _____.
`int a=1, b=2, c=3;
if (a>b) a=b;
if (b>c) b=c;
if (c>a) c=a;
printf ("%d,%d,%d", a,b,c);`

```
int x=1,y=0,a=0,b=0;
switch(x) {
    case 1:
        switch(y) {
            case 0: a++; break;
            case 1: b++; break;
        }
    case 2:
        a++; b++; break;
}
printf("a=%d,b=%d",a,b);
```

75. Writing the declaration with `typedef`, which makes NAME a synonym for a character pointer array, 20 elements ____.

76. With the code fragment below, function call `f(3)`, the return value is ____.

```
f (int n)
{   return ((n>0) ? f(n-1)+2*f(n-2) :1); }
```

77. The output of the following program is ____.

```
# include <stdio.h>
int f()
{   static int k=0;
    k++;
    return k;
}
void main()
{   int k;
    for(k=0;k<2;k++)  f();
    printf("%d", f());
}
```

78. The output of the following program is ____.

```
#include <stdio.h>
main()
{
    int a=1,b=2,m;
    int s(int,int *);
    m=s(a,&b);
    printf("%d#%d#%d#",m,a,b);
}
int s(int a,int *b)
{
    int m;
    m = a+*b;
    *b -= a;
    return m;
}
```

79. In the following command-line program run.c, the output after running “run –abc –nlsworld” is _____.

```
#include <stdio.h>
main(int argc, char *argv[])
{
    printf("%c#%c", *++argv[0], (*++argv)[0]);
}
```

80. After executing the following block, the value of variable i is _____.

```
static int a[3][4]={{1,2,3},{4,5,6}}, i;
i= a[2][-2] + a[1][3];
```

Section 3: Read each of the following programs and answer questions (5 marks for each item, total marks: 30)

1. The output of the following program is _____.

```
#include <stdio.h>
void main(void)
{
    int x,y=-2,z;
    if((z=y)<0) x=4;
    else if(y==0) x=5;
    else x=6;
    if(z=(y==0))
        x=5;
    printf("%d#%d#",x,z);
    if(x=z=y)
        x=4;
    printf("%d$%d$",x,z);
}
```

2. The output of the following program is _____.

```
#include <stdio.h>
int x,y,z;
void p(int x, int *y)
{
    int z;
    ++x;
    ++*y;
    z=x+*y;
    printf("%d#%d#%d#", x,*y,z);
}
void main()
{
    x=y= 2;
    p(y, &x);
    printf("%d#%d#%d#", x,y,z);
}
```

3. The output of the following program is _____.

```
#include <stdio.h>
void main( )
{   char ch, *s = "ACEDB";
    int x;
    while(*s != 0){
        f(*s++, &x);
        printf("%d#", x);
    }
}
int f(char ch, int *x)
{
    *x = 0;
    switch (ch) {
        case 'A': *x = 95; break;
        case 'B': *x = 85; break;
        case 'C': *x = 75; break;
        case 'D': *x = 60; break;
    }
}
```

4. The output of the following program is _____.

```
#include <stdio.h>
void main()
{   void div(int*, int*);
    int a[5] = {-1, 0, 60, 45, 34 }, i = 0, j = 2;

    while(a[i] <= 0) i++;
    while(a[i] != 1) div(a+i, &j);
}

void div(int *n, int *k)
{   if(*n%*k == 0) {
        printf("%d,%d#", *n, *k);
        *n /= *k;
    }
    else (*k)++;
}
```

5. The output of the following program is _____.

```
#include <stdio.h>
int f(int a[], int n)
{   int cur, final, j;
    cur=final=0;
    for(j=0; j<n; j++) {
        cur += a[j];
        if(cur > final) final= cur;
        else if(cur<0) cur=0;
```

```
    }
    return final;
}
void main()
{
    int a[]={1,2,-4,3,2,6,-1,5,-4,-7,8};
    printf("%d\n", f(a, sizeof(a)/sizeof(int)));
}
```

6. The output of the following program is _____.

```
#include <stdio.h>
int f(int n, int m)
{
    int n1, m1, result=0;
    if (n && m) {
        n1=n%10;
        m1=m%10;
        if (n1==m1) result= f(n/10, m/10)*10;
        else result = f(n/10, m/10)*10 + ((n1>m1)?n1:m1);
        return result;
    } else return n+m;
}
void main()
{
    printf("%d\n", f(12345, 3333));
}
```

Section 4: According to the specification, complete each program (2 mark for each blank, total: 20 marks)

1. The following program is to calculate the sum of the first **n** items of the sequence as $1/2, 2/3, 3/5, 5/8, 8/13, 13/21, \dots$

```
#include <stdio.h>
main()
{
    float x,y,t, sum=0;
    int i,n;
    x=1,y=2;
    scanf("%d", &n);
    for(i=0; ____ (1) ____ ; i++) {
        sum+= x/y;
        t=x;
        ____ (2) ____ ;
        ____ (3) ____ ;
    }
    printf("%f", sum);
}
```

2. There is an increasing ordered character list in a text file **file1**. The following program

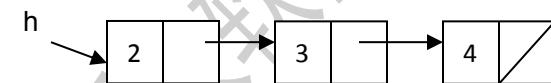
read in this list, calculate the number of duplicates(重复) and write each character and its frequency of occurrence (> 1) (大于 1 的出现次数) into the file **file2**. For example, if the **file1** contains list “abbcdddeffg” , the list “ab2cd3ef2g” will be written into **file2**.

```
#include <stdio.h>
main()
{   FILE *fp1, *fp2;
    char last, c;
    int count=0;

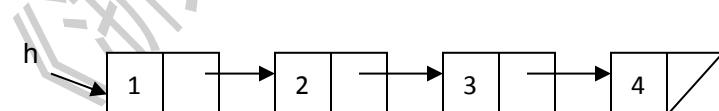
        fp1=fopen("file1", "r");
        fp2=fopen("file2", "w");
        if (_____ (4) _____) exit(0);
        last='\\0';
        while (_____ (5) _____) {
            count++;
            if (c!=last) {
                if (count>1) _____ (6) _____;
                count=0;
                fputc(c,fp2);
                last=_____(7)_____;
            }
        }
        fclose(fp1);
        fclose(fp2);
    }
```

3. There is a simple chaining list **h** (may be NULL). The following two functions: **void push(int n)** to insert the **n** into the head of the list, and **void pop()** to delete the first node from the list.

For example,



after push(1), list will be:



```
struct node {
    int info;
    struct node *next;
} *h;
void push(int n)
{   struct node *t;
    t=_____ (8) _____;
    t->info=n;
```

```
if (h==NULL) t->next=NULL;
else _____(9)_____;
h=t;
}
void pop()
{ struct node *t;
if (h==NULL) return NULL;
else { t=h;
       _____(10)_____;
       free(t);
}
}
```

浙江大学 2005-2006 学年冬季学期

《C Programming》课程期末考试题参考答案

开课学院：计算机学院， 考试形式：闭卷

考试时间：2006 年 1 月 11 日，所需时间：120 分

姓名

学号

订.....线.....

装.....业.....

Section	1	2	3	4	Total	
Subtotal	20	30	30	20		
Mark					Teacher1	
Teacher					Teacher2	

Section 1: Single Choice (2 mark for each item, total 20 marks)

1 C
6 C

2 A
7 A

3 D
8 D

4 B
9 B

5 A
10 C

Section 2: Fill in the blanks (2 mark for each item, total 30 marks)

1 _____ 'A'+'Z'-ch 2 _____ 20

3 _____ 20 4 _____ $y=x>0?1/x:(x==0)?0:-2/x$

5 _____ 12 6 _____ -1

7 _____ 6,4,60 _____ 8 _____ -2,1 _____

9 _____ a=2,b=1 _____ 10 _____ typedef char *NAME[20] _____

11 _____ 11 _____ 12 _____ 3 _____

13 _____ 3#1#1# _____ 14 _____ u # - _____

15 _____ 6 _____

Section 3: Read each of the following programs and answer questions (5 marks for each item, total marks: 30)

1 _____ 4#0#4\$-2\$ _____

2 _____ 3#3#6#3#2#0# _____

3 _____ 95#75#0#60#85# _____

4 _____ 60,2#30,2#15,3#5,5# _____

5 _____ 15 _____

6 _____ 13045 _____

Section 4: According to the specification, complete each program (2 mark for each blank, total: 20 marks)

(1) _____ i<n 或 i<=n-1 _____ (2) _____ x=y _____

(3) _____ y+=t 或 y=t+y 或 y=t+x _____ (4) _____ fp1=NULL||fp2==NULL _____

(5) _____ ((c=fgetc(fp1))!=EOF) _____ (6) _____ fprintf(fp2,"%d",count) _____

(7) _____ c _____ (8) _____ (struct node*)malloc(sizeof(struct node)) _____

(9) _____ t->next=h _____ (10) _____ h=h->next _____

浙江大学 2006-2007 学年夏季学期

《程序设计基础及实验》课程期末考试试卷 (A)

开课学院：计算机学院， 考试形式：闭卷

考试时间：2007 年 7 月 7 日，所需时间：120 分钟，任课教师_____

考生姓名：_____ 学号：_____ 专业：_____

(注意：答题内容必须写在答题卷上，写在本试题卷上无效)

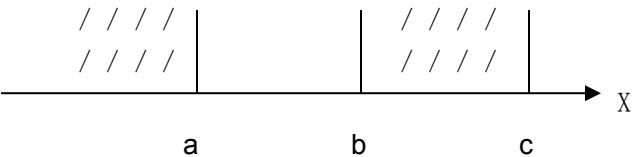
Section 1: Single Choice (2 marks for each item, total 20 marks)

1. According to the declaration: char ch; the assignment expression _____ is correct.
A. ch='412' B. ch=255 C. ch='08' D. ch=''
2. According to the declarations: float x=16/5/2.0, y=16/5.0/2; the values of x and y are _____ respectively.
A. 1.5 1.6 B. 1.6 1.6 C. 1.5 1.5 D. 1.6 1.5
3. According to the declarations: int a[10], *p=a; the expression *p++ is equivalent to _____.
A. a[0]++ B. a[1]++ C. *(p++) D. (*p)++
4. In the following initializations/assignments, _____ is correct.
A. char s[10]; s="hello!"; B. int x[2][]={1,2,4,6};
C. int x[][3]={1,2,3,4,5,6}; D. char s[10]; s[]="hello!";
5. In the following statements, _____ will print out "%d".
A. printf("%d"); B. printf("%%d"); C. printf("\%d"); D. printf("%%%d")
6. The following code fragment prints out _____.

```
int x=1, y=012;
printf("%d",y*x++);
```

A. 12 B. 10 C. 20 D. 24
7. The following code fragment prints out _____.

```
int a=1,b=2,c=2,t;
while(a<b<c) { t=a; a=b; b=t; c--; }
printf("%d,%d,%d",a,b,c);
```

A. 1,2,0 B. 2,1,0 C. 1,2,1 D. 2,1,1
8. In the following figure, the shading regions of the axis can be described as expression _____.


A. $(x \leq a) \&\& (x \geq b) \&\& (x \leq c)$
B. $(x \leq a) \mid\mid (b \leq x \leq c)$
C. $(x \leq a) \mid\mid (x \geq b) \&\& (x \leq c)$
D. $(x \leq a) \&\& (b \leq x \leq c)$

9. The following code fragment will print out _____.

```
int x = -3, y = 2;
```

```
if(x>y);
```

```
    x = -x;
```

```
if(x<y)
```

```
    y = -y;
```

```
printf("x=%d,y=%d\n", x, y);
```

- A. x=3,y=-2 B. x=3,y=2 C. x=-3,y=2 D. x=-3,y=-2

10. According to the declaration: int x[5]; _____ of the following operations can increase the value of the second element of x by 1 (Assume that x[0] is counted as 1st element).

- A. x[2]++;
C. x++; (*x)++;

- B. int p=x[1];p++;
D. int p=x+1; (*p)++;

Section 2: Fill in the blanks (2 marks for each item, total 30 marks)

1. The value of expression 13&(6<<1) is _____.

2. The following **for** loop statement will print out: 0 1 2 0 1 2 0 1 2.

```
for( i=1; i<=9; i++ ) printf("%2d", _____);
```

3. The value of expression sizeof("%%\%dxyz") is _____.

4. The following code fragment prints out _____.

```
printf("%d# %X#", 027, 27);
```

5. According to the declaration: char s[]={‘s’, ‘t’, ‘r’, ‘0’, ‘l’, ‘e’, ‘n’, 0, ‘=’, ‘\0’}; the value of expression strlen(s+1) is _____.

6. The following code fragment prints out _____.

```
unsigned char x=255,y='1';
```

```
switch(!lx) {
```

```
    case 0: printf("*0*#");break;
```

```
    case 1:
```

```
        switch(y){
```

```
            case 0: printf("*1*#");break;
```

```
            case 1: printf("*2*#");break;
```

```
        }
```

```
        default: printf("*3*#");
```

```
}
```

7. According to the function declaration: **void main(int argc, char *argv[])** in the file **learn.c**, run the command: **learn the C programming language**, the value of expression *(*(argc+argc-1)+4) is _____.

8. The following program prints out _____.

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

```
void f(int x,int y)
```

```
{
```

```
    int t=x;
```

```
    x=y; y=t;
```

```
}
```

```
void main()
```

```
{
```

```
    int x=3, y=2;
```

```
    f(y, x);
```

```
    printf("%d%d", x, y);
```

```
}
```

9. The following program prints out ____.

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

```
int s;
```

```
int f(int m)
```

```
{
```

```
    static int k=0;
```

```
    for( k<=m; k++) s++;
```

```
    return s;
```

```
}
```

```
void main( )
```

```
{
```

```
    int s=1;
```

```
    s=f(2)+f(2);
```

```
    printf("%d%d", s, f(20));
```

```
}
```

10. The following program prints out ____.

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

```
int f(int n)
```

```
{
```

```
    if (n > 1)
```

```
        return f(n-1)+f(n-2);
```

```
    return 1;
```

```
}
```

```
int main()
```

```
{
```

```
    printf("%d", f(10));
```

```
    return 0;
```

```
}
```

11. The following program prints out ____.

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

```
#define P(a) a*a+5
```

```
#define Q(a, b) a*b
```

```
int main()
```

```
{
```

```
    printf("%d", P(3)/Q(2, 3));
```

```
    return 0;
```

- }
12. The following code fragment prints out _____.
char *p="";
printf("%d", p==NULL?0:1);
13. Use **typedef** to define a pointer type **PA** that points to an one-dimension array which contains 10 elements of type int _____.
14. For the following array a, the value of *(a[1]+4) is _____.
int a[][3]={1,2,3,4,5,6,7,8,9};
15. For the following definition, the value of s[1].b/++p->a is _____.
struct {
 int a;
 int b;
} s[2]={2,4,6,8}, *p=s;

Section 3: Read each of the following programs and answer questions (5 marks for each item, total marks: 30)

1. The output of the following program is _____.
#include <stdio.h>

```
int main()  
{  
    int a = 57, b = 113;  
    b^=a ^= b ^= a ^= b;  
    printf("a=%d#b=%d#", a, b);  
    return 0;  
}
```

2. When input: 26734<ENTER>, the output of the following program is _____.
#include <stdio.h>
#include <string.h>

```
int main()  
{  
    char n[1000];  
    int i, tmp1, tmp2;  
    scanf("%s", n);  
    i = 0;  
    tmp1 = 0;  
    while(n[i]!='0') {  
        tmp1 += n[i] - '0';  
        i++;  
    }  
    while(tmp1>9) {  
        tmp2 = tmp1;  
        tmp1 = 0;  
        while(tmp2!=0) {  
            tmp1 += tmp2%10;
```

- ```
 tmp2 /= 10;
 }
}
printf("%d\n", tmp1);
return 0;
}

3. The output of the following program is _____.

#include <stdio.h>
char *c[] = {"ENTER", "NEW", "POINT", "FIRST"};
char **cp[] = {c+3, c+2, c+1, c};
char ***cpp = cp;
main()
{
 printf("%s#", **++cpp);
 printf("%s#", *--++cpp);
 printf("%s#", *cpp[-2]+3);
 printf("%s#", cpp[-1][-1]+1);
}

4. The output of the following program is _____.

#include <stdio.h>
void f(int A[], int n, int x)
{
 int t, i;
 i=n; A[n+1]=x;
 while ((i>=0) && (A[i]>A[i+1])) {
 t=A[i];
 A[i]=A[i+1];
 A[i+1]=t;
 i--;
 }
 n++;
}

int A[10]={0,2,4,6,8,10,12,14,10};
void main()
{
 int n=sizeof(A)/sizeof(A[0]);
 f(A,7,A[8]);
 f(A,8,A[9]);
 while (n--)
 printf("%d#", A[n]);
}

5. The output of the following program is _____.

#include <stdio.h>
```

```
#define NUM 5
void p1(char v[])
{
 int i,j,temp;
 for (i=1; i<NUM; i++)
 for(j=i-1; j>=0&&v[j]<v[j+1]; j--)
 temp = v[j], v[j]=v[j+1], v[j+1]=temp;
}
void p2(char v1[], char v2[])
{
 int i=0, j=0;
 while (i<NUM && j<NUM)
 if (v1[i]>v2[j]) putchar(v1[i++]);
 else putchar(v2[j++]);
 while (i<NUM) putchar(v1[i++]);
 while (j<NUM) putchar(v2[j++]);
}
void main()
{
 char s[2][5]={'v','d','x','c','b','e','a','y','z','w'};
 p1(s[0]);
 p1(s[1]);
 p2(s[0],s[1]);
}
```

6. The text file **a.txt** contains the following three lines(<ENTER> for the **Enter** key):

*Zhejiang University<ENTER>*  
*Programming <ENTER>*  
*Hello, World!*

In the following command-line program **prog.c**, the output of the program after running as “**prog -n a.txt**” is \_\_\_\_\_.

```
#include <stdio.h>
#include <string.h>
void main(int argc, char *argv[])
{
 int flag=0,count=0;
 FILE *fp;
 char *file, c;
 if (argc == 2)
 file = argv[1];
 else if (argc == 3 && strcmp(argv[1], "-n") == 0) {
 file = argv[2]; flag = 1;
 } else {
 printf("Usage: prog [-n] filename\n");
 exit(-1);
```

```
 }
 if ((fp=fopen(file, "r")) == NULL) {
 printf("Cannot open file \"%s\"\n", file);
 exit(-1);
 }
 while ((c=fgetc(fp))!=EOF) {
 if (flag && (count == 0 || c == '\n'))
 printf("%d:", ++count);
 putchar (c=='\n'?c:'#');
 }
 fclose(fp);
}
```

**Section 4: According to the specification, complete each program (2 marks for each blank, total: 20 marks)**

- Given a simple chain list, which contains a special head node **header** with no value (This list is null if header->next==NULL). The following program fragment inserts **value** node to the list, makes its value be greater than that of its previous node, and be less than or be equal to that of its next node. Fill in the blanks.

```
#include <stdio.h>
#include <stdlib.h>
struct student
{
 int num;
 struct student *next;
};
void insert(struct student *header, int value)
{
 struct student *tmp, *stud;

 stud = (struct student *) malloc(sizeof(struct student));
 _____(1)_____;
 tmp = header;
 while(tmp->next != NULL) {
 if(stud->num <= tmp->next->num) {
 stud->next = tmp->next;
 _____(2)_____;
 return;
 }
 tmp = tmp->next;
 }

 tmp->next = stud;
 _____(3)_____;
```

- }
2. There is a text file, which contains a number (less than 100) of positive decimal integers (一批不超过 100 个的十进制正整数). Write a command-line program **prog.c** to read in these numbers, count the average value of all even numbers(偶数), and find out the number(s) which sum of all digits of each number (各位数字之和) is greater than 10, and store these results above into the other text file. Requirements: take the input and output file names as command-line parameters. If the output file name is omitted(省略), the default output will be the standard terminal. Fill in the blanks.

For example:

The command-line is:

**prog infile.txt outfile.txt**

The input text file **infile.txt** contains the following numbers:

123 234 345 456 567 678 789 890 901 012

The result be stored in the output file **outfile.txt** will contain the followings:

454.00

345 456 567 678 789 890

```
#include <stdio.h>
#define MAX_NUM 100
```

```
int ds(int num) /*count whether the sum of each digits of num is greater than 10 */
{
```

```
 int s = 0;
```

```
 while (num > 0) {
```

```
 ____(4)__;

```

```
 num /= 10;
```

```
}
```

```
 if (s > 10)
```

```
 return(1);
```

```
 return(0);
}
```

```
void main(int argc, char *argv[])
{
```

```
 int a[MAX_NUM], n, k, i, sum;
```

```
 FILE *fp_in, *fp_out;
```

```
 float aver;
```

```
 if (argc < 2) {
```

```
 printf("Usage: prog in_file [out_file]\n");
```

```
 exit(0);
 }
```

```
 if (_____(5)_____) {
```

```
 printf("Cannot open input file %s\n", argv[1]);
```

```
 exit(0);
 }
```

```
}

if (argc > 2) {
 if ((fp_out=fopen(argv[2], "w")) == NULL) {
 printf("Cannot open output file %s\n", argv[2]);
 fp_out = stdout;
 }
} else
 _____(6) _____;

n = 0;
while (!feof(fp_in)) {/*feof() return 1 if end of file has been reached, return 0
otherwise*/
 _____(7) _____;
 n++;
}
fclose(fp_in);

sum = 0; k = 0;
for (i = 0; i < n; i++) {
 if (a[i]%2 == 0) {
 sum += a[i];
 k++;
 }
}
_____ (8) _____;
fprintf(fp_out, "%.2f\n", aver);
for (i = 0; i < n; i++)
 if (ds(a[i]))
 _____(9) _____;
if (_____ (10) _____)
 fclose(fp_out);
}
```

## 浙江大学 2006-2007 学年夏季学期答案

### 《程序设计基础及实验》课程期末考试参考答案 (A)

开课学院：计算机学院， 考试形式：闭卷

考试时间：2007 年 7 月 7 日， 所需时间：120 分钟

姓名

| Section | 1 | 2 | 3 | 4 | Total |
|---------|---|---|---|---|-------|
|         |   |   |   |   |       |

|          |    |    |    |    |          |  |
|----------|----|----|----|----|----------|--|
| Subtotal | 20 | 30 | 30 | 20 |          |  |
| Mark     |    |    |    |    | Teacher1 |  |
| Teacher  |    |    |    |    | Teacher2 |  |

**Section 1: Single Choice (2 marks for each item, total 20 marks)**

1 B

2 A

3 C

4 C

5 B

6 B

7 A

8 C

9 B

10 D

**Section 2: Fill in the blanks (2 marks for each item, total 30 marks)**

1 \_\_\_\_\_ 12 \_\_\_\_\_ 2 \_\_\_\_\_  $(i-1) \% 3$  \_\_\_\_\_

3 \_\_\_\_\_ 9 \_\_\_\_\_ 4 \_\_\_\_\_ 23#1B# \_\_\_\_\_

5 \_\_\_\_\_ 6 \_\_\_\_\_ 6 \_\_\_\_\_ \*2\*#\*3\*# \_\_\_\_\_

7 \_\_\_\_\_ u \_\_\_\_\_ 8 \_\_\_\_\_ 3#2# \_\_\_\_\_

9 \_\_\_\_\_ 6#21# \_\_\_\_\_ 10 \_\_\_\_\_ 89 \_\_\_\_\_

11 \_\_\_\_\_ 15 \_\_\_\_\_ 12 \_\_\_\_\_ 1 \_\_\_\_\_

13 \_\_\_\_\_ `typedef int (*PA)[10]` \_\_\_\_\_ 14 \_\_\_\_\_ 8 \_\_\_\_\_

15 \_\_\_\_\_ 2 \_\_\_\_\_

**Section 3: Read each of the following programs and answer questions ( 5 marks for each item, total 30 marks)**

1 \_\_\_\_\_ `a=113#b=72#` \_\_\_\_\_

2 \_\_\_\_\_ 4 \_\_\_\_\_

3 \_\_\_\_\_ `POINT#ENTER#ST#EW#` \_\_\_\_\_

4 \_\_\_\_\_ `14#12#10#10#8#6#4#2#0#0#` \_\_\_\_\_

5 \_\_\_\_\_ `zyxwvedcba` \_\_\_\_\_

---

6 1:Zhejiang University#2:Programming #3>Hello, World!

---

**Section 4: According to the specification, complete each program (2 mark for each blank, total 20 marks)**

- (1) stud->num=value (2) tmp->next=stud  
(3) stud->next=NULL (4) s+=num%10  
(5) (fp\_in=fopen(argv[1], "r")) == NULL (6) fp\_out = stdout  
(7) fscanf(fp\_in, "%d", &a[n]) (8) aver = (float)sum/(float)n  
(9) fprintf(fp\_out, "%d ", a[i]); (10) fp\_out!=stdout
- 

**四、程序填空题（每空 2 分，共 30 分）**

- (1) s=0 (2) f=1  
(3) j<=k (4) f=f\*j  
(5) prime(n) (6) m%i==0  
(7) 0 (8) return 1  
(9) ss=s (10) '\0'  
(11) s-ss-1 (12) ptr=f(s, 30)  
(13) pmax=stud (14) pmax=p  
(15) pmax
-

# 浙江大学 2007 - 2008 学年夏季学期

## 《C 程序设计基础与实验》课程期末考试试卷

开课学院: 计算机学院, 考试形式: 闭卷, 允许带        /        入场

考试时间: 2008 年 6 月 26 日上午 10:30-12:30, 所需时间: 120 分钟

任课教师: \_\_\_\_\_

考生姓名: \_\_\_\_\_ 学号: \_\_\_\_\_ 专业: \_\_\_\_\_

注意: 答题内容必须写在答题卷上, 写在本试题卷上无效。

### 一、单选题(每题 1 分, 共 10 分)

1. 假定以下表达式中所有变量均已定义为 int 类型并已赋值, 请问其中哪个 表达式没有语法错误?

- A.  $x = 3.14\%2;$
- B.  $y = 2a+b;$
- C.  $m = a < b < c;$
- D.  $n = (-x)++;$

2. 表达式 'A'+'B' 的类型为:

- A. char
- B. int
- C. double
- D. void

3. 执行以下这段程序后,

```
unsigned char a=0;
int b=1;
do {
 b++;
 a--;
} while(a!=0);
```

变量 b 的值等于多少?

- A. 0
- B. 1
- C. 128
- D. 257

4. 如果一个函数的返回值类型为 void, 则以下哪个说法是正确的?

- A. 该函数中不可以用 return 语句
- B. 该函数中可以用 return 语句, 但必须是最后一个语句
- C. 该函数中可以用 return 语句, 但 return 后面不能跟任何表达式
- D. 该函数可以返回一个任意类型的值, 也可以没有返回值

5. 执行以下程序后,

```
char a[]="ABC\\XYZ\\007\\xFF";
```

- ```
int n;
n = sizeof(a);
```
- 变量 n 的值等于多少?
- A. 10 B. 14 C. 9 D. 17
6. 用 scanf() 输入一个 double 类型的变量的值，则输入格式要用
- A. "%f"
B. "%lf"
C. "%f" 或 "%lf"
D. "%ld"
7. 假定有以下变量定义语句,
- ```
float * p,q;
```
- 请问哪个说法是正确的?
- A. p 和 q 的类型均为(float \*)  
B. p 的类型为(float \*), q 的类型为 float  
C. p 的类型为 float, q 的类型为(float \*)  
D. 此变量定义语句有语法错误
8. 执行以下程序后,
- ```
char a = '0';
printf("%X", a);
```
- 将输出什么结果?
- A. 30 B. 0x30 C. 0X30 D. 0
9. 假定有如下定义,
- ```
struct st {
 char name[10];
 int score;
} a[3], *p=a;
```
- 请问哪个用法是正确的?
- A. p->name="Tom";  
B. \*p.name[0]='B';  
C. (\*p)->name[0]='C';  
D. \*(\*p).name='D';
10. 执行以下语句后,
- ```
int n = strcmp("ABC", "AB");
```
- 则以下关于 n 值的描述哪一个是正确的?
- A. n=0
B. n=1
C. n>1
D. n<0

二、填空题(每题 2 分, 共 20 分)

1. 执行以下代码后, 将输出_____。

```
#define FOR while(x--);
int x=3, y=0;
FOR
{
```

```
y++;  
}  
printf("x=%d, y=%d", x, y);
```

2. 执行以下代码后，将输出_____。

```
int a=2, b=5, c=8;  
c |= (~a && b) ^ !(a & b);  
printf("c=%d", c);
```

3. 执行以下代码后，变量 d 的值=_____。

```
int a=-10, b=0, c=20, d;  
d = a > b ? -3.14 : b > c ? 2.68 : 4.99;
```

4. 执行以下代码后，变量 x 的值=_____。

```
int a=3, b=4, c=3, x;  
x = a != (b = c);
```

5. 执行以下代码后，将输出_____。

```
char t[]="BUGWET";  
int a[]={5,1,2,3,4,0,2,4,5}, i=0, go;  
do  
{  
    putchar( t[go=a[i++]] );  
} while(go);
```

6. 若定义 `unsigned long int a`;且变量 a 已赋值，试写出判断 a 的二进制最高位即最左边那位与 1 相等的表达式(要求不调用任何函数)_____。

7. 执行以下代码后，将输出_____。

```
long int a[5]={10, 20, 30, 40, 50};  
long int *p = a+1;  
long int *q = &a[3];  
printf("q-p=%d", q-p);
```

8. 执行以下代码后，变量 max 的值=_____。

```
int x=3, y=2, z=1, max;  
if(x>=y && x>=z)  
    max = x;  
if(y>=x && y>=z)  
    max = y;  
else  
    max = z;
```

9. 执行以下代码后，将输出_____。

```
char a[][5]={"ABCD", "1234", "BOMB"};  
printf("%s,%c", *a+1, *a[2]+1);
```

10. 若定义函数时省略该函数的返回值类型，则它的实际返回值类型是_____。

三、程序阅读题(每题 5 分，共 30 分)

1. 以下程序的输出结果是_____。

```
#include <stdio.h>  
#include <string.h>  
typedef char (*AP)[5];
```

```
AP defy(char *p)
{
    int i;
    for(i=0; i<3; i++)
        p[strlen(p)] = 'A';
    return (AP)p + 1;
}
void main()
{
    char a[]="FROG\0SEAL\0LION\0LAMB";
    puts( defy(a)[1]+2 );
}
```

2. 以下程序的输出结果是_____。

```
#include <stdio.h>
void main()
{
    int i, j, sum=0;
    for(i=1; i<=3; i++)
    {
        if(i==2)
            continue;
        for(j=1; j<=6; j++)
        {
            sum += j;
            if(sum >= 6)
                break;
        }
        sum += i*10;
    }
    printf("sum=%d", sum);
}
```

3. 以下程序的输出结果是_____。

```
#include <stdio.h>
int g=96;
void gf(void)
{
    g++;
}
int df(void)
{
    int g=10;
    g++;
    return g;
}
int * sf(void)
{
    static int g=20;
    g++;
    return &g;
}
void main()
{
    int a, b, c, *p;
```

```
gf(); gf(); a=g;
b=df(); b=df();
p=sf(); *p=32765; p=sf(); c=*p;
printf("a=%d, b=%d, c=%d", a, b, c);
}
```

4. 假定运行以下程序时输入

$3+4*5-3/2=$

则程序的输出结果是_____。

```
#include <stdio.h>
int calc(void)
{
    int x; char p;
    scanf("%d %c", &x, &p);
    if(p == '=')
        return x;
    switch(p)
    {
        case '+': return x + calc();
        case '-': return x - calc();
        case '*': return x * calc();
        case '/': return x / calc();
    }
}
void main()
{
    printf("%d", calc());
}
```

5. 以下程序的输出结果是_____。

注意：本题输出结果中的空格请用_____标出，一个空格用一个_____代替。_____

```
#include <stdio.h>
#define N 4
void main()
{
    static int a[N+1][N+1]={1};
    int i, j;
    for(i=1; i<=N; i++)
        for(j=1; j<=i; j++)
            a[i][j] = a[i-1][j-1] + a[i-1][j];
    for(i=1; i<=N; i++)
    {
        int space = N-i;
        while(space--)
            putchar(' ');
        for(j=1; j<=i; j++)
            printf("%-2d", a[i][j]); /* 左对齐输出 */
        printf("\n"); /* 左数字右空格 */
    }
}
```

6. 以下程序的输出结果是_____。

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

```
#define N 5
void main()
{
    static int a[N][N], i, j;
    int left=0, right=N-1, up=0, down=N-1;
    int x=1, row=right, col=down, d=1, done;
    while(x<=N*N)
    {
        a[row][col]=x;
        x++;
        done = 0;
        do
        {
            switch(d)
            {
                case 1: col--;
                    if(col<left) {col=left; d=2; continue;}
                    else {done=1; break;}
                case 2: row--;
                    if(row<up) {row=up; d=3; continue;}
                    else {done=1; break;}
                case 3: col++;
                    if(col>right) {col=right; d=4; continue;}
                    else {done=1; break;}
                case 4: row++;
                    if(row==down) {
                        down--; left++; up++; right--;
                        row=down; d=1; continue; }
                    else {done=1; break;}
            } /* end of switch */
        } while(!done);
    } /* end of while(x<=N*N) */
    for(i=0; i<N; i++)
    {
        for(j=0; j<N; j++)
            printf("%-3d", a[i][j]);
        printf("\n");
    }
}
```

四、程序填空题(每空 2 分, 共 40 分)

- 输入 n ($2 \leq n \leq 5$, 程序不需要对此范围进行判断), 再输入 n 个整数保存到数组 a 中, 通过循环查找 n 个数中是否有重复的数, 如果有则输出 Yes, 否则输出 No。要求在循环过程中, 任何两个数的比较次数不得超过 1 次(比如对 $a[0]$ 与 $a[1]$ 比较后接下去又对 $a[1]$ 与 $a[0]$ 比较是不符合要求的), 并且要求一旦找到有数重复则立即结束循环。

```
#include <stdio.h>
void main()
{
    int a[5], i, j, n;
    scanf("%d", &n);
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    for(i=0; i<=n-2; i++)
```

```
{  
    for(_____(1)_____; j<=n-1; j++)  
    {  
        if(a[i] == a[j])  
            _____(2);  
    }  
    if(_____(3)_____)  
        break;  
}  
if(_____(4)_____)  
    puts("No");  
else  
    puts("Yes");  
}
```

2. hex()是一个递归函数，该函数的功能是通过递归调用把一个长整数转化成十六进制字符串并返回所得十六进制数的位数。main()输入长整数 x，调用 hex()对它进行转化（转化结果存放在字符数组 a 中），最后输出结果。例如：输入 13，则返回 n=1 输出 D；输入 100，则返回 n=2 输出 64；输入 65535，则返回 n=4 输出 FFFF。

```
#include <stdio.h>  
void main()  
{  
    char a[9];  
    int n;  
    unsigned long int x;  
    scanf("%lu", &x);  
    n = hex(x, a);  
    _____(5);  
    puts(a);  
}  
  
int hex(_____(6)_____)  
{  
    char t[]="0123456789ABCDEF";  
    int n;  
    if(x<16)  
    {  
        a[0] = t[x];  
        return 1;  
    }  
    n = hex(x/16, a);  
    a[n] = _____(7);  
    return _____(8);  
}
```

3. 设文件 file1.txt 中以如下文本格式存放一批学生的姓名及成绩：

Tom 80

Jerry 90

姓名与成绩之间用至少一个空格分隔并且每个学生的姓名中不含空格。现通过编程读取 file1.txt 中的资料，找出其中成绩最高的那位学生（假定最高分唯一），并把该生的资料写入 file2.txt 中。

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

```
typedef struct student
{
    char name[10];
    int score;
} ST;
void main()
{
    FILE *fp1;
    FILE *fp2;
    ST a, max;
    fp1 = fopen("file1.txt", "r");
    fp2 = fopen("file2.txt", "w");
    if( (9) )
    {
        puts("Cannot open file!");
        exit(0);
    }
    max.score = -1;
    while(_____(10)__)
    {
        _____(11);
        if(a.score > max.score)
        {
            _____(12);
            max.score = a.score;
        }
    }
    fprintf(fp2, "%s %d\n", max.name, max.score);
    fclose(fp1);
    fclose(fp2);
}
```

4. `rotate()`函数的功能是把字符串 `s` 循环右移 `t` 次，所谓循环右移是指各个字符分别向右移动 1 个位置，原先最右边的字符移出后填到最左边的空位上。例如字符串"ABC123"右移 1 次就变成"3ABC12"，字符串"ABCXYZ"右移 2 次就变成"YZABCX"。函数 `main()`输入一行字符存放到字符数组 `a` 中，调用 `rotate()`循环右移 3 次，最后用 `puts()`输出结果。

```
#include <stdio.h>
#include <string.h>
void rotate(char *s, int t)
{
    int i,j,n;
    char c;
    _____(13);
    for(i=0; i<t; i++)
    {
        c = s[n-1];
        for(_____(14)__)
            s[j] = s[j-1];
        _____(15);
    }
}
void main()
{
```

```
char a[81];
gets(a);
_____(16)_____;
puts(a);
}
```

5. 输入一个英文句子保存到字符数组 a 中（句子中各个单词用一个或多个空格分隔，第一个单词前面可以有空格，最后一个单词后面也可以有空格），接下去对句子中构成一个单词的各个字符进行逆向变换，逆向后的单词重新存放到数组 a 中原先的位置，最后输出新的句子。例如，输入以下句子：

pots & pans I saw a reed

将输出

stop & snap I was a deer

注意：只要是非空格字符均可构成一个单词，例如这里的&也算一个单词；另外，原句子中的空格经过变换后必须保持原样，不可增加或减少。

```
#include <stdio.h>
void main()
{
    char a[81], c;
    int i, j, n;
    gets(a);
    i=0;
    while(a[i]!='\0')
    {
        while(a[i]==' ')
            i++;
        if(_____(17)_____)
            break;
        n=0;
        while(_____(18)_____)
            n++;
        for(j=0; j<n/2; j++)
        {
            c = a[i+j];
            a[i+j] = _____(19)_____ /* 本行所空与下行同 */;
            _____(19)_____ = c; /* 本行所空与上行同 */
        }
        _____(20)_____;
    } /* end of while(a[i]!='\0') */
    puts(a);
}/* end of main() */
```

浙江大学 2007 - 2008 学年夏季学期答案

《C 程序设计基础与实验》课程期末考试答卷

一、单选题(共 10 题，每小题 1 分，共 10 分)

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

二、填空题(共 10 题，每小题 2 分，共 20 分)

1	x=-1, y=1	6	(a>>31)==1 或~a<a 或 a&0x80000000==1 或 a&(1<<31)或(long int)a<0
2	c=8	7	q-p=2
3	4(写成 4.99 不扣分)	8	1
4	0	9	BCD,C
5	TUGWEB (TUGWEBGET 扣 1)	10	int

三、程序阅读题(共 6 题，每小题 5 分，共 30 分)

1
ONALAMB
(ON 得 2 分； ONA 得 4 分)

2
sum=47

3
a=98, b=11, c=32766
(1 个 2 分， 2 个 4 分， 3 个 5 分)

4
19

任课教师：_____

专业：_____

学号：_____

考生姓名：_____

5
□ □ □ 1 □
□ □ 1 □ 1 □
□ 1 □ 2 □ 1 □
1 □ 3 □ 3 □ 1 □
(空格错乱不扣分)

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

四、程序填空题(每空 2 分, 共 40 分)

1	j=i+1	11	fscanf(fp1, "%s %d ", a.name, &a.score) 写出 fscanf 但格式错误扣 1 分
2	break	12	strcpy(max.name, a.name)
3	j<=n-1 (写成 a[i]==a[j]没有分)	13	n = strlen(s)
4	i==n-1 或 i>=n-1 或 i>n-2 或 j>n-1	14	j=n-1; j>=1; j-- (写成 j=1;j<=n-1;j++没分)
5	a[n] = '\0'	15	s[0] = c
6	unsigned long int x, char a[] 或 unsigned long int x, char *a	16	rotate(a,3) (写出 rotate 但参数格式错误扣 1 分)
7	t[x%16]	17	a[i]=='\0'
8	n+1	18	a[i+n]!=' ' && a[i+n]!='\0' (少写 1 个扣 1 分)
9	fp1==NULL fp2==NULL (写出 1 个给 1 分)	19	a[i+n-1-j]
10	!feof(fp1)	20	i+=n

注意：有些同学填写答案时可能横竖方向弄错，或者中间有一题做不出把后一题答案提前一格填在空位上造成错位。

浙江大学 2008-2009 学年春夏学期

《C 程序设计基础与实验》课程期末考试试卷

请考生仔细阅读以下注意事项：

1. 诚信考试，沉着应考，杜绝违纪。
2. 开课学院：计算机学院
3. 考试形式：闭卷，允许带_____入场
4. 考试日期：2009 年 6 月 21 日，考试时间：120 分钟，任课教师：_____

考生姓名：_____ 学号：_____ 所属院系：_____

(注意：答题内容必须写在答题卷上，写在本试题卷上无效)

试题一、单选题（每小题 1 分，共 10 分）

1. 已知 int k,m=1; 执行语句 k=-m++; 后，k 的值是_____。
A. -1 B. 0 C. 1 D. -2
2. 已知定义 int m; float k; 正确的表达式是_____。
A. (int k) %m B. int(k) %m C. int (k%m) D. (int) K%m
3. 当 k 的值不为 0 时，在下列选项中能够将 k 的值赋给变量 m, n 的是_____。
A. m=k=n B. (m=k) && (n=m)
C. (m=k) || (n=k) D. (n=m) && (m=k)
4. 已知 int x=5; 表达式 6>x>1 值是_____。
A. 0 B. 1 C. 5 D. 表达式语法错
5. 下列程序段执行后 c 值为_____。

```
Int k=7, c=0;  
While(--k)  
    if (k%3) continue; else c++;
```


A. 5 B. 4 C. 3 D. 2
6. 已知字母 a 的 ASCII 码为十进制数 97，且 ch 为字符型变量，则执行语句
ch='a'+'D'-'A'; 后，ch 中的值为_____。
A. d B. c C. 100 D. 不确定
7. 设 m, n 已正确定义，执行语句{for(m=0, n=10; m<n; m+=3, n--); }以后，
m, n 的值是_____。
A. 6, 7 B. 7, 6
C. 9, 7 D. 7, 9
8. 已知 char*t□={"How", "are", "you!"}, 则 t 是_____。

- A. 指向 char 变量的指针 B. 指向字符指针的指针常量
C. 指向“How”的指针变量 D. 指向‘H’的指针
9. 以下能使 puts(s)语句正确输出 ABCDE 字符串的程序段是_____。
A. char s[5] = {"ABCDE"}; puts(s)
B. char s[5] = {'A', 'B', 'C', 'D', 'E'}; puts(s);
C. char *s; s = "ABCDE"; puts(s);
D. char *s; scanf("%s", s); puts(s); /*假设执行时输入 ABCDE<回车>*/
10. 以下程序有错，错误原因是_____。

```
main()
{
    int *p, i;    char *q, ch;
    p = &i;
    q = &ch;
    *p = 40;
    *q = *p;
    .....
}
```


A. p 和 q 的类型不一致，不能执行*p=*q; 语句
B. *p 中存放的是地址值，因此不能执行*p=40; 语句
C. q 没有指向具体的存储单元，所以*q 没有实际意义
D. q 虽然指向了具体的存储单元，但该单元中没有确定的值，所以执行 *p=*q 没有意义，可能会影响后面语句的执行结果

试题二、填空题（每小题 2 分，共 30 分）

- 若有定义：int a=2, b=5, c=8；则表达式 c |= (~a && b) ^ !了(a & b)的值为_____。
- 设 int a=5, b=7, 表达式(++a == b--)? ++ a : --b 的值是_____。
- 设 a=3, b=2, 那么表达式(a=b=7, b=a!=7)的值是_____。
- 设有：int a=1, b=2, c=3, d=4, m=2, n=2；执行语句(m=a>b) && (n=c>d); 后 n 的值是_____。
- 下列程序段的输出结果是_____。

```
int k=2, s=0;
switch (k) {
    case 2: s++; k++;
    case 1: s++;
}
```

```
case 3: s+=1;  
    switch(s){  
        case 2: s+=2; break;  
        default: k++; break;  
    }  
    default: s+=3;  
}  
printf("%d#%d#", s, k);
```

6. 下列程序的输出结果是_____。

```
#include <stdio.h>  
int f(int x)  
{    static int k = 0;  
    return ++k +x;  
}  
main()  
{    int k;  
    for(k=1; k<4; k++) printf("%d#", f(k));  
}
```

7. 下列程序的输出结果是_____。

```
main()  
{    char x=040;  
    printf ("%o\n", x<<1);  
}
```

8. 下列函数 strncpy 实现将字符串 t 的第 m ($0 \leq m < \text{strlen}(t)$) 个字符开始拷贝到 s, 请填充完整。

```
void strncpy(char*s, char*t, unsigned int m)  
{    t += m;  
    while(______);  
}
```

9. 下列程序段的输出结果是_____。

```
main()  
{    int x= -9, y=5, z=8;  
    if (x==y) if (x>0) z=0; else z+=1;  
    printf ("z=%d\n", z);
```

}

10. 下列程序段的输出结果是_____。

```
#define SQ(a,b)  (a*a-2*a*b+b*b)  
main()  
{  int k=4;  
    printf("%d", SQ(k, k-1));  
}
```

11. 设变量定义为 char str[]={"ts=%d\n"}, 数组 str 中的元素个数是_____。

12. 语句 `typedef char *STR[8]` 的含义是_____。

13. 下列程序段输出_____。

```
#include <stdio.h>  
#include <string.h>  
main()  
{  char str[2][5] = {"abcd"};  
    strcpy(str[0], "You");  strcpy(str[1], "Me");  
    str[0][3] = '&';  
    printf("%s\n", *str);  
}
```

14. 对于函数 f, 调用 `f(12345)` 的输出是_____。

```
void f(int n)  
{  if(n<10)  printf("%1d", n);  
  else{  f(n/10);  printf("%1d", n%10);  }  
}
```

15. 运行下列程序段后, 文件中的内容是_____。

```
#include <stdio.h>  
#include <string.h>  
void fun(char *fname, char *st)  
{  FILE *myf;  int i;  
    myf=fopen(fname, "w");  
    for(i=0; i<strlen(st); i++)  fputc(st[i], myf);  
    fclose(myf);  
}  
void main()  
{
```

```
fun("test", "Programming_in_C_");
fun("test", "is_interesting");
}
```

试题三、程序阅读题（每小题 5 分，共 30 分）

1. 运行下面程序的输出结果是_____。

```
#include <stdio.h>
#include <stdio.h>
int x, y;
int f(int n)
{ static int x=0;
    x++;
    if(!(n%3)) printf("%d", n);
    else y++;
    return x;
}
void main()
{ int i;
    x=1; y=5;
    for(i=0; i<4; i++) x=f(i);
    printf("#%d#%d\n", x, y);
}
```

2. 运行下面程序的输出结果是_____。

```
#include <stdio.h>
void main()
{int a[3][3];
int i, j, temp;
for(i=0; i<3; i++)
    for(j=0; j<3; j++)
        a[i][j]=i+j*3+1;
for(i=0; i<3; i++)
    for(j=0; j<3; j++)
        if(i<=j){
            temp = a[i][j];
```

```
a[i][j] = a[j][i];
a[j][i] = temp;
}
for(i=0; i<3; i++)
    for(j=0; j<3; j++)
        printf("%d#", a[i][j];
}
```

3. 以下程序的输出结果是_____。

```
#include <stdio.h>
main()
{   char ch[2][5]={“6937”, “8254”}, *p[2];
    int i,j,s=0;
    for(i=0; i<2; i++)p[i]=ch[i];
    for(i=0; i<2; i++)
        for(j=0; p[i][j]>‘0’; j+=2)
            s=10*s+p[i][j]-‘0’;
    printf(“%d\n”, s);
}
```

4. 运行下列程序的输出结果是_____。

```
#include <stdio.h>
struct st
{   int x;
    int *y;
} *p;
int dt[4]={10, 20, 30, 40};
struct st aa[4]={50, &dt[0], 60, &dt[1], 70, &dt[2], 80, &dt[3]};
main()
{   p=aa;
    printf(“%d#”, ++p->x); printf(“%d#”, (++p)->x); printf(“%d”, ++(*p->y));
}
```

5. 下列程序的输出结果是_____。

```
#include <stdio.h>
#define M 100
void fun(int m, int *a, int *n)
```

```
{    int i,j=0;
    for(i=1; i<=m; i++)
        if(i%3==0 && i%7==0)
            a[j++]=i;
    *n=j;
}
void main()
{
    int aa[M], n, k;
    fun(50, aa, &n) ;
    for(k=0; k<n; k++)
        printf("%d#", aa[k]);
}
```

6. 运行下列程序的输出结果为_____。

```
#include <stdio.h>
#include <string.h>
void f(char *a[], int n); void swap(char **s1, char **s2);
main()
{
    int i,n=6;
    char *a[]={"China", "America", "Australia", "English", "Switzerland",
               "Italy"};
    f(a, n);
    for (i=0; i<n; i++)
        printf("%c", *(a[i]+i));
}
void f(char *a[], int n)
{
    int i, j;
    char *temp;
    for(i=1; i<n; i++)
        for(j=0; j<n-i; j++)
            if(strcmp(a[j], a[j+1])>0)swap(&a[j], &a[j+1];
}
void swap(char **s1, char**s2)
```

```
{  
    char *temp;  
    temp=*s1; *s1=*s2; *s2=temp;  
}
```

试题四、程序（段）填空题（每空 2 分，共 30 分）

1. 输出如下格式的九九表。

```
#include <stdio.h>  
void main()  
{int a[9][9], i, j;  
for(i=0; i<9; i++)  
    for(j=0; j<9; j++)  
        a[i][j]=(i+1)*(j+1);  
for(i=0; i<9; i++){  
    _____(1)_____  
    printf("%d%d=%d", j+1, i+1, a[i][j]);  
    _____(2)_____  
}  
}
```

1*1=1								
1*2=2	2*2=4							
1*3=3	2*3=6	3*3=9						
1*4=4	2*4=8	3*4=12	4*4=16					
1*5=5	2*5=10	3*5=15	4*5=20	5*5=25				
1*6=6	2*6=12	3*6=18	4*6=24	5*6=30	6*6=36			
1*7=7	2*7=14	3*7=21	4*7=28	5*7=35	6*7=42	7*7=49		
1*8=8	2*8=16	3*8=24	4*8=32	5*8=40	6*8=48	7*8=56	8*8=64	
1*9=9	2*9=18	3*9=27	4*9=36	5*9=45	6*9=54	7*9=63	8*9=72	9*9=81

2. 下面的函数计算并返回年 year、月 month 和日 day 对应的是该年的第几天。

```
int day_of_year(int year, int month, int day)  
{    int leap;  
    leap=(____(3)____) || year %400==0;  
    switch(month){  
        case 12: day +=30;
```

```
case 11: day +=31;  
case 10: day +=30;  
case 9: day +=____(4)____;  
case 8: day +=31;  
case 7: day +=30;  
case 6: day +=31;  
case 5: day +=30;  
case 4: day +=31;  
case 3: day +=____(5)____;  
case 2: day +=31;  
}  
____(6)____;  
}
```

3. 下面函数用“折半查找法”从有 n 个已按从小到大排序的 a 数组中查找关键字 m，若找到，返回其下标值，否则返回-1。算法提示：先确定待查元素的范围，将其分成两半，然后比较位于中间点元素的值。如果该待查元素的值大于中间点元素的值，则将范围重新定义为大于中间点元素的范围，反之亦反。

```
int search(int *a, int n, int m)  
{  int x1=0, x2=n-1, mid;  
    while (x1<=x2){  
        mid = (x1+x2)/2;  
        if (m<*(a+mid)) ____ (7) ____;  
        else if (m>*(a+mid)) ____ (8) ____;  
        else return ( ____ (9) ____ );  
    }  
    return (-1);  
}
```

4. 下面的程序把文本文件 input.txt 中的所有字母修改成另一个字母并输出到文件 output.txt 中。修改的规则是：字母 a 改写成字母 z，字母 A 改写成字母 Z，其它字母改写成该字母的前一个字母。大写字母仍为大写字母，小写字母仍为小写字母，其它字符不做改变。例如：把内容 “An.13zyaZ” 变成内容 “Zm.13yxzY” 。

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

```
#include <stdlib.h>
main()
{ FILE *fp1, *fp2;
char c1, c2;
if ((fp1=fopen("input.txt", "r"))==NULL || ____(10)____) {
    printf("Cannot open files\n");
    exit(0);
}
c1=fgetc(fp1);
while (____(11)____) {
    if (c1=='a' || c1=='A') c2=____(12)____;
    else if(____(13)____) c2=c1-1;
    else c2=c1;
    fputc(c2, fp2);
    ____(14)____;
}
____(15)____
fclose(fp2);
return 0;
}
```

浙江大学 2008-2009 学年春夏学期答案

《C 程序设计基础与实验》课程期末考试试卷

开课学院：计算机学院，考试形式：闭卷，允许带 / 入场

考试时间：2009 年 6 月 21 日，所需时间：120 分钟

试题号	一	二	三	四	总分	
满分	20	30	30	20		
得分					统分人 1	
阅卷人					统分人 2	

试题一、单选题（每小题 2 分，共 20 分）

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

试题二、填空题（每空 2 分，共 30 分）

1 _____ 8 2 _____ 5
3 _____ 0 4 _____ 2
5 _____ #4# 6 _____ 2#4#6#
7 _____ 100 8 _____ *s++=*t++
9 _____ z=0 10 _____ -18

定义了类型 STR，该类型是有 8 个元素

11 _____ 7 12 的数组，而其中每个元素都是字符指针
13 _____ You& 14 _____ 12345
15 _____ is_interesting

试题三、程序阅读题（每小题 5 分，共 30 分）

1 _____ 03#4#7
2 _____ 1#2#3#4#5#6#7#8#9#
3 _____ 6385
4 _____ 51#60#21
5 _____ 21#42#
6 _____ Auilye

试题四、程序（段）填空题（每空 2 分，共 30 分）

- (1) for(j=0; j<=i; j++) (2) printf("\n");
(3) year%4==0&&year%100!=0 (4) 31
(5) 28+leap (或: leap?29:28) (6) return day
(7) x2=mid-1 (8) x1=mid+1

(fp2=open("output.txt", "w")) ==
(9) mid (10) NULL
(11) c1!=EOF 或 !feof(fp1) (12) c1+25 (或 c1+'z'-'a')
(13) c1>'a'&&c1<='z' || c1>'A'&&c1<='Z' (14) c1=fgetc(fp1)
(15) fclose(fp1);

模拟题（一）

一、单项选择题：1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项是最符合题目要求的。请在答题卡上将所选项的字母涂黑。

1. 从一个具有 n 个结点的单链表中查找其值等于 x 结点时，在查找成功的情况下，需平均比较多少个结点？
A. n B. $n/2$ C. $(n-1)/2$ D. $(n+1)/2$
2. 线性表 L 在什么情况下适用于使用链式结构实现？
A. 需经常修改 L 中的结点值 B. 需不断对 L 进行删除插入
C. L 中含有大量的结点 D. L 中结点结构复杂
3. 循环队列用长度为 N 的数组存放其元素值，已知其头尾指针分别是 front 和 rear，则当前队列中的元素个数是
A. $(\text{rear} - \text{front} + N) \% N$ B. $\text{rear} - \text{front} + 1$
C. $\text{rear} - \text{front} - 1$ D. $\text{rear} - \text{front}$
4. 设一个栈的输入序列是 1, 2, 3, 4, 5，则下列序列中，是栈的合法输出序列的是
A. 5 1 2 3 4 B. 4 5 1 3 2
C. 4 3 1 2 5 D. 3 2 1 5 4
5. 三叉树中，度为 1 的结点有 5 个，度为 2 的结点 3 个，度为 3 的结点 2 个，问该树含有几个叶结点？
A. 8 B. 10 C. 12 D. 13
6. 下面关于 m 阶 B 树说法正确的是
A. 每个结点至少有两棵非空子树
B. 树中每个结点至多有 $m-1$ 个关键字
C. 叶子有可能不在同一层上
D. 当插入一个数据项引起 B 树结点分裂后，树长高一层
7. 对于一个共有 N 个结点、 K 条边的森林，共有几棵树？
A. $N-K$ B. $N-K+1$ C. $N-K-1$ D. 不能确定
8. 有 m 个叶子结点的哈夫曼树所具有的结点数为
A. m B. $m+1$ C. $2m$ D. $2m-1$
9. 判断有向图是否存在回路，除了可以利用拓扑排序方法外，还可以利用
A. 求关键路径的方法 B. 求最短路径的 Dijkstra 方法
C. 深度优先遍历算法 D. 广度优先遍历算法
10. 在基于排序码比较的排序算法中，哪种算法的最坏情况下的时间复杂度不高于 $O(N \log_2 N)$ ？
A. 冒泡排序 B. 希尔排序 C. 归并排序 D. 快速排序

二、综合应用题

41. (15 分) 已知一棵二叉树采用二叉链表存储，其结点构造为

LeftChild	Data	RightChild
-----------	------	------------

现定义二叉树中结点 X_0 的根路径为从 X_0 到根结点的一条路径，请编写算法输出该二叉树中最长的根路径（多条最长根路径中只输出一条即可）。

42. (10分) 将100000个数字按升序排列,有一个数字排错,如何纠错?请写出效率最高的方法。两个数字呢?

模拟题 (二)

一、单项选择题:1~40小题,每小题2分,共80分。下列每题给出的四个选项中,只有一个选项是最符合题目要求的。请在答题卡上将所选项的字母涂黑。

1. 带头结点的单链表 first 为空的判定条件是

- A. first == NULL; B. first->link == NULL
C. first->link == first D. first != NULL

2. 以二叉链表作为二叉树存储结构,在具有 n 结点的二叉链表中($n > 0$),空链域的个数为:

- A. $2n - 1$ B. $n - 1$ C. $n + 1$ D. $2n + 1$

3. 对二叉排序树进行什么遍历可以得到从小到大的排序序列?

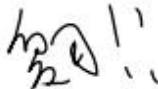
- A. 前序遍历 B. 中序遍历 C. 后序遍历 D. 层次遍历

4. 若 AVL 树的深度是 6,则该树的最少结点数是?(空树的深度定义为 -1)

- A. 13 B. 17 C. 20 D. 33

5. 用 DFS 遍历一个无环有向图,并在 DFS 算法退栈返回时打印相应的顶点,则输出的顶点序列是?

- A. 逆拓扑有序 B. 拓扑有序
C. 无序的 D. 以上都不对



6. 设有一组记录的关键字为 {19, 14, 23, 1, 68, 20, 84, 27, 55, 11, 10, 79},用链地址法构造散列表,散列函数为 $h(key) = key \% 13$,散列地址为 1 的链中有几个记录?

- A. 1 B. 2 C. 3 D. 4

7. 设有 100 个元素的有序序列,如果用折半插入排序再插入一个元素,则最大比较次数是

- A. 25 B. 50 C. 10 D. 7

8. 若用冒泡排序方法对序列 {10, 14, 26, 29, 41, 52} 从大到小排序,需进行多少次比较?

- A. 3 B. 10 C. 15 D. 25

9. 下面四种排序算法中,稳定的算法是

- A. 快速排序 B. 归并排序 C. 堆排序 D. 希尔排序

10. 结点中序遍历为 x,y,z 的不同二叉树,它有几种不同状态?

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

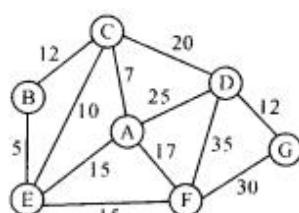
二、综合应用题

41. (10分) 右图中每个顶点表示一个岛,每条边表示岛屿间建桥的成本,找到一个算法计算正确的造桥方法,使得所有的岛屿都能连通,并且总的造价最小。

(1) 给出这个算法的描述,并说明这个算法的名字;

(2) 给出针对右图例子的求解结果。

42. (15分) 已知两个栈 S1 和 S2 的相关操作为:



`Push(X, S)`: 将元素 X 压入栈 S;

`Pop(S)`: 将栈 S 的栈顶元素退栈;

`IsFull(S)`: 判断栈 S 是否已满, 若满则返回 1, 否则返回 0;

`IsEmpty(S)`: 判断栈 S 是否为空, 若空则返回 1, 否则返回 0。

现想用这两个栈实现一个队列, 请用上述栈操作实现队列的入列(Enqueue)和出列(Dequeue)功能。

模拟题 (三)

一、单项选择题: 1~40 小题, 每小题 2 分, 共 80 分。下列每题给出的四个选项中, 只有一个选项是最符合题目要求的。请在答题卡上将所选项的字母涂黑。

1. 在具有 N 个结点的单链表中, 实现下列哪个操作, 其算法的时间复杂度是 $O(N)$?
 - A. 遍历链表和求链表的第 i 个结点
 - B. 在地址为 p 的结点之后插入一个结点
 - C. 删除开始结点
 - D. 删除地址为 p 的结点的后继结点
2. 有六个元素以 6, 5, 4, 3, 2, 1 的顺序进栈, 问下列哪一个不是合法的出栈序列?
 - A. 5 4 3 6 1 2
 - B. 4 5 3 1 2 6
 - C. 3 4 6 5 2 1
 - D. 2 3 4 1 5 6
3. 在一棵高度为 h (假定树根结点的层号为 0) 的完全二叉树中, 所含结点个数不小于
 - A. 2^{h-1}
 - B. 2^{h+1}
 - C. $2^h - 1$
 - D. 2^h
4. 设每个 d 叉树的结点有 d 个指针指向子树, 有 n 个结点的 d 叉树有多少空链域?
 - A. $n(d-1)$
 - B. $n(d-1)+1$
 - C. nd
 - D. 以上都不是
5. 设一段文本中包含字符 {a, b, c, d, e}, 其出现频率相应为 {3, 2, 5, 1, 1}, 则经过哈夫曼编码后, 文本所占字节数为
 - A. 40
 - B. 36
 - C. 25
 - D. 12
6. 从邻接矩阵 $A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ 可以看出, 该图共有()个顶点; 如果是有向图, 该图共有()条弧; 如果是无向图, 则共有()条边。
 - A. 9, 5, 5
 - B. 3, 4, 2
 - C. 6, 3, 3
 - D. 1, 2, 2
7. 具有 n 个顶点的无向图最多有几条边?
 - A. $n(n-1)/2$
 - B. $n(n+1)/2$
 - C. $n^2/2$
 - D. $2n$
8. 在 AOE 网中, 什么是关键路径?
 - A. 短回路
 - B. 第一个事件到最后一个事件的最短路径
 - C. 长回路
 - D. 第一个事件到最后一个事件的最长路径

9. 散列表的地址区间为 $0 \sim 16$, 散列函数为 $H(K) = K \% 17$ 。采用线性探测法处理冲突, 并将关键字序列 $26, 25, 72, 38, 8, 18, 59$ 依次存储到散列表中。元素 59 存放在散列表中的地址是 ()

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

10. 下列排序算法中, 时间复杂度不受数据初始状态影响, 恒为 $O(N \log_2 N)$ 的是

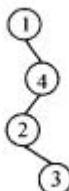
- A. 堆排序 B. 冒泡排序 C. 直接选择排序 D. 快速排序

三、综合分析题

41. (10 分) 某省调查城镇交通状况, 得到现有高等级道路统计表, 表中列出了每条高等级道路直接连通的城镇。省政府“畅通工程”的目标是使全省任何两个城镇间都可通过高等级道路实现交通(但不一定有直接的高等级道路相连, 只要互相间可通过若干条高等级道路接通即可), 并要求增设的高等级道路条数为最少。要解决这个问题, 问:

- (1) 可用什么数据结构表示城镇和高等级道路?
(2) 请用伪码描述效率最高的解法, 要求返回所要增加的高等级道路数目, 不要求具体的路。

42. (15 分) 给定一个整数 X , 以及一个可能的查找的关键字序列 $|K[0], \dots, K[N-1]|$, 请设计算法判别一个序列是否是一个可能的二叉排序树上进行的查找序列。(例如: $|1, 4, 2, 3|$ 就是查找 3 的序列, 对应二叉排序树如右图所示, 而 $|2, 4, 1, 3|$ 就不可能是。) 要求算法时间复杂度为 $O(N)$ 。



模拟题（四）

一、单项选择题：1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项是最符合题目要求的。请在答题卡上将所选项的字母涂黑。

1. 某线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，则采用什么存储方式最节省运算时间？
A. 单链表 B. 仅有头指针的单循环链表
C. 双链表 D. 仅有尾指针的单循环链表
2. 令 P 代表入栈，O 代表出栈。则将一个字符串 $a * b / c$ 变为 $a * b \cdot c / +$ 的栈操作序列是哪个？（例如将 ABC 变成 BCA 的操作序列是 PPOPOO。）
A. PPPOOOPPOPOOO B. POPOPOPPOPOOO
C. POPOOPPOPOOO D. POPPOOPPOPOO
3. 线性表若采用链式存储结构，要求内存中可用存储单元的地址：
A. 必须是连续的 B. 部分地址必须是连续的
C. 一定是不连续的 D. 连续或不连续都可以
4. 设森林 F 中有三棵树，第一、第二、第三棵树的结点个数分别为 M_1, M_2 和 M_3 。与森林 F 对应的二叉树根结点的右子树上的结点个数是
A. M_1 B. $M_1 + M_2$ C. M_3 D. $M_2 + M_3$
5. 哪种树，树中任何结点到根结点路径上的各结点值是有序的？
A. 堆 B. 二叉排序树 C. 完全二叉树 D. 以上都不是
6. 在有 N 个结点且为完全二叉树的二叉排序树中查找一个键值，其平均比较次数的数量级为
A. $O(\log_2 N)$ B. $O(N)$ C. $O(N \log_2 N)$ D. $O(N^2)$
7. 任何一个带权无向连通图的最小生成树
A. 是唯一的 B. 是不唯一的 C. 有可能不唯一 D. 有可能不存在
8. 给出关键字序列 {321, 156, 57, 46, 28, 7, 331, 33, 34, 63}，下面哪个选择是按 LSD 链式基数排序进行了一趟分配和收集的结果？
A. →331→321→33→63→34→156→46→57→7→28
B. →156→28→321→331→33→34→46→57→63→7
C. →321→331→33→63→34→156→46→57→7→28
D. →57→46→28→7→33→34→63→156→321→331
9. 下面关于哈希查找的说法正确的是
A. 哈希函数构造的越复杂越好，因为这样随机性好，冲突小
B. 除留余数法是所有哈希函数中最好的
C. 不存在特别好与坏的哈希函数，要视情况而定
D. 若需在哈希表中删去一个元素，不管用何种方法解决冲突都只要简单地将该元素删去即可
10. 栈和队列的共同特点是
A. 都是先进后出 B. 都是先进先出
C. 只允许在同一端点处插入和删除 D. 没有共同点

二、综合分析题

41. (10 分) 带权图（权值非负，表示边连接的两顶点间的距离）的最短路径问题是找出从初始顶点到目标顶点之间的一条最短路径。若最短路径不止一条，在找到一条最短路径的同时，还需要输出不同的最短路径的条数。现有一种解决该问题的方法：

- (1) 初始化结点集合 S 为仅包含源结点 s ; 用一个整型数组 $\text{Counter}[v]$ 来记录从 s 到结点 v 的最短路径的条数, 各结点 Counter 的初始值均为 0。
- (2) 从未加入 S 的结点中选择当前距离最小的结点 v (“当前距离”指从 s 到 v 且仅经过 S 中结点的最短距离), 将其加入 S 。
- (3) 对每个与 v 相邻的结点 w , 若 w 不在 S 内, 检查:
 - ① 若 v 的加入使得 w 的当前距离变小, 则更新 w 的当前距离为 (v, w) 的边长与 v 的当前距离之和, 并且令 $\text{Counter}[w] = 1$;
 - ② 若 v 的加入是 s 到 w 的长度相同的另一条最短路, 则 $\text{Counter}[w]++$ 。
- (4) 重复步骤(2)、(3), 直到所有结点都被收入集合 S 中。
若该方法可行, 请证明之; 否则, 请举例说明。

42. (15 分) 给定一个带有表头结点的单链表, 结点结构为

Data	Next
------	------

。假设该链表只给出了头指针 L , 请设计尽可能高效的算法按 Data 域递增排序。

模拟题 (五)

一、单项选择题: 1 ~ 40 小题, 每小题 2 分, 共 80 分。下列每题给出的四个选项中, 只有一个选项是最符合题目要求的。请在答题卡上将所选项的字母涂黑。

1. 判定一个栈 ST(最多元素为 m_0) 为空的条件是

A. $ST \rightarrow top \neq 0$	B. $ST \rightarrow top = 0$
C. $ST \rightarrow top \neq m_0$	D. $ST \rightarrow top = m_0$
2. 链表不具有的特点是

A. 插入、删除不需要移动元素	B. 可随机访问任一元素
C. 不必事先估计存储空间	D. 所需空间与线性长度成正比
3. 在一个单链表中, 若 p 所指的结点不是最后结点, 在 p 之后插入 s 所指结点, 则执行

A. $s \rightarrow next = p; p \rightarrow next = s;$	B. $s \rightarrow next = p \rightarrow next; p \rightarrow next = s;$
C. $s \rightarrow next = p \rightarrow next; p = s;$	D. $p \rightarrow next = s; s \rightarrow next = p;$
4. 将 5 个字母 ‘oops’ 按此顺序入栈, 则有多少种不同的出栈顺序可以仍然得到 ‘oops’ ?

A. 1	B. 3	C. 5	D. 6
------	------	------	------
5. 已知有向图 $G = (V, E)$, 其中 $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, $E = \{\langle v_1, v_2 \rangle, \langle v_1, v_4 \rangle, \langle v_2, v_6 \rangle, \langle v_3, v_1 \rangle, \langle v_3, v_4 \rangle, \langle v_4, v_5 \rangle, \langle v_5, v_2 \rangle, \langle v_5, v_6 \rangle\}$ 。 G 的拓扑序列是

A. $v_3, v_4, v_1, v_5, v_2, v_6$	B. $v_1, v_3, v_4, v_5, v_2, v_6$
C. $v_3, v_1, v_4, v_5, v_2, v_6$	D. $v_1, v_4, v_3, v_5, v_2, v_6$
6. 一个无向图采用邻接矩阵存储, 该邻接矩阵一定是

A. 对称矩阵	B. 对角矩阵
C. 三角矩阵	D. 稀疏矩阵
7. 设高为 h 的二叉树只有度为 0 和 2 的结点(一个结点的高度为 1), 则此类二叉树的结点数至少为(), 最多为()?

A. $2h - 1 + 1, 2h - 1$	B. $2h, 2h - 1$
C. $2h - 1, 2h - 1$	D. $2h - 1, 2h - 1 - 1$

8. 假定有 k 个关键字互为同义词,若用线性探测法把这 k 个关键字存入散列表中,至少要进行多少次探测?
 A. $k - 1$ B. k C. $k + 1$ D. $k(k + 1)/2$
9. 就排序算法所用的辅助空间而言,堆排序、快速排序、归并排序的关系是
 A. 堆排序 < 快速排序 < 归并排序 B. 堆排序 < 归并排序 < 快速排序
 C. 堆排序 > 归并排序 > 快速排序 D. 堆排序 > 快速排序 > 归并排序
10. 一个对象序列的排序码为 {46, 79, 56, 38, 40, 84}, 采用快速排序(以位于最左位置的对象为基准)而得到的第一次划分结果为:
 A. {38, 46, 79, 56, 40, 84} B. {38, 79, 56, 46, 40, 84}
 C. {40, 38, 46, 56, 79, 84} D. {38, 46, 56, 79, 40, 84}

二、综合应用题:41 ~ 47 小题,共 70 分。请将答案写在答题纸指定位置上。

41. (10 分) 已知一棵二叉树采用二叉链表存储,其结点构造为

LeftChild	Data	RightChild
-----------	------	------------

请设计算法求二叉树中叶结点的个数。

42. (15 分) 请设计尽可能高效的算法,颠倒一个英文句子中单词的顺序,例如将“this is a test”转换为“test a is this”。设该句子以字符数组方式存放在 S 中,并已知其长度为 N 。

模拟题 (六)

一、单项选择题:1 ~ 40 小题,每小题 2 分,共 80 分。下列每题给出的四个选项中,只有一个选项是最符合题目要求的。请在答题卡上将所选项的字母涂黑。

1. 若某表最常用的操作是在最后一个结点之后插入一个结点或删除最后一个结点。则采用哪种存储方式最节省运算时间?
 A. 单链表 B. 双链表
 C. 单循环链表 D. 带头结点的双循环链表
2. 若已知一个栈的入栈序列是 1, 2, 3, ..., n , 其输出序列为 $p_1, p_2, p_3, \dots, p_n$ 。若 $p_1 = n$, 则 p_i 为:
 A. i B. $n - i$ C. $n - i + 1$ D. 不确定
3. 树最适合用来表示
 A. 有序数据元素 B. 无序数据元素
 C. 元素之间无联系的数据 D. 元素之间有分支层次关系
4. 127 阶 B - 树中除根结点外所有非终端结点至少有多少棵子树?
 A. 2 B. 64 C. 126 D. 63
5. 图的广度优先搜索类似于树的
 A. 前序遍历 B. 中序遍历 C. 后序遍历 D. 层次遍历
6. 无向图 $G = (V, E)$, 其中: $V = \{a, b, c, d, e, f\}$, $E = \{(a, b), (a, e), (a, c), (b, e), (c, f), (f, d), (e, d)\}$, 对该图进行深度优先遍历, 得到的顶点序列正确的是
 A. a, b, e, c, d, f B. a, c, f, e, b, d C. a, e, b, c, f, d D. a, e, d, f, c, b
7. 判定一个有向图是否存在回路,除了拓扑排序,还可以用
 A. 图的遍历 B. 求最小生成树 C. 最短路径 D. 求关键路径

8. 以下命题正确的是

- A. 对于带权无向图 $G = (V, E)$, M 是 G 的最小生成树, 则 M 中任意两点 V_1 到 V_2 的路径一定是它们之间的最短路径
- B. P 是顶点 s 到 t 的最短路径, 如果该图中的所有路径的权值都加 1, P 仍然是 s 到 t 的最短路径
- C. 深度优先遍历也可用于完成拓扑排序
- D. 以上都不是

9. 排序方法中, 从未排序序列中依次取出元素与已排序序列中的元素进行比较, 将其放入已排序序列的正确位置上的方法称为

- A. 插入排序
- B. 选择排序
- C. 快速排序
- D. 归并排序

10. 有组记录的排序码为 $(46, 79, 56, 38, 40, 84)$, 则利用堆排序的方法建立的初始堆为

- A. 79, 46, 56, 38, 40, 80
- B. 84, 79, 56, 38, 40, 46
- C. 84, 79, 56, 46, 40, 38
- D. 84, 56, 79, 40, 46, 38

二、综合应用题: 41 ~ 47 小题, 共 70 分。请将答案写在答题纸指定位置上。

41. (10 分) 在一散列查找中, 设散列表大小为 11, 散列函数为 $H(x) = x \% 11$ 。现要把 1, 13, 12, 34, 38, 33, 27, 22 插入到散列表中, 冲突解决策略为线性探测 (Linear Probing)。请写出:

- (1) 上述数据插入后的散列表状态;
- (2) 在上述散列表状态下, 进行等概率成功与不成功查找的平均查找次数。

42. (15 分) 给定一棵用二叉链表存储的二叉树, 其结点构造为

LeftChild	Data	RightChild
-----------	------	------------

请设计算法判断其是否为完全二叉树。

模拟题 (七)

一、单项选择题: 1 ~ 40 小题, 每小题 2 分, 共 80 分。下列每题给出的四个选项中, 只有一个选项是最符合题目要求的。请在答题卡上将所选项的字母涂黑。

1. 一棵完全二叉树共有 2435 个结点, 则其中度为 0 的结点数为

- A. 1218
- B. 1217
- C. 不确定
- D. 812

2. 若某线性表最常用的操作是存取任一指定序号的元素和在最后进行插入和删除运算, 则利用哪种存储方式最节省时间?

- A. 顺序表
- B. 双链表
- C. 单循环链表
- D. 带头结点的双循环链表

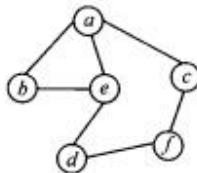
3. 在任意一棵二叉树中, 任何两个叶结点在先序、中序、后序遍历中的相对次序

- A. 不变
- B. 不一样
- C. 不能确定
- D. 以上都不是

4. 对于一个具有 N 个结点的单链表, 在给定值为 x 的结点后插入一个新结点的时间复杂度为

- A. $O(N)$
- B. $O(1)$
- C. $O(N/2)$
- D. $O(N^2)$

5. 设森林 F 对应的二叉树为 B , 它有 m 个结点。 B 的根为 p , p 的右子树结点个数为 n , 森林 F 中第一棵树的结点个数是
 A. $m - n$ B. $m - n - 1$ C. $n + 1$ D. 无法确定
6. 对于有向图, 其邻接矩阵表示比邻接表表示更易于
 A. 求一个顶点的入度 B. 求一个顶点的出边邻接点
 C. 进行图的深度优先遍历 D. 进行图的广度优先遍历
7. 一个有 n 个顶点的强连通图至少有多少条边?
 A. $n(n - 1)$ B. $n - 1$ C. n D. $n + 1$
8. 在图中自 a 点开始进行广度优先搜索算法可能得到的结果为
 A. a, b, e, c, d, f B. a, e, d, f, c, b
 C. a, c, f, e, b, d D. a, e, b, c, f, d
9. 用直接插入排序方法对下面四个序列进行排序(由小到大), 元素比较次数最少的是哪个?
 A. 93, 33, 40, 90, 80, 46, 20, 69 B. 33, 40, 20, 46, 69, 93, 90, 80
 C. 20, 33, 46, 40, 80, 69, 90, 93 D. 90, 69, 80, 46, 20, 33, 93, 40
10. 下列说法不正确的是
 A. 图的遍历是从给定的源点出发每一个顶点仅被访问一次
 B. 遍历的基本算法有两种: 深度遍历和广度遍历
 C. 图的深度遍历不适用于有向图
 D. 图的深度遍历是一个递归过程



二、综合应用题: 41 ~ 47 小题, 共 70 分。请将答案写在答题纸指定位置上。

41. (10 分) 树中任意两结点之间都存在一条路径, 两结点的距离即定义为路径的长度。距离最远的两个结点的距离定义为树的“直径”。给定一棵用二叉链表存储的二叉树, 其结点构造为

LeftChild	Data	RightChild
-----------	------	------------

指针 Root 指向根结点。请设计时间复杂度为 $O(n)$ 的算法(n 为树中结点的个数)求二叉树的直径。

42. (15 分) 已知散列表大小为 11, 散列函数为 $H(x) = x \% 11$, 冲突解决策略为线性探测 (Linear Probing)。给定当前散列表状态如下:

地址	0	1	2	3	4	5	6	7	8	9	10
元素	33	1	13	12	34	38	27	22	32		21

请写出:

- 一个可能的元素插入序列(当有多种插入选择时, 总是选择最小的候选数字);
- 针对上述给定的散列函数和冲突解决方法设计算法, 读入任意散列表状态, 可给出一个可能的元素插入序列。

模拟题 (八)

一、单项选择题：1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项是最符合题目要求的。请在答题卡上将所选项的字母涂黑。

1. 在 n 个结点的顺序表中，算法的时间复杂度是 $O(1)$ 的操作是：
 - A. 访问第 i 个结点 ($1 \leq i \leq n$) 和求第 i 个结点的直接前驱 ($2 \leq i \leq n$)
 - B. 在第 i 个结点后插入一个新结点 ($1 \leq i \leq n$)
 - C. 删除第 i 个结点 ($1 \leq i \leq n$)
 - D. 将 n 个结点从小到大排序
2. 一棵树有 n_1 个孩子数为 1 的结点， n_2 个孩子数为 2 的结点， \dots ， n_m 个孩子数为 m 的结点，则该树的叶结点数为
 - A. $1 + \sum_{i=2}^m (i-1)n_i$
 - B. $1 + \sum_{i=2}^m (i)n_i$
 - C. $\sum_{i=2}^m (i-1)n_i$
 - D. $1 + \sum_{i=1}^{m-1} (i-1)n_i$
3. 已知一棵二叉树的前序遍历结果为 ABCDEF，中序遍历结果为 CBAEDF，则后序遍历的结果为
 - A. CBEFDA
 - B. FEDCBA
 - C. CBEDFA
 - D. 不定
4. 某二叉树的中序序列和后序序列正好相反，则该二叉树一定是
 - A. 空或只有一个结点
 - B. 任一结点无右孩子
 - C. 任一结点无左孩子
 - D. 不确定
5. 在一个图中，所有顶点的度数之和等于所有边数的多少倍？
 - A. 1/2
 - B. 1
 - C. 2
 - D. 4
6. 设哈希表长 $m = 14$ ，哈希函数 $H(\text{key}) = \text{key} \% 11$ 。表中已有 4 个结点： $\text{addr}(15) = 4$ ， $\text{addr}(38) = 5$ ， $\text{addr}(61) = 6$ ， $\text{addr}(84) = 7$ ，其余地址为空。如果用二次探测再散列处理冲突，关键字为 49 的结点的地址是
 - A. 8
 - B. 3
 - C. 5
 - D. 9
7. 在内部排序中，排序不稳定的有
 - A. 插入排序
 - B. 冒泡排序
 - C. 快速排序
 - D. 归并排序
8. 对于序列 (49, 38, 65, 97, 76, 13, 27, 50)，按由小到大进行排序，下面哪一个是初始步长 $d = 4$ 的希尔排序法第一趟的结果？
 - A. 49, 76, 65, 13, 27, 50, 97, 38
 - B. 13, 27, 38, 49, 50, 65, 76, 97
 - C. 97, 76, 65, 50, 49, 38, 27, 13
 - D. 49, 13, 27, 50, 76, 38, 65, 97
9. 在单链表中，指针 p 指向元素为 x 的结点，实现“删除 x 的后继”的语句是
 - A. $p = p \rightarrow \text{next};$
 - B. $p \rightarrow \text{next} = p \rightarrow \text{next} \rightarrow \text{next};$
 - C. $p \rightarrow \text{next} = p;$
 - D. $p = p \rightarrow \text{next} \rightarrow \text{next};$
10. 设有关键码序列 $(q, g, m, z, a, n, p, x, h)$ ，下面哪一个序列是从上述序列出发按字典顺序建堆的结果？
 - A. $a, g, h, m, n, p, q, x, z$
 - B. $a, g, m, h, q, n, p, x, z$
 - C. $g, m, q, a, n, p, x, h, z$
 - D. $h, g, m, p, a, n, q, x, z$

二、综合应用题:41 ~ 47 小题,共 70 分。请将答案写在答题纸指定位置上。

41. (15 分) 给定 N 个村庄之间的交通图,若村庄 i 和 j 之间有道路,则将顶点 i 和 j 用边连接,边上的 W_{ij} 表示这条道路的长度。现在要从这 N 个村庄中选择一个村庄建一所医院,问这所医院应建在哪个村庄,才能使离医院最远的村庄到医院的路程最短? 请设计算法解决此问题,且算法复杂度不超过 $O(N^3)$ 。
42. (10 分) 给定 2 个带有表头结点的单链表的头指针 L_1 和 L_2 ,结点结构为

Data	Next
------	------

。假设这 2 个链表的结点已经按 Data 域递增有序,请设计算法把它们合并成一个按 Data 域递增有序的链表。注意:算法不能使用额外的结点空间。

模拟题（九）

一、单项选择题：1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项是最符合题目要求的。请在答题卡上将所选项的字母涂黑。

1. 设稀疏矩阵 A 有 t 个非零元素，用二维数组存储时占用 $m \times n$ 个单元。问稀疏矩阵的三元组表示法在什么情况下比二维数组存储节省空间？（设行号、列号及元素值占用空间相同）
A. $t < m \times n$ B. $t < m \times n/3$
C. $t < m \times n/3 - 1$ D. $t < m \times n - 1$
2. 将线性表 La 和 Lb 头尾连接，要求时间复杂度为 $O(1)$ ，且占用辅助空间尽量小。应该使用哪种结构？
A. 单链表 B. 普通单循环链表
C. 带尾指针的单循环链表 D. 带头结点的双循环链表
3. 在下述结论中，正确的是
①只有一个结点的二叉树的度为 0；
②二叉树的度为 2；
③二叉树的左右子树可任意交换；
④深度为 K 的完全二叉树的结点个数小于或等于深度相同的满二叉树。
A. ①②③ B. ②③④ C. ②④ D. ①④
4. 前序遍历和中序遍历结果相同的二叉树为
A. 根结点无右孩子的二叉树 B. 根结点无左孩子的二叉树
C. 所有结点只有左子树的二叉树 D. 所有结点只有右子树的二叉树
5. 以顺序表为存储结构，在 300000 个任意排序的数据中选择前 100 个最大值，下列哪种算法最快？
A. 快速排序 B. 堆排序
C. 归并排序 D. 插入排序

6. 在下列查找的方法中，平均查找长度与结点个数 n 最无关的查找方法是
A. 顺序查找 B. 二分法
C. 利用二叉搜索树 D. 利用哈希(hash)表
7. 在哈希查找中，采用线性探查法解决冲突时所产生的一系列后继散列地址
A. 必须大于等于原散列地址 B. 必须小于等于原散列地址
C. 可以大于或小于但不等于原散列地址 D. 对地址在何处没有限制
8. 使用递归的归并排序算法时，为了保证排序过程的时间复杂度不超过 $O(N \log_2 N)$ ，必须做到
A. 每次序列的划分应该在线性时间内完成
B. 每次归并的两个子序列长度接近
C. 每次归并在线性时间内完成
D. 以上全是
9. 设 n, m 为一棵二叉树上的两个结点，在中序遍历时， n 在 m 前的条件是
A. n 在 m 右方 B. n 是 m 祖先
C. n 在 m 左方 D. n 是 m 子孙
10. 若一个栈的输出序列为 $1, 2, 3, \dots, n$ ，输出序列的第一个元素是 i ，则第 j 个输出元素是
A. $i - j - 1$ B. $i - j$ C. $j - i - 1$ D. 不确定的

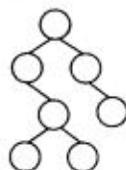
二、综合应用题:41~47 小题,共 70 分。请将答案写在答题纸指定位置上。

- 41.(10分)弗洛伊德(Floyd)算法是求图中每一对结点之间最短路径的算法。请描述该算法,并且回答:如何用该算法判断图是否有回路?请说明理由。
- 42.(15分)给定 N 个浮点数,请设计一个高性能的算法,从中找出 $M (< N)$ 个最大的数。

模拟题(十)

一、单项选择题:1~40 小题,每小题 2 分,共 80 分。下列每题给出的四个选项中,只有一个选项是最符合题目要求的。请在答题卡上将所选项的字母涂黑。

1. 从一个栈顶指针为 ST 的链栈中删除一个结点时,用 x 保存被删结点的值,则执行
A. $x = ST; ST = ST \rightarrow next;$ B. $x = ST \rightarrow data;$
C. $ST = ST \rightarrow next; x = ST \rightarrow data;$ D. $x = ST \rightarrow data; ST = ST \rightarrow next;$
2. 一个非空二叉树的叶结点的个数(n_0)和有两个分支的结点的个数(n_2)的关系为
A. $n_0 = n_2 - 1$ B. $n_0 = n_2 + 1$
C. $n_2 = n_0 - 1$ D. $n_2 = n_0 + 1$
3. 在线索化二叉树中,t 所指结点没有左子树的充要条件是
A. $t \rightarrow left = NULL$ B. $t \rightarrow ltag = 1$
C. $t \rightarrow ltag = 1$ 且 $t \rightarrow left \neq NULL$ D. 以上都不对
4. 对线性表进行二分查找时,要求线性表必须
A. 以顺序方式存储 B. 以顺序方式存储,且数据元素有序
C. 以链接方式存储 D. 以链接方式存储,且数据元素有序
5. 已知一棵 7 个结点的二叉排序树结构如右图所示,其中各结点的值是 1~7 之间的整数。假设没有两个结点的值相同,则对此树进行后序遍历的结果是
A. 1234567 B. 6742531 C. 5132467 D. 2431765
6. 给定一个整数序列{2 6 8 4 5 3 1 7},下面哪个排序方法中需要非常数级的额外存储空间?
A. 插入排序 B. 希尔排序 C. 归并排序 D. 堆排序
7. 高度为 3 的二叉树有多少种?(设一个结点的高度为 1)
A. 20 B. 22 C. 19 D. 21
8. 在有向图的邻接矩阵表示中,计算第 i 个顶点入度的方法是
A. 第 i 行非零元素个数 B. 第 i 列非零元素个数
C. 第 i 行零元素个数 D. 第 i 列零元素个数
9. 无向图有 15 条边,有 3 个 6 度顶点,2 个 3 度顶点,其余顶点的度数都小于 3,则该图中至少有多少个顶点?
A. 7 B. 8 C. 10 D. 6
10. 已知一棵二叉树的后序遍历结果为 EDBFCCA,中序遍历结果为 EDBAFCG,则前序遍历的结果为
A. ABDECFC B. FEDCGBA C. CBEDFGA D. 不定



二、综合应用题:41 ~ 47 小题,共 70 分。请将答案写在答题纸指定位置上。

41. (15 分)已知一棵二叉排序树采用二叉链表存储,其结点构造为

LeftChild	Data	RightChild
-----------	------	------------

给定树中两个结点的值 V_1 和 V_2 ,请设计一个尽可能高效的算法,找到它们的最近公共祖先。可以假定这两个值都在这棵树中。

42. (10 分)给定一个字符串 S ,包含有大写或小写的英文字母,请设计尽可能高效的算法把小写字母移动到字符串的头部、大写字母移到尾部。要求额外辅助空间为 $O(1)$ 。

模拟试题(一)参考答案及解析

一、单项选择题:1~40 小题,每小题 2 分,共 80 分

1.【答案】D

【分析】在一个单链表中查找其值等于 x 结点时,从单链表的头上开始逐个往后查找,如找到的是第 i 个记录,则要比较 i 次。在等概率情况下,所需平均比较的次数是 $\sum_{i=1}^n i/n = (n+1)/2$ 。

2.【答案】B

【分析】链式结构在进行插入和删除时,不需要对其他结点进行操作,而如果是数组的话,在做插入和删除时还需要对其他结点进行移动。

3.【答案】A

【分析】循环队列中只要 front 和 rear 到达数组的尾端,就又绕回到开头。所以答案是 A。

4.【答案】D

【分析】这个题是考查对栈的基本操作是否掌握。栈必须符合先进后出的原则。所以 5 如果是最后出的,那么后面那个数不可能是 1。B 答案后出的是 4 和 5,后面也不可能 1。如此推论答案 D 是对的。

5.【答案】A

【分析】一棵树的边数等于结点个数减 1,从题目中可得到该三叉树的度为 1、2、3 的结点总数是 $5+3+2=10$,从度中又可以得到该二叉树的结点总数是 $1+5+6+6=18$,所以,叶结点数是 8 个。

6.【答案】B

【分析】 m 阶 B 树是所有的树叶都在相同的深度上,树的根或者是一片叶子或者其儿子数在 2 和 m 之间,除根外,所有非树叶结点的儿子数在 $\lceil m/2 \rceil$ 和 m 之间,如有 m 个儿子,则关键字有 $m-1$ 个。当插入一个数据项引起 B 树结点分裂时,不一定会引起树长高一层。所以,答案是 B。

7.【答案】A

【分析】一棵树中结点数 N 和边数 K 的关系是 $N=K+1$ 。所以,如果有 m 棵树的话,则 $N_1=K_1+1$, $N_2=K_2+1, \dots, N_m=K_m+1$ 。将 m 个等式两边分别相加,得到 $N=K+m$ (则 $m=N-K$),于是所有的结点之和与边之和的差值就是树的棵数。

8.【答案】D

【分析】对任何一棵二叉树,如果其叶结点数为 n_0 ,度为 2 的结点数为 n_2 ,则 $n_0=n_2+1$ 。哈夫曼树也是二叉树,且只有度为 0 和度为 2 的结点。所以若 $n_0=m$,则 $n_2=m-1$,结点数总数为 $2m-1$ 。

9.【答案】C

【分析】深度优先遍历图,用数组 $visited[]$ 标记已遍历过的结点。如 vi 是图结点, $visited[vi]$ 为 1,表示 vi 这个结点已经遍历过了, $visited[vi]$ 为 0,表示未遍历过。如果图中存在环,则某个结点必然会被遍历两次。而广度优先遍历算法即使某个结点遍历两次,也不能判定有回路。A、B 的选择也不可能。

10.【答案】C

【分析】快速排序算法最坏情况下的时间复杂度为 $O(N^2)$,而平均时间复杂度为 $O(N \log_2 N)$ 。冒泡排序最坏情况下时间复杂度是 $O(N^2)$,希尔排序的最坏情况下运行时间是 $O(N^2)$,归并排序的平均时间和最坏时间复杂度都是 $O(N \log_2 N)$ 。

二、综合应用题:41~47 小题,共 70 分

41.【参考答案】

```
typedef struct TreeNode *Tree;
typedef struct TreeNode {
    Tree LeftChild;
    ElementType Data;
    Tree RightChild;
};

void BinaryTreeDepth( Tree T, Tree *NodeP, int *MaxDepthP, int Depth )
/* T 为待搜索的树(或子树)根,NodeP 是当前最深结点指针的地址,MaxDepthP 为保留当前最深
   长度的变量地址,Depth 为当前树的深度 */
{
    if ( !T ) return;
    if ( Depth + 1 > (*MaxDepthP) ) {
        (*MaxDepthP) = Depth + 1;
        (*NodeP) = T;
    }
    BinaryTreeDepth( T->LeftChild, NodeP, MaxDepthP, Depth + 1 );
    BinaryTreeDepth( T->RightChild, NodeP, MaxDepthP, Depth + 1 );
    return;
}

void PrintRootPath( Tree T, Tree Node, bool *FlagP )
/* T 为待搜索的树(或子树)根,Node 为当前最深结点的指针,FlagP 为是否找到最深结点的标记变
   量地址 */
{
    if ( !T ) return;
    if ( T == Node ) (*FlagP) = 1;
    PrintRootPath( T->LeftChild, Node, FlagP );
    if ( *FlagP ) | Output( T->Data ); return; |
    PrintRootPath( T->RightChild, Node, FlagP );
    if ( *FlagP ) | Output( T->Data ); return; |
}

void PrintLongestRootPath( Tree Root )
{
    Tree Node = Root;
    int MaxDepth = 0;
    bool Flag = 0;
    if ( !Root ) return;
    BinaryTreeDepth( Root, &Node, &MaxDepth, 0 );
    PrintRootPath( Root, Node, &Flag );
    return;
}
```

【分析】

此题要求最长的根路径,该路径的长度就是树的深度。解题思路分两步走:首先,用函数 `BinaryTreeDepth` 求树的深度,同时用 `Node`(其相应地址指针为 `Nodep`)记住最深的那个叶子结点;然后,调用函数 `PrintRootPath` 对树做先序遍历,在遍历过程中找 `Node`。当找到 `Node` 时,存在系统栈中的结点和 `Node` 就构成了最长的根路径。整个算法在最坏情况下会执行两次树的遍历,复杂度是 $O(N)$ (N 为树中结点的个数)。

另外有算法用一个数组保存路径上的结点，在求树深的同时不断更新最深路径，虽然程序结构相对简单，但复杂度显然超过线性。

42.【参考答案】

当数列基本有序时，插入排序是最快的算法。插入排序的时间复杂度是 $O(I + N)$ ，其中 N 是元素个数， I 是序列中逆序对的个数。

当只有一个数字错位时， I 最坏情况下等于 $N - 1$ ，所以用插入排序的时间复杂度是线性的。有 k 个数字错位时，最坏情况下有 $I = O(kN)$ 。

```
void InsertionSort( ElementType A[], int N )
{
    int i, P;
    ElementType Tmp;
    for ( P = 1; P < N; P++ ) {
        Tmp = A[ P ];
        for ( i = P; i > 0 && (A[ i - 1 ] > Tmp); i-- )
            A[ i ] = A[ i - 1 ];
        A[ i ] = Tmp;
    }
}
```

【分析】

这里需要注意的是，如果题目中没有指定数列是按升序还是按降序排列，我们必须用一个函数 IsIncrease 来判断数列是否按升序排列。当已知有 k 个数字错位时，我们只需要比较 $2k + 1$ 对相邻元素的相对大小，其中至多有 k 对元素的相对大小是错的，另外至少 $k + 1$ 对相邻元素的相对大小是一致的。我们用计数器加 1 表示递增，减 1 表示递减，则做了 $2k + 1$ 次比较后，若计数器是正的，则数列必定按升序排列，否则按降序排列。在假设 k 比元素个数 N 小很多 ($2k + 1 < N$) 的前提下，我们可以这样实现 IsIncrease 函数：

```
bool IsIncrease( ElementType A[], int k )
/* k 是排错的数字个数 */
{
    int n = 2 * k + 1;
    int cnt = 0;
    int i;

    for ( i = 0; i < n; i++ ) /* 比较  $2k + 1$  对相邻元素的相对大小 */
        if ( A[ i ] < A[ i + 1 ] ) cnt++;
        else cnt--;
    if ( cnt > 0 ) return TRUE;
    else return FALSE;
}
```

则在插入排序算法中，我们应先获得数列递增还是递减的信息，例如可用

```
bool Flag = IsIncrease( A, k );
```

来记录。程序中比较元素的条件 ($A[i - 1] > Tmp$) 应被替换为：

```
((Flag && A[ i - 1 ] > Tmp) || (!Flag && A[ i - 1 ] < Tmp))
```

模拟试题(二)参考答案及解析

一、单项选择题:1~40 小题,每小题 2 分,共 80 分。

1.【答案】B

【分析】带头结点的单链表如果为空的话就只剩下一个头结点 first,其 link 域为空。所以应该是 B。

2.【答案】C

【分析】在有 n 个结点的二叉树中一共有 $2n$ 个链域,而该树一共有 $n - 1$ 条边(即非空链域),所以空链域的个数为 $2n - (n - 1) = n + 1$ 。

3.【答案】B

【分析】二叉排序树中的每一个结点 X ,它的左子树中所有关键字值小于 X 的关键字,而它的右子树中所有关键字值大于 X 的关键字。中序遍历可以递归地理解为:输出所有小于 X 的关键字;输出 X ;输出所有大于 X 的关键字,于是得到从小到大的有序序列。

4.【答案】D

【分析】 $N(0) = 1, N(1) = 2, N(n) = N(n - 1) + N(n - 2) + 1$,得 $N(6) = 33$ 。

5.【答案】A

【分析】用 DFS 遍历一个无环有向图,退栈时打印次序刚好和拓扑排序相反。

6.【答案】D

【分析】一共有 4 个值,分别是 14, 1, 27, 79。

7.【答案】D

【分析】最大的比较次数即是二分查找该元素所需要的比较次数,所以是 $\lfloor \log_2 100 \rfloor + 1$ 。

8.【答案】C

【分析】因为该序列是逆序,所以需要比较的次数一共是 $5 + 4 + 3 + 2 + 1 = 15$ 。

9.【答案】B

【分析】稳定的排序算法有直接插入排序、二分法插入排序、归并排序、冒泡排序和基数排序。不稳定的排序算法有快速排序、直接选择排序、希尔排序和堆排序。

10.【答案】C

【分析】实际上考查的是有 3 个结点的不同二叉树一共有多少种,所以是 5 种。

二、综合应用题:41~47 小题,共 70 分

41.【参考答案】

(1)用求最小生成树的克鲁斯卡尔(Kruskal)算法。算法伪码描述如下:

```
void Kruskal( Graph G )
{
    T = | 空集 | ;
    while( T 包含的边少于 |V| - 1 && T 非空集 ) |
        从当前边集合 E 中选出最小造价的边(v, w);
        从当前边集合 E 中删除此边(v, w);
        if( (v, w) 的加入不会在 T 中产生环 )    将(v, w)加入 T ;
        else    抛弃(v, w);
    |
    if( T 中包含的边数少于 |V| - 1 )
        Error( "No spanning tree" );
}
```

(2)根据该算法,要造的桥梁依次选择为:(B, E, 5), (A, C, 7), (C, E, 10), (D, G, 12), (E, F, 15), (C, D, 20)。

【分析】

求最小生成树至少有两种不同的算法。注意到给出的图中有 7 个结点、12 条边，属于边稀疏的情况，用克鲁斯卡尔算法是合适的。若给出一个完全图，则普里姆(Prim)算法将是更好的选择。

42.【参考答案】

```
void Enqueue( ElementType X, Stack S1, Stack S2 )
{
    ElementType Tmp;
    if ( IsFull( S1 ) && !IsEmpty( S2 ) )
        Error("Queue is full!");
    else if( IsFull( S1 ) ) | //S1 满但 S2 空
        while ( !IsEmpty( S1 ) ) |
            Tmp = Pop( S1 ); Push( Tmp, S2 );
    |
    | //至此 S1 不满
    Push( X, S1 );
}

ElementType Dequeue( Stack S1, Stack S2 )
{
    ElementType Tmp;
    if ( IsEmpty( S1 ) && IsEmpty( S2 ) )

        return ErrorElement;
    if ( IsEmpty( S2 ) ) |
        while ( !IsEmpty( S1 ) ) |
            Tmp = Pop( S1 ); Push( Tmp, S2 );
    |
    |
    return Pop( S2 );
}
```

【分析】

栈的特点是后进先出，队列的特点是先进先出。本题的目的是用两个栈 S1 和 S2 模拟一个队列。我们可以利用栈的后进先出特点，经过两次“后进先出”转变为“先进先出”，即元素先经过栈 S1 再经过栈 S2，从而实现队列的功能。

用两个栈 S1 和 S2 模拟一个队列时，入队的原理相对简单，向 S1 中压入即可。问题是当 S1 满时，为了能最大限度不浪费空间，我们应检查 S2。这里假设 S1 和 S2 容量相同。若 S2 是空的，我们可以将 S1 中的元素逐一退栈，并压入 S2 中。新入列的元素就可以压入腾空的 S1 了。只有在 S2 还没有空且 S1 已满的情况下，模拟队列中才能不再接受新元素。

当需要出队时，首先判断模拟队列是否为空。若 S1 和 S2 都是空的，则应返回错误信息。若出队请求是合法的，则检查 S2。若 S2 为空，则 S1 必不为空，将 S1 中的元素逐一退栈，并压入 S2 中。这时 S2 中栈底存的是 S1 的栈顶元素，即最晚入列的元素；栈顶存的是 S1 的栈底元素，即最早入列的元素。这时 S2 退栈就相当于队列的出队，实现了先进先出。

模拟试题(三)参考答案及解析

一、单项选择题:1~40 小题,每小题 2 分,共 80 分

1.【答案】A

【分析】B、C、D 选项都仅需要 $O(1)$ 的时间,A 的操作时间复杂度是 $O(n)$ 。

2.【答案】C

【分析】按照栈的操作,如果 6 不是第 1 个出来,则它不可能在 5 之前出来。如果 3 最先出来,那么栈里的元素从顶往下依次是 4,5,6,所以 C 是不可能的。

3.【答案】D

【分析】一棵完全二叉树的结点数在结点最多时是满二叉树,结点个数为 $2^{h+1} - 1$,结点最少时,第 h 层只有一个结点,其余层都是满的,结点个数为 $2^h - 1 + 1 = 2^h$ 。

4.【答案】B

【分析】每个 d 叉树的结点有 d 个指针指向子树,如果结点数为 n ,则一共有 nd 个指针。而该树的边的条数为 $n-1$,即有 $n-1$ 个指针非空,所以空链域的个数是 $nd - (n-1) = n(d-1) + 1$ 。

5.【答案】C

【分析】根据哈夫曼编码原理,每个字符的编码长度分别是 a:2,b:3,c:1,d:4,e:4,则文本所占字节数为 $2 \times 3 + 3 \times 2 + 1 \times 5 + 4 \times 1 + 4 \times 1 = 25$ 。

6.【答案】B

【分析】邻接矩阵是 3×3 ,所以是 3 个顶点,如果是有向图,共有 4 条弧,如果是无向图,则是 2 条边。

7.【答案】A

【分析】具有 n 个顶点的无向图最多有边的条数为: $(n-1) + (n-2) + \dots + 1 = n(n-1)/2$ 。

8.【答案】D

【分析】这个题主要考查对关键路径的定义是否了解。对于给出的事件结点网络,关键路径上任何一个事件的延长(边权重加大)都将导致从第一个事件到最后一个事件的路径长度变长,所以是从第一个事件到最后一个事件的最长路径。

9.【答案】D

【分析】根据散列表以及线性探测法处理冲突的方法,关键字序列依次存放在 9,8,4,5,10,1,11 的位置上,所以,元素 59 存放的地址为 11。

10.【答案】A

【分析】快速排序算法最坏情况下的时间复杂度为 $O(N^2)$,而平均时间复杂度为 $O(N \log_2 N)$ 。冒泡排序最坏情况下时间复杂度是 $O(N^2)$,直接选择排序的运行时间复杂度是 $O(N^2)$ 。只有堆排序的时间复杂度在任何情况下都是 $O(N \log_2 N)$ 。

二、综合应用题:41~47 小题,共 70 分

41.【参考答案】

(1) 用图结点表示城镇,边表示道路。因为题目的要求与道路长短及造价无关,故应采用无向无权图。实现时可用邻接表存储。

(2) 算法伪码如下:

```
typedef struct GraphNode {
    ElementType Vert;
    struct GraphNode * Next;
} Graph;
```

```

typedef struct GraphNode * NodePtr;
Graph G[ MaxNumOfVertices ];
bool Visited[ MaxNumOfVertices ];

void DFS( Graph G, Vertex V )
| /* 从结点 V 开始对图 G 进行深度优先搜索 */
    Visited[V] = TRUE;
    for ( NodePtr W=G[V]; W; W=W->Next )
        if ( !Visited[W->Vert] )
            DFS(G, W->Vert);
}

int FindNumOfRoads( Graph G, int N )
| /* N 为图 G 的结点数 */
    int cnt = 0;
    for ( Vertex V=0; V<N; V++ ) Visited[V] = FALSE;
    for ( Vertex V=0; V<N; V++ ) |
        if ( !Visited[V] ) |
            DFS( V, G );
            cnt++;
|
|
|
return( cnt - 1 );
}

```

【分析】

题目要求用最少的边使得图连通，实际上就是数出原图中连通子集的个数，则需要添加的边数就是连通子集个数减1。求连通集的标准算法是深度优先遍历，用邻接表存储图可以比较方便地实现。这里省略了图的邻接表表示的具体定义。

42.【参考答案】

```

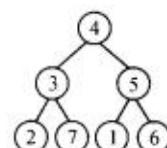
#define IsInRange(n) ((n < max) && (n > min))

bool Is_BST_sequence( int K[], int N, int x )
|
    min = MIN_ELEMENT;  max = MAX_ELEMENT;
    for ( i=0; i < n; i++ ) |
        if ( IsInRange( K[i] ) ) |
            if ( K[i] > x )           max = K[i];
            else if ( K[i] < x )       min = K[i];
            else if ( i == N-1 ) return TURE;
            else return FALSE;
|
        else return FALSE;
|
    return TURE;
|

```

【分析】

此题最直接的算法是先按照给定序列构造一棵二叉树，再判断其是否是二叉排序树，而判断是否二叉排序树时，最容易犯的错误是用一个简单的后序遍历。右图给出了一个反例，即根结点的左右子树都是二叉排序树，但对于根



结点 4 来说,7 和 1 分别在错误的位置。正确的解法是在递归进行后序遍历的同时,检查并记录当前结点的上下界。

参考答案给出了一个更巧妙的解法,充分利用了此问题产生的二叉树每个结点至多只有一个孩子的特点,从根结点开始向下扫描,用 min 和 max 跟踪记录到目前为止可判别的查找范围的上下界。

模拟试题(四)参考答案及解析

一、单项选择题:1~40 小题,每小题 2 分,共 80 分

1.【答案】D

【分析】这个题目 C 和 D 似乎都可以,但注意到要删除第一个元素,我们必须知道它的前驱结点,所以 D 是正确的选择。

2.【答案】C

【分析】这个题目是考查对栈操作的掌握,关键是要符合先进后出的原则。

3.【答案】D

【分析】对于链式存储结构,每一个单元都可以是单独申请的存储空间,所以,可以是连续的,也可以是不连续的。

4.【答案】D

【分析】首先将森林中的每棵树变为二叉树,因为转换所得的二叉树的根结点的右子树均为空,故可将各二叉树的根结点视为兄弟从左至右连在一起,就形成了森林到二叉树的转换。所以对应的二叉树根结点的右子树中包含了除第一棵树以外的两棵树,所以结点个数是 $M_2 + M_3$ 。

5.【答案】A

【分析】这个题是考查对堆、二叉排序树、完全二叉树的了解。完全二叉树对结点的值没有任何限制,二叉排序树要求结点和它的左右孩子分别有序,但这个相对大小的顺序左右不一致,所以根路径不一定有序。符合要求的只有堆,因为堆结构限制了父子结点必须有序,而这个相对大小的顺序在任何一层都是一致的。

6.【答案】A

【分析】注意是“完全二叉树”,所以平均比较次数就是这棵树的高度,应该选 A。

7.【答案】C

【分析】按照最小生成树的产生原理,当有多条权重相同的边存在时,有可能会不唯一。

8.【答案】C

【分析】按照 LSD 基数排序的分配和收集原则,首先会按个位数来排,则 B 和 D 显然不对。根据 A 的前 2 项,我们判断对个位数相同的数字采取了后进先出的方法进行分配和收集(331 后进,但排在 321 之前),但是后面 2 项又与这个判断矛盾(63 后进,却排在 33 之后),所以符合要求的只有 C。

9.【答案】C

【分析】根据哈希查找的概念,哈希函数的好坏要看应用的具体情况来定。对哈希表中的元素做删除操作,不能简单地将该元素删除,否则,就会影响到对其他元素的搜索。

10.【答案】D

【分析】栈是一种数据项按序排列的数据结构,具有先进后出的特点,只能在一端(称为栈顶 top)对数据项进行插入和删除。队列具有先进先出(FIFO)的特点,它只允许在表的前端 front 进行删除操作,而在表的后端 rear 进行插入操作。

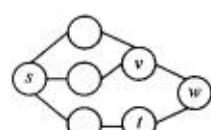
二、综合应用题:41~47 小题,共 70 分

41.【参考答案】

这个算法是错误的。右图给出了反例:设图中各边长为 1,则从 s 到 w 有 3 条长度相同的最短路径,而使用该方法得到的结果是 2。

问题出在 3 个地方:

(1) 当 v 的加入产生新的最短路径时,从 s 到 v 再到 w 的最短路径条数等于 v 的最短路径条数,而不是仅等于 1。所以(3)①中 Counter[w] = 1 应改为 Counter[w] = Counter[v]。



(2) 若 v 的加入是 s 到 w 的长度相同的另一条最短路，则 w 的最短路径条数不是只增加了 1，而是 v 带过来的所有最短路径，即(3)②中的 $\text{Counter}[w]++$ 应改为 $\text{Counter}[w] += \text{Counter}[v]$ 。

(3) 根据前 2 步修改， $\text{Counter}[s]$ 的初始值必须为 1，而不是 0，否则后面所有结点的 Counter 都会一直是 0。

【分析】

此题属于对经典 Dijkstra 算法的推广应用，类似题目还有累计所有长度相同的最短路径上的结点个数等，其实现原理相同。

另一类相似的推广应用是找到包含边数最少的那条最短路径。这时第(3)步应改为：

(3) 对每个与 v 相邻的结点 w ，若 w 不在 S 内，检查：

①若 v 的加入使得 w 的当前距离变小，则更新 w 的当前距离为 (v, w) 的边长与 v 的当前距离之和，更新 $\text{Counter}[w] = \text{Counter}[v] + 1$ ，并且将 v 记录在 w 的路径上。

②若 v 的加入是 s 到 w 的长度相同并且边数更少 ($\text{Counter}[w] > \text{Counter}[v] + 1$) 的另一条最短路，则 $\text{Counter}[w] = \text{Counter}[v] + 1$ ，并且将 v 记录在 w 的路径上。

这里的整型数组 $\text{Counter}[v]$ 记录从 s 到结点 v 的最短路径包含的边数，各结点 Counter 的初始值均为 0。此问题很容易做错的地方，是当发现另一条长度相同但包含边数更少的最短路径时，只更新了 Counter ，却忘了更新 w 的路径。

类似题目还有求包含最多边的最短路径，或者给每个结点赋予价值，求价值最高的最短路径等，其实现原理相同。

42. 【参考答案】

采用冒泡排序法。

```
List BubbleSort( List L )
{
    bool Flag = FALSE;           //若为 TRUE,则表示排序完成
    List Tail;                  //从表尾开始,始终指向最后一个未排序的元素
    List P, Q, Tmp;             //P, Q 前后相邻,同步扫描链表,并在 Tmp 的帮助下交换元素
    if ( ! L->Next ) return L; //L 是空链表
    for ( Tail = L; Tail->Next; Tail = Tail->Next );      //找表尾
    while ( ! Flag && (Tail != L->Next) ) |
        P = L; Q = P->Next;
        Flag = TRUE;          //若此趟不改变任何元素顺序,则排序结束
        for ( ; Q != Tail; P = Q, Q = Q->Next ) |
            if ( Q->Data > Q->Next->Data ) |
                Flag = FALSE;          //需要改变顺序

                Tmp = Q->Next;
                P->Next = Tmp; Q->Next = Tmp->Next; Tmp->Next = Q;
                Q = Tmp;
            |   //end if
        |   //end for
        Tail = P;
    |   //end while
    return L;
}
```

【分析】

选择冒泡排序是因为链表是单向的,而冒泡排序只比较并交换相邻元素,故比较容易实现。作为提高效率的手段,Flag 检查每一趟冒泡过程中是否有元素进行了交换,若整趟冒泡过程中都没有任何元素交换,则说明链表已经有序,就可以直接跳出 while 循环了。

选择排序也可以比较方便地用链表实现,但其复杂度恒为 $O(N^2)$,例如在原始链表有序的情况下,效率显然没有冒泡排序高。

标准的插入排序需要对链表进行逆向查找,不适合处理单链表。当然也可以将逆向查找改为每次从表头开始顺序查找合适的插入位置。但必须注意的是,实现时如果不先与有序子列的末尾元素比较就直接从头扫描,则处理初始有序的序列就有复杂度 $O(N^2)$,这是不能接受的。这样实现的“插入排序”改变了原始插入排序的性质——当序列基本有序时(例如仅有一个最大元错位排到了表头),链表实现的复杂度反而较高,所以不推荐这种方法。无论如何,下面给出了链表实现的程序。

```
List InsertionSort( List L )
{
    List Tail;      //指向当前有序子列的最后一个元素
    List Tmp;       //指向当前待插入的结点
    List P1, P2;    //用于扫描链表

    if( !L->Next ) return L;      //L 是空链表
    Tail = L->Next;
    for ( ; Tail->Next; Tail = Tail->Next ) {
        if ( Tail->Data > Tail->Next->Data ) {
            Tmp = Tail->Next;
            for ( P1 = L, P2 = L->Next;
                  P2->Data < Tmp->Data; P1 = P2, P2 = P2->Next ) ;|
            Tail->Next = Tmp->Next;
            Tmp->Next = P2;   P1->Next = Tmp;           //插入
            if( !Tail->Next ) break;
        }
    }
    return L;
}
```

模拟试题(五)参考答案及解析

一、单项选择题:1~40 小题,每小题 2 分,共 80 分

1.【答案】B

【分析】这个题是考查对栈操作的掌握,指针 top 为 0 时栈为空。

2.【答案】B

【分析】这个题是考查对链表的掌握程度。可随机访问任一元素不符合链表的特点。

3.【答案】B

【分析】这个题也是考查对链表基本操作的掌握程度。符合操作的就只有 B。

4.【答案】C

【分析】这个题实际上考查 3 个元素(o)有多少种不同的出栈顺序。给三个 o 分别加上下标以区分,则 5 种不同的出栈顺序为: $o_1 o_2 o_3$, $o_1 o_3 o_2$, $o_2 o_1 o_3$, $o_2 o_3 o_1$, $o_3 o_2 o_1$ 。

5.【答案】C

【分析】首先应该输出入度为 0 的结点 v3,所以 B 和 D 肯定不对;在删除 v3 后,下一个入度为 0 的结点是 v1,不是 v4,所以 C 正确。

6.【答案】A

【分析】根据无向图的邻接矩阵的特点,因为 $\langle v1, v2 \rangle$ 和 $\langle v2, v1 \rangle$ 是同一条边,对应的矩阵元素在对称的位置上,必定取相同值,所以该矩阵是对称的。

7.【答案】C

【分析】最少的情况是每一个非叶子结点的其中一个子树只有一个叶子结点,最多的情况是满二叉树,即每个结点都有两个孩子(除了叶子结点)。

8.【答案】D

【分析】 k 个互为同义词的关键字,存入散列表,线性探测一共需要的探测次数为 $1 + 2 + 3 + \dots + k = k(k+1)/2$ 。

9.【答案】A

【分析】归并排序需要额外 $O(n)$ 的存储空间来辅助排序,快速排序的辅助空间为 $O(\log_2 N)$,堆排序仅需要一个辅助空间,空间复杂度为 $O(1)$ 。

10.【答案】C

【分析】考查快速排序的基本方法:先从数据序列中选一个元素,并将序列中所有比该元素小的元素都放到它的右边或左边,再对左右两边分别用同样的方法处理直到每一个待处理的序列的长度为 1,处理结束。把基准元 46 放在正确位置的只有答案 C。

二、综合应用题:41~47 小题,共 70 分

41.【参考答案】

```
int CountLeaf( Tree T )
{
    // 这里假设树不为空
    int LeafNum;
    if ( !T->LeftChild && !T->RightChild )  return 1;
    else {
        if ( T->LeftChild )
            LeafNum = CountLeaf( T->LeftChild );
        if ( T->RightChild )
            LeafNum += CountLeaf( T->RightChild );
    }
}
```

```

    LeafNum += CountLeaf( T->RightChild );
    return LeafNum;
}

```

【分析】

这是一个简单的递归应用,即一棵树的叶子结点个数等于其左右子树的叶子结点个数之和。

需要注意的是,如果将第一个 if 错写为判断 T 是否为空指针,如果是空指针则返回 0,那么在访问叶子结点时,LeafNum 将得到 $0+0$,而不是正确的 1。

42.【参考答案】

```

void InvertString( char S[], int N )
{
    char *p1, *p2, Tmp;
    p1 = S;  p2 = S+N-1;
    for ( ; p1 < p2; *p1++, *p2-- ) {
        Tmp = *p1;  *p1 = *p2;  *p2 = Tmp;
    }
}

void InvertSentence( char S[], int N )
{
    int i, Length = Start = 0;
    InvertString( S, N );
    for ( i=0; i <=N; i++ ) {
        if ( S[i] == ' ' || S[i] == '\0' ) {
            if ( Length ) {
                InvertString( S+Start, Length );
                Start = i+1;  Length = 0;
            }
            else  Start++;
        }
        else  Length++;
    }
}

```

【分析】

此题的核心在于颠倒一个字符串。InvertString 函数将一个长度为 N 的字符串颠倒,方法是令 p1 和 p2 两个指针分别从字符串头尾向中心移动,不断交换它们各自所指的字符。

函数 InvertSentence 用于颠倒整个句子,方法是先将整句作为一个字符串,用 InvertString 完全颠倒,再将其中的每个单词识别出来,分别颠倒回来。识别单词时,用 Start 记录该单词的起始位置,Length 记录该单词的字符串长度。参考答案中的代码可以处理单词间不止一个空格的情况。

模拟试题(六)参考答案及解析

一、单项选择题:1~40 小题,每小题 2 分,共 80 分

1.【答案】D

【分析】在最后一个结点之后插入一个结点,需要知道最后一个结点的位置;删除最后一个结点,需要知道倒数第 2 个结点的位置。比较 4 种候选结构,只有带头结点的双循环链表比较合适。

2.【答案】C

【分析】从题目来看,第一个退栈的是 n ,那么 n 退栈时其他所有数字都在栈内,做的操作必然是所有的都先进栈,然后再一个个出栈,所以 p_i 应该为 $n - i + 1$ 。

3.【答案】D

【分析】这个题是考查对树结构的掌握程度,最贴切的答案应该是 D。

4.【答案】B

【分析】127 阶 B-树中除根外,所有非树叶结点的儿子数在 $\lceil 127/2 \rceil$ 和 127 之间,所以至少是 64。

5.【答案】D

【分析】树其实是一种特殊的图,树的层次遍历就是一种图的广度优先搜索。

6.【答案】D

【分析】把不可能的深度优先遍历排除掉,A 答案中 e 后面不可能是 c ,B 答案中 f 后面不可能 e ,C 答案中 b 后面应该是 d ,所以正确的应该是 D。

7.【答案】D

【分析】在 AOE 网中,完成整个工程所必须花费的时间应该为源点到终点的最大路径长度,具有最大路径长度的路径称为关键路径。所以,如果有回路的话,就不存在关键路径。

8.【答案】D

【分析】A 答案中最小生成树和最短路径没有必然的联系,最小生成树中的点之间的距离不一定是它们的最短路径。B 答案中每条边都加了 1,原有的最短路径不一定还是最短路径,经历的边数多的话,加的值也就多了。C 答案中深度优先遍历可以检查是否存在回路,但不一定能完成拓扑排序,在拓扑排序中要考虑到出度和入度的因素。

9.【答案】A

【分析】这个题考查对各类排序方法的理解。答案是 A。

10.【答案】B

【分析】虽然题目中没有说明是最大堆还是最小堆,但根据 4 个选择判断不可能是最小堆。根据最大堆的调整方法,得到的是 B。

二、综合应用题:41~47 小题,共 70 分

41.【参考答案】

(1)

地址	0	1	2	3	4	5	6	7	8	9	10
元素	33	1	13	12	34	38	27	22			

(2) 成功查找: $(1+1+3+4+1+1+2+8)/8 = 2.625$

不成功查找: $(9+8+7+6+5+4+3+2+1+1+1)/11 = 4.273$

【分析】

在插入过程中,1 和 13 都是一次到位;12 在地址 1 遇到冲突,根据线性探测原则,依次探测地址 2 和 3,发现 3 是空的,所以入位;34 也在地址 1 遇到冲突,依次探测到地址 4 入位;38 和 33 都是一次到位;27 在地址 5 遇到冲突,探测到地址 6 入位;22 在地址 0 遇到冲突,探测到地址 7 入位。

对这 8 个数字进行等概率成功查找,平均查找次数就是各自查找次数之和再对 8 求平均。

所谓等概率不成功查找,是指 n 次不成功的查找中,被散列函数映射到 11 个地址的概率相同,都是 $1/11$ 。则对于被映射到地址 0 的元素,需要查找 9 个地址,直到地址 8 才能确定查找失败。同理对于被映射到地址 1~7 的元素,也需要一直查找到地址 8 才能确定失败。而对于被映射到地址 8~10 的元素,只要查找 1 次就发现该单元为空,即可知查找失败。所以失败的平均查找次数是上述查找次数的和对 11 取平均的结果。

另外,若题目要求采用平方探测,即第 i 次交错探测地址 $[H(x) \pm i^2] \% 11$,则答案变为

地址	0	1	2	3	4	5	6	7	8	9	10
元素	12	1	13		27	34	38	22			33

这里除了 1、13 外,其他元都不能一次到位。12 分别探测了地址 1、1+1、1-1;34 探测了地址 1、1+1、1-1、1+4;38 探测了地址 5、5+1;33 探测了地址 0、0+1、(0-1+11)%11=10;27 探测了地址 5、5+1、5-1;22 探测了地址 0、0+1、(0-1+11)%11=10、0+4、(0-4+11)%11=7,则成功的平均查找次数为:(1+1+3+4+2+3+3+4)/8 = 2.625。

同样,也可以根据散列函数值的 11 种情况,计算每种情况下不成功的查找次数。比如,当散列值为 0 时,其不成功查找次数(即找到空位置时的比较次数)为 6 次,探测顺序是 0、1、10、4、7、9。平方探测并非线性地遍历每个单元,甚至不一定遍历所有单元,还有可能存在散列表中有空位,但找不到的情况。所以,要计算失败的查找次数是个相对比较困难的问题,在此不做深入讨论。

42.【参考答案】

```
bool IsCompleteBinaryTree( Tree T )
{
    int Flag = 0; // 0 表示尚未发现结点有空子树
    if( !T ) return TRUE;
    EnQueue( Q, T );
    while( !IsEmpty( Q ) )
    {
        T = DeQueue( Q );
        if( T->LeftChild && !Flag )
            EnQueue( Q, T->LeftChild );
        else if( T->LeftChild ) return FALSE;
            else Flag = 1; // 结点有空子树
        if( T->RightChild && !Flag )
            EnQueue( Q, T->RightChild );
        else if( T->RightChild ) return FALSE;
            else Flag = 1; // 结点有空子树
    }
    return TRUE;
}
```

【分析】

对于完全二叉树,其关键特点是:若某结点无左孩子就不应有右孩子。算法基本思路是利用队列做树的层次遍历,当发现第一个缺少孩子的结点时,用 Flag 标记。而一旦发现了一个

缺少孩子的结点，其后的所有结点都不能再有孩子。若在 Flag 为 1 的情况下还发现某结点有非空孩子结点，则一定是非完全二叉树。

模拟试题(七)参考答案及解析

一、单项选择题:1~40 小题,每小题 2 分,共 80 分

1.【答案】A

【分析】若设二叉树的高度为 h ,除第 h 层外,其他各层($1 \sim h-1$)的结点数都达到最大个数。第 h 层从右向左连续缺若干结点,这就是完全二叉树。根据一般二叉树满足的公式 $n_0 = n_2 + 1$,以及 $n_0 + n_1 + n_2 = 2435$,可推出 $2n_0 + n_1 = 2436$ 。由此可见 n_1 是偶数。而完全二叉树最多只有 1 个结点度为 1,所以 n_1 只能为 0,于是 $n_0 = 1218$ 。

2.【答案】A

【分析】要求能方便地存取任一指定序号的元素,则只有顺序表比较合适,链表是没有这种特性的;同时顺序表也很方便做在最后进行插入和删除的操作。

3.【答案】A

【分析】先序、中序、后序遍历的区别仅是每个子树的根结点相对其左右子树的访问顺序,而其子树始终是遵循先左后右的原则被访问。任何两叶结点必然分别属于它们最低公共祖先的左右子树,所以在任何一种遍历中,它们的相对顺序都不变。

4.【答案】A

【分析】在给定值为 x 的结点后插入一个新结点时,首先要找到该值,然后再在该结点后插入新结点,因为 x 的结点可能在链表的任何位置,所以应该是 $O(N)$ 。

5.【答案】A

【分析】该题主要考查森林到二叉树的转换,首先将森林中的每棵树变为二叉树,因为转换所得的二叉树的根结点的右子树均为空,故可将各二叉树的根结点视为兄弟从左至右连在一起,就形成了森林到二叉树的转换。所以,转换后的二叉树的右子树的结点个数是森林中除了第一棵树以外的所有树的结点数,那么森林中的第一棵树的结点个数是 $m - n$ 。

6.【答案】A

【分析】根据邻接表的定义,B、C、D 三个选项都是适合用邻接表表示来实现的。因为邻接表只根据边构建链接,所以不方便求顶点的入度。邻接矩阵与邻接表相比更加容易进行顶点入度的计算,对于有向图来说,只需统计矩阵的该列非零元素的个数就可以。

7.【答案】C

【分析】强连通图是指一个有向图中任意两点 v_1, v_2 间存在 v_1 到 v_2 的路径及 v_2 到 v_1 的路径的图。一个有向图 D 是强连通的当且仅当在 D 中存在回路,它经过 D 中每个顶点至少一次。所以,至少需要 n 条边。

8.【答案】A

【分析】该题考查对广度优先搜索算法的掌握,B 答案中 e 后面不应该是 d ,而应该是 b 或 c ;C 答案中 c 后面不应该是 f ,而应该是 b 或 e ;D 答案中 c 后面应该是 d ,所以正确答案只能是 A。

9.【答案】C

【分析】从直接插入排序的算法可见,操作的次数取决于待排记录序列中逆序对的多少。显然逆序对是 C 最少,所以选 C。

10.【答案】C

【分析】常见的图遍历方式有两种:深度优先遍历和广度优先遍历,这两种遍历方式对有向图和无向图均适用。

二、综合应用题:41~47 小题,共 70 分

41.【参考答案】

```
int BinaryTreeHeight( Tree T, int * MaxHeightSumP )  
{
```

```

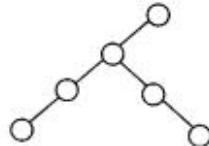
if( !T )  return 0;
int LeftHeight =
    BinaryTreeHeight( T ->LeftChild, MaxHeightSumP );
int RightHeight =
    BinaryTreeHeight( T ->RightChild, MaxHeightSumP );
(*MaxHeightSumP) = max( LeftHeight + RightHeight, *MaxHeightSumP );
return max( LeftHeight, RightHeight ) + 1;
}

int FindRadOfTree( Tree Root )
{
    int MaxHeightSum = 0;
    BinaryTreeHeight( Root, &MaxHeightSum );
    return MaxHeightSum;
}

```

【分析】

此题关键在于对“直径”的正确理解。很容易犯的错误是，将直径简单理解为根结点左右子树的高度之和。右图给出了一个反例：该树的正确直径是4，即两个叶子结点之间的距离，但根结点左子树高度为3，右子树高度为0，则得到错误的答案3。



实际上，树的直径的确是某一结点的左右子树的高度之和，只是这个结点不一定是根结点。于是我们需要遍历所有结点，找出所有左右子树高度和最大的那个结果。这个过程可以与求子树的高度同时进行。

求树高是一个简单的后序遍历，因为树的总高度是其左右子树的最大高度加1。在BinaryTreeHeight这个求二叉树高度的函数中，我们加入了MaxHeightSum（其相应地址指针为MaxHeightSumP）来记录当前所有遍历过的子树中的最大左右高度之和，于是在主函数FindRadOfTree中调用BinaryTreeHeight之后，不仅得到树的高度，同时也得到了树的直径。

42.【参考答案】

(1) 1, 13, 12, 21, 33, 34, 38, 27, 22, 32

(2) 事实上这是拓扑排序的应用，算法思路为：

第1步：构造有向图。将每个元素作为有向图中的一个结点，比较每个元素 x 的散列值 $H(x)$ 与该元素所在单元的地址 $I(x)$ ，若两者不相等，则遵循冲突解决方法从 $H(x)$ 开始查找 x ，并且为查找过程中探测的每一个结点都添加一条指向 x 的边。

第2步：对该图应用拓扑排序，得到的序列就是一个可能的元素插入序列。若还要满足“当有多种插入选择时，总是选择最小的候选数字”，则在实现时将所有入度为0的结点保存在一个最小堆里即可。

算法实现的伪码如下：

```

struct TableRecord {
    int TableSize;
    int T[MAXN];
    int Status[MAXN]; // 1 代表该单元非空
};

typedef struct TableRecord *Table; // 定义散列表

struct GraphRecord {
    int NumVert;

```

```

int M[MAXN][MAXN];
};

typedef struct GraphRecord *Graph; //定义图

struct HeapRecord {
    int Index;
    int Num;
};

typedef struct HeapRecord *MinHeap; //定义最小堆

Graph ConstructG( Table Hash )
//根据输入的散列表构造图
{
    int i, j;
    Graph G;

    G = CreateGraph( Hash ->TableSize ); MakeEmpty( G ); //图初始化
    for ( i=0; i < Hash ->TableSize; i++ ) {
        if ( Hash ->Status[i] == 1 ) { //只处理非空的单元
            j = Hash ->T[i] % Hash ->TableSize; //计算散列值
            while ( i != j ) { //若当前位置 i 与探测位置 j 不同
                G ->M[j][i] = 1; //在图中添加一条指向 i 的边
                j = (j+1) % Hash ->TableSize; //探测下一个位置
            }
        }
    }
    return G;
}

void Topsort( Graph G, Table Hash )
//拓扑排序输出插入序列
{
    MinHeap Q;
    HeapRecord Tmp = malloc( sizeof( struct HeapRecord ) ); //临时堆结点
    int i, j, InDegree[MAXN];

    Q = CreateMinHeap( G ->NumVert ); MakeEmpty( Q ); //初始化最小堆
    for ( i=0; i < G ->NumVert; i++ ) {
        InDegree[i] = 0;
        if ( Hash ->Status[i] == 1 ) { //对每个非空单元的元素,计算入度
            for ( j=0; j < G ->NumVert; j++ )
                InDegree[i] += G ->M[j][i];
            if ( !InDegree[i] ) //入度为 0 者存入最小堆
                Tmp ->Index = i; Tmp ->Num = Hash ->T[i];
                InsertHeap( Tmp, Q );
        }
    }

    while( !IsEmpty( Q ) ) {
        Tmp = DeleteMin( Q ); //输出当前最小的入度为 0 的元
        Output( Tmp ->Num );
        i = Tmp ->Index;
        for ( j=0; j < G ->NumVert; j++ ) {

```

```
    if ( G->M[i][j] ) InDegree[j] --; //将该结点从图中删去
    if ( !InDegree[j] ) | /入度为0者存入最小堆
        Tmp->Index = j; Tmp->Num = Hash->T[j];
        InsertHeap( Tmp, Q );
    }
}
|
|
|
```

【分析】

该题目将散列查找与拓扑排序巧妙结合，考查了对两个知识点的掌握程度。之所以用拓扑排序，是因为若在插入 x 的过程中遇到冲突，则冲突元必然在 x 之前被插入，所以是 x 的前驱结点，即在插入 x 之前，必须保证其所有前驱结点已经被插入。

实现中需要注意的细节问题是：空单元若不做特殊识别，也可能被当作没有前驱的结点输出，就会导致错误。所以我们需要对散列表中的每个单元做标记 ($\text{Hash} \rightarrow \text{Status}[i]$)，仅处理标记为非空 (取值为 1) 的单元。

模拟试题(八)参考答案及解析

一、单项选择题:1~40 小题,每小题 2 分,共 80 分

1.【答案】A

【分析】对顺序表来说,插入、删除和排序都要移动元素位置,不可能是常数时间复杂度,只有 A 可以在常数时间内完成。

2.【答案】A

【分析】该树的结点总数为 $n = n_0 + n_1 + n_2 + \dots + n_m$, 根据孩子数计算该树边的总数为 $1 \times n_1 + 2 \times n_2 + \dots + m \times n_m$ 。因为树中边的条数是 $n - 1$, 所以联合上述两式, 可以得到叶结点数 $n_0 = (1 \times n_1 + 2 \times n_2 + \dots + m \times n_m) - (n_1 + n_2 + \dots + n_m - 1)$ 。

3.【答案】A

【分析】这个题考查对树的遍历的掌握。根据本书数据结构部分中讲“二叉树的遍历”时提供的算法,可以通过前序遍历和中序遍历的结果把树结构确定下来,那么后序遍历也就可以得到。

4.【答案】C

【分析】这个题同样是考查对树遍历的掌握。中序序列顺序是 LNR, 后序序列顺序是 LRN, 我们可以看出 N 和 R 部分是相反的。所以若要两序列完全相反, 树中必须没有 L, 即每一个结点都没有左孩子。

5.【答案】C

【分析】因为在度的计算中, 每一条边被它连接的两个顶点计算了两次, 所以, 顶点的度数之和等于所有边数的 2 倍。

6.【答案】D

【分析】二次探测再散列也就是平方探测法, $h_i(X) = (\text{Hash}(X) + F(i)) \% 14$, $F(0) = 0$, $F(i) = i^2$ 。
49 最初被映射到 5 ($i=0$ 时), 之后是 5+1 ($i=1$), 最后是 5+2² ($i=2$)。

7.【答案】C

【分析】稳定的排序算法有直接插入排序、二分法插入排序、归并排序、冒泡排序和基数排序。
不稳定的排序算法有快速排序、直接选择排序、希尔排序和堆排序。

8.【答案】D

【分析】该题是考查对希尔排序的掌握, 初始步长为 4, 在序列中每隔 4 个长度进行排序, 即 49 和 76, 38 和 13, 65 和 27, 97 和 50 分别进行排序。

9.【答案】B

【分析】单链表做删除元素的操作应该是 B。

10.【答案】B

【分析】观察 4 个选项, 可知建立的是最小堆。按照建立最小堆的方法进行操作, 应该是 B。

二、综合应用题:41~47 小题,共 70 分

41.【参考答案】

设交通图用邻接矩阵 A 存储, 对角元为 ∞ , $A[i][j] = W_{ij}$ 。算法伪码如下:

```
int SetHospital( TwoDimArray A, int N )
{
    int i, j, k;
    TwoDimArray D;
    int Hospital, MaxDist, MinDist = MAXNUM;

    for ( i = 0; i < N; i++ )
        for ( j = 0; j < N; j++ )
            D[i][j] = A[i][j];
```

```

for ( k = 0; k < N; k ++ )
    for ( i = 0; i < N; i ++ )
        for ( j = 0; j < N; j ++ )
            if ( D[i][k] + D[k][j] < D[i][j] )
                D[i][j] = D[i][k] + D[k][j];
for ( i = 0; i < N; i ++ )
    MaxDist = 0;
    for ( j = 0; j < N; j ++ )
        MaxDist = max( MaxDist, D[i][j] );
    if ( MinDist > MaxDist ) {
        MinDist = MaxDist; Hospital = i;
    }
}
return Hospital;
}

```

【分析】

算法的基本思路是：首先用弗洛伊德(Floyd)算法求出任意两村庄间的最短距离；对每个村庄*i*,用MaxDist记录它到其他村庄的最短距离中的最长距离；MinDist记录的是所有最长距离中最短的那个距离，其对应的村庄*i*就是医院所在地。

42.【参考答案】

```

List MergeSortedLists( List L1, List L2 )
{
    List L = L1;
    List Tail = L2;
    L1 = L1 -> Next; L2 = L2 -> Next;
    free( Tail ); Tail = L;
    //释放第二个链表的表头，并将新链表的表尾指针Tail初始化
    while ( L1 && L2 ) {
        if ( L1 -> Data > L2 -> Data ) {
            Tail -> Next = L2; L2 = L2 -> Next;
            //第二个链表的第一个结点值小，将其插入新链表尾
        }
        else {
            Tail -> Next = L1; L1 = L1 -> Next;
            //第一个链表的第一个结点值小，将其插入新链表尾
        }
        Tail = Tail -> Next;
    }
    if( L1 ) Tail -> Next = L1;           //将剩余的第一个链表接到新链表表尾
    else if( L2 ) Tail -> Next = L2;       //将剩余的第二个链表接到新链表表尾
    return L;
}

```

【分析】

算法思路为顺序比较L1和L2当前所指的两结点的Data域，将小的那个结点从原链表删除，贴到合并链表的尾部。如此进行到至少其中一个链表为空。若此时另一个链表不为空，则将另一个链表整个贴到合并链表的尾部。

注意原始 2 个链表有 2 个头结点，合并后只保留一个，需要释放另一个的空间。

稍复杂一点的类似题目有：将两个按 Data 域递增有序的链表合并成一个按 Data 域递减有序的链表。这时解题的基本思路不变，只要将原来的表尾插入改为表头插入即可。当然，当其中一个链表为空而另一个链表不为空时，不能直接将剩余链表贴到表尾，而必须将剩余结点一个一个插入表头。

另外还有一类题目是用数组实现的合并。例如，要求将 L2 并入 L1，并且要求移动元素的次数尽可能少，则必须事先算好合并后 L1 的长度，然后从 L1 的最终尾部向前插入元素。插入过程与前面算法类似，但是是从尾部开始比较 L1 和 L2 相应元素大小，将较大的元素先插入到后面的位置。算法伪码如下：

```
void MergeSortedArrays( ElementType L1[], int N1, ElementType L2[], int N2 )
{
    int p1, p2, cur;
    p1 = N1 - 1;           //p1 指向 L1 当前处理的元素位置
    p2 = N2 - 1;           //p2 指向 L2 当前处理的元素位置
    cur = N1 + N2 - 1;     //cur 指向下一个要插入的元素在 L1 中的最终位置
    while ( (p1 >= 0) && (p2 >= 0) ) {
        if ( L1[p1] > L2[p2] ) {
            L1[cur] = L1[p1]; p1--;
        }
        else {
            L1[cur] = L2[p2]; p2--;
        }
        cur--;
    }
    while( p2 >= 0 ) //将 p2 的剩余元素插入到 L1 中
        L1[p2] = L2[p2]; p2--;
}
```

模拟试题(九)参考答案及解析

一、单项选择题:1~40 小题,每小题 2 分,共 80 分

1.【答案】C

【分析】稀疏矩阵的三元组表示法需要为每个元素配备 3 个单位的存储空间(行,列,元素值),另外还需要 3 个单位存储矩阵的行数、列数、非零元数,所以需要的空间为 $3(t+1)$ 个单元。若三元组表示法比二维数组存储节省空间,则要求 $3(t+1) < m \times n$,所以应该选答案 C。

2.【答案】C

【分析】要求两个线性表头尾连接,又要求辅助空间尽量小,那所选结构必须是能马上找到头和尾,所以就选单循环链表,而且必须是带尾指针的单循环链表。

3.【答案】D

【分析】正确的只有①和④。②不正确,二叉树也存在只有 1 个结点或没有度为 2 的结点的情况;③也不正确,左右子树交换后就不是同一棵树了。④考查的是完全二叉树和满二叉树的定义。

4.【答案】D

【分析】前序遍历结果是 NLR,中序遍历结果是 LNR,要两种结果相同,必然不能出现 L(N 是不能缺少的),所以是 D。该题考查对树的遍历的掌握。

5.【答案】B

【分析】四项选择中,只有堆排序可以在排序过程中得到最终的部分连续的有序子列,其他三种算法都可能需要在完全排序后才能得到前 100 个最大值。

6.【答案】D

【分析】顺序查找、二分法和利用二叉搜索树查找都与结点个数 n 有关,只有哈希表的查找与 n 无关。

7.【答案】C

【分析】线性探查法按照公式: $h_i(X) = (\text{Hash}(X) + i) \% \text{TableSize}$ 。即,每次探测与前一次相邻的位置,若越过散列表的边界,则循环从地址 0 开始,所以可以大于或小于原散列地址。当探测回到原散列地址时,表示散列表已满,不能插入。

8.【答案】D

【分析】按照分治算法的时间复杂度的计算,必须符合 A、B、C,才能达到题目所要求的时间复杂度。

9.【答案】C

【分析】答案 A 肯定不对, n 在 m 的右方,那么中序遍历时 n 在 m 的后面;答案 B 也不对, n 是 m 的祖先,那么 m 也可以是 n 的左孩子,则遍历时 m 在 n 的前面;答案 D 的错误与 B 同理。只有 C 是对的。

10.【答案】D

【分析】根据栈的特性,如果输出序列的第一个元素是 i ,只能说明从 1 到 $i-1$ 已经在栈里,但对第 j 个元素来说是不确定的。

二、综合应用题:41~47 小题,共 70 分

41.【参考答案】

弗洛伊德(Floyd)算法基本思路是:

定义 $D^k[i][j]$ 为从结点 i 到结点 j ,且中间结点序号不超过 k ($k=0,1,\dots,N-1$) 的最短路径长度,则有递推关系式:

$$D^k[i][j] = \min(D^{k-1}[i][j], D^{k-1}[i][k] + D^{k-1}[k][j]).$$

从图的邻接矩阵 $\text{Cost}[i][j] = W_{ij}$ 出发(其中 $W_{ij} = \infty$ 表示结点 i 和 j 之间没有边; Cost 矩阵的

对角元为 ∞),令 $D^{-1}[i][j] = \text{Cost}[i][j]$;则 $D^{N-1}[i][j]$ 就是从结点*i*到结点*j*的最短路径长度。

要用该算法判断图是否有回路,只要检查对角元 $D^{N-1}[i][i]$,若存在非 ∞ 的对角元,就表明图中有回路。若 $D^{N-1}[i][i]$ 非 ∞ ,说明存在某个*k*,使得 $D^k[i][i]$ 的值被更新为($D^{k-1}[i][k] + D^{k-1}[k][i]$),即存在一条从*i*到*k*、再从*k*到*i*的路径。

【分析】

类似的题目还有:如何判断图中有负回路(即回路中边的权值之和为负数)?这时应将对角元 $\text{Cost}[i][i]$ 初始化为0,当 $D^k[i][i] < 0$ 时,图中必存在负回路。

42.【参考答案】

设原始的*N*个浮点数存储在数组*A*中,找出*M*个最大数存储在数组*B*中。

```
void PercDown( double A[], int i, int N )
{
    int child;
    double Tmp = A[i];
    for ( ; (i * 2 + 1) < N; i = child ) {
        child = i * 2 + 1;
        if ( (child + 1) < N && A[child + 1] > A[child] )
            child++;
        if ( A[child] > Tmp )
            A[i] = A[child];
        else
            break;
    }
    A[i] = Tmp;
}

void FindMLargestElements( double A[], int N, double B[], int M )
{
    int i, j;
    for ( i = N / 2; i >= 0; i-- )
        PercDown( A, i, N );
    for ( i = N - 1, j = 0; i > 0 && j < M; i--, j++ ) {
        B[j] = A[0]; A[0] = A[i];
        PercDown( A, 0, i );
    }
}
```

【分析】

堆排序是可以不经过完全排序就从排序过程中得到部分有序集合的算法,且最坏情况下的复杂度最好,即 $O(N \log_2 N)$ 。而其他排序算法或者必须完全排序(如插入、冒泡、快速、归并、希尔排序等),或者复杂度比较高(如选择排序)。这里将堆排序的算法略作修改,在建立最大堆后,只循环*M*次,每次不是将堆顶元素换到堆底,而是将之存入*B*,再将堆底元素放到堆顶,进行下一轮调整。

若不要求找到的*M*个最大数有序,还可以进一步优化算法:若*M* $\leq N/2$,则上述算法适用;若*M* $> N/2$,则可以改为建立最小堆,把最小的(*N*-*M*)个数字沉到堆底,则前面*M*个就是最大的。

另外一种类似题目是求第*M*个最大元,而不是前*M*个最大元。解决这种问题有很多不同算法,其中效率较高的是利用类似于快速排序算法的思路:

- 首先选取支点将集合划分为 2 部分(不包括支点本身): S_1 和 S_2 ;
- 若 $M \leq |S_2|$, 则第 M 大元必在 S_2 中, 递归;
- 若 $M = |S_2| + 1$, 则支点就是第 M 大元;
- 若 $M > |S_2| + 1$, 则第 M 大元必在 S_1 中, 并且是第 $(M - |S_2| - 1)$ 大元, 递归。

参考程序如下:

```

ElementType FindMthElement( ElementType A[ ], int Left, int Right, int M )
{
    int i, j;
    ElementType Pivot;
    Pivot = Median3( A, Left, Right );
    //Median3 找左、中、右三元的中间值作为支点 Pivot
    //并将最小值存于 A[ left ], 支点存于最右边 A[ Right - 1 ], 最大值存于 A[ Right ]

    i = Left;      j = Right - 1;
    for ( ; ; ) {
        while( A[ ++i ] < Pivot ) ++
        while( A[ --j ] > Pivot ) --
        if ( i < j )
            Swap ( &A[i], &A[j] );
        else break;
    }
    Swap( &A[i], &A[Right - 1] );      //将支点换到正确的位置
    //此时支点左边  $S_1$  集合有  $(i - Left)$  个元素, 右边  $S_2$  有  $(Right - j - 1)$  个元素
    if ( M <= Right - j - 1 )
        return FindMthElement( A, i + 1, Right, M );
    if ( M == Right - j )
        return Pivot;
    else
        return FindMthElement( A, Left, j, M - Right + j );
}

```

模拟试题(十)参考答案及解析

一、单项选择题:1~40 小题,每小题 2 分,共 80 分

1.【答案】D

【分析】该栈是用链表的数据结构,ST 指向表头,所以,符合要求的是 D。C 答案中操作次序颠倒了。

2.【答案】B

【分析】记非空二叉树的度为 1 的结点个数为 n_1 ,则根据二叉数的边数($n_1 + 2n_2$)比总结点数($n_0 + n_1 + n_2$)少 1,于是推出 $n_0 = n_2 + 1$ 。

3.【答案】C

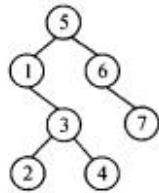
【分析】线索化二叉树的特点是树中没有空链域,t 结点若没有左子树,则该结点的左指针可以用来做线索。所以 $t \rightarrow \text{ltag} = 1$ 并且 $t \rightarrow \text{left}! = \text{NULL}$ 。

4.【答案】B

【分析】能够进行二分查找,线性表必须是顺序方式存放,并且元素是已排序的。

5.【答案】D

【分析】此题先要构造出树,解法是自顶向下分析每个子树的根结点的值。从给出的树结构看,根结点左子树有 4 个结点,右子树有 2 个结点,那么根值必定在这 7 个数中排第 5 位,即是 5;考察根的左子树,该子树的根没有左孩子,说明它在 4 个数中最小,于是一定是 1;根的右子树也是同理,其根值一定是 6,7 中的最小值,即是 6;在 2,3,4 组成的子树中,根值居中,是为 3;以下同理类推,在构造出树以后,就得到答案。



6.【答案】C

【分析】所有 4 种候选排序算法中只有归并排序是需要非常数级的额外的存储空间。

7.【答案】D

【分析】若前 2 层是满树,则第三层最多 4 个结点,每个结点可能是空,也可能不是空,但是不能完全是空,所以有 $2^4 - 1$ 种;若前 2 层不是满树,即第 2 层只有 1 个结点,理由同上,得到 $2^2 - 1$ 种;又因为第 2 层少哪一个不一定,所以乘以 2。由此推出不同的二叉数一共有 $2^4 - 1 + 2(2^2 - 1) = 21$ 。

8.【答案】B

【分析】对于有向图,行的非零元素个数是顶点的出度,列的非零元素个数是顶点的入度。

9.【答案】B

【分析】无向图中所有顶点的度数之和等于边的 2 倍,所以,其余顶点的度数都为 2 的话,顶点个数最少。 $2 \times 15 = 3 \times 6 + 2 \times 3 + x \times 2$,得到 $x = 3$ 。所以,顶点个数至少有 $3 + 2 + 3 = 8$ 。

10.【答案】A

【分析】这个题是考查对树的遍历的掌握,后序遍历是 EDBFGCA,则 A 肯定是树的根结点,从中序遍历中又可知左子树有 EDB,右子树有结点 FCG。以此类推,可得到这棵二叉树的结构。

二、综合应用题:41~47 小题,共 70 分

41.【参考答案】

```
Tree FindLowestCommonAncestor( Tree T, int V1, int V2 )
{
    while( T )
        if( (T->Data > V1) && (T->Data > V2) )
            T = T->LeftChild;
```

```

        else if( ( T -> Data < V1 ) && ( T -> Data < V2 ) )
            T = T -> RightChild;
        else
            return T;
    }
    return NULL;
}

```

【分析】

根据二叉排序树的特性,若 V_1 和 V_2 都在根结点的同一边子树里,则从根结点开始逐层向下找。这样发现的第一个将 V_1 和 V_2 分别包含在左右不同子树里的那个结点,就是 V_1 和 V_2 的最近公共祖先,因为再向下一层,无论是向左还是向右,都不可能找到同时包含这两个结点的子树了。

42.【参考答案】

```

void SortString( char S[] )
{
    int i = -1;
    int j = strlen( S );
    for( ; ; )
        while( islower( S[ ++i ] ) && i < j ) ++
        while( ! islower( S[ --j ] ) && i < j ) ++
        if( i < j )
            Swap( &S[ i ], &S[ j ] );
        else
            break;
}

```

【分析】

注意题目并不要求字母是有序的,所以没必要采用任何排序算法,但是快速排序中为了确定支点位置而将排序对象分为大小两个子集的算法可以略加修改用于解决这个问题。只是这里在移动 i 和 j 时,不是与支点比较,而是判断其对应的字母是大写还是小写。当 i 指向大写字母, j 指向小写字母时,交换两字母的位置即可。算法的时间复杂度是线性的。

另外还有一种更简单的实现:

```

void SortString( char S[] )
{
    int i, last;           // i 为待处理子串的第一个位置, last 是大写块的最后一个位置
    i = strlen( S );
    while( ! islower( S[ --i ] ) && i > 0 ) ++
    last = i + 1;          // last 初始化
    for( ; i >= 0 ; i -- )
        if( ! islower( S[ i ] ) )
            Swap( &S[ --last ], &S[ i ] );
}

```

该算法的思路非常简单而且直观,即从右向左扫描字符的同时,保持已经扫描过的子串中是两块连续的大小写区块,如下图所示。但是这个算法在处理如“ $A\cdots Aa$ ”这样的字符串时,就需要做 $n-1$ 次交换(设字符串长度为 n),而参考答案中的算法只需要 1 次交换。算法效率较低的关键原因,是该算法只允许一个指针 i 单边扫描,另一个指针 $last$ 实际上受制于 i ,只能每次通过交换一对字符来移动一格。而参考答案中的算法允许两个指针双向独立扫描,所以效率较高。



更复杂一点的类似题目要求将元素分 3 类,例如在上述字符串中加入空格,要求所有小写字母移动到字符串的头部、空格移动到中间、大写字母移到尾部,并且限制只能通过交换字符来调整(若无此限制,则可以用桶排序非常简单地解决这个问题)。这时用与上述方法类似的思路,可用 3 个指针 i, j, k ,分别指向小写字母、空格、大写字母区块的各自最后一个字符,每轮循环的目的是将未处理部分的第一个字符正确地交换到三块之一中,使当前循环执行后,仍然保持三块的连续性,如下图所示,即每一步检查 $S[k+1]$:



- 若它是大写,则 $k++$;
- 若它是空格,则交换 $S[k+1]$ 与 $S[j+1]$,然后 $k++, j++$;
- 若它是小写,则先交换 $S[k+1]$ 与 $S[j+1]$, $k++, j++$,然后交换 $S[i+1]$ 与 $S[j]$, $i++$ 。

该算法的优点是思路简单,比较容易实现。当然算法的效率与其可读性经常是一对矛盾,这个简单的算法仍然存在着单指针扫描所带来的效率问题,例如处理形如“a 空 Aaa……a”这样的字符串时,就需要 $O(N)$ 次交换。

一个效率较高的算法必须令至少 2 个指针相对独立扫描。用 3 个指针 i, j, k ,分别指向小写字母、空格、大写字母区块的下一个待处理的字符,在每一趟依次处理这 3 个指针所指的字符。这里 i 和 j 均从左向右扫描, k 从右向左扫描。如下图所示。



当 i 扫描下一个字符发现空格时,若 j 还在初始位置 -1 或者与 i 重叠,则令 j 指到 i 的下一个位置;否则交换 i 和 j 所指的字符, $j++$ 。当 i 发现大写字母时,就交换 i 和 k 所指的字符, $k--$ 。

然后处理 j ,在发现了第 1 个空格之后, j 就指向当前发现的最后一个空格的下一位置,若这个位置是小写字母,交换 i 和 j 所指的字符, $i++$ 。当 j 发现大写字母时,就交换 j 和 k 所指的字符, $k--$ 。

最后从右向左检查 k ,若发现的非大写字母是空格,交换 k 和 j 所指的字符, $j++$ 。当 k 发现小写字母时,就交换 i 和 k 所指的字符, $i++$ 。

算法的伪码如下:

```
void SortString( char S[] )
{
    int i = 0;
    int j = 0;
    int k = strlen( S ) - 1;
    for( ; ; ) {
        while( i < k && S[i] != ' ' && islower( S[i] ) ) i++;
        if ( S[i] == ' ' ) {
            if ( i < j ) Swap( &S[i], &S[j] ); j++;
            else j = i + 1;
        }
    }
}
```

```
    else if( i < k ) | Swap( &S[i], &S[k] ); k--; |
        else break;
    while ( j > i && j <= k && S[j] == ' ' ) | j++; |
    if ( j > i && j <= k ) |
        if ( islower(S[j]) ) | Swap( &S[i], &S[j] ); i++; |
        else | Swap( &S[k], &S[j] ); k--; |
    |
    while ( k > j && S[k] != ' ' && ! islower(S[k]) ) | k--; |
    if ( islower(S[k]) ) | Swap( &S[i], &S[k] ); i++; |
    else if ( S[k] == ' ' && k < j ) break;
        else if ( j > i ) | Swap( &S[j], &S[k] ); j++; |
}
}
```