对外经济贸易大学

2005 年攻读硕士学位研究生入学考试

441 应用软件基础试题

注意：请将所有答案写在答题纸上

第一部分：C 语言程序设计（共 75 分）

一、选择题(下列各题 A,B,C,D 四个选项中,只有一个选项是正确的。请选择正确选项)(每小题 1 分,共 20 分)

1. 设 a=2, b=2, c=2; 执行表达式 c +=a* =b++ ; 变量 a、b、c 三个变量的值分别是【1】

A) 6、2、8　　B) 4、3、6　　C) 6、3、8　　D) 6、3、9

2. 若有定义：int a[5],*p=a; 下列哪个语句是不正确的【2】

A) scanf("%d",a+1);　　B) scanf("%d",*p+1);

C) scanf("%d",&a[1]);　　D) scanf("%d", p+1);

3. 下面程序的运行结果是【3】

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

A) 28　　B) 22　　C) 16　　D) 4

4. 执行以下程序的输出是【4】

```
main()
{  int a=10,b=0, c=1, d=0;
   if( c=a<b  &&  d =a>c)   ;
    printf("%d,   %d \n",c,d);}
```

A) 1, 0　　B) 0, 1　　C) 0, 0　　D)1, 1

5. 下面程序的运行结果是【5】

【3】 A) --j>0 B) j-->0

C) --len>0 D) len-->0

五、程序填空(每空2分,共20分)

下面程序的功能是从键盘上输入两个字符串，对两个字符串分别排序；然后将它们合并，合并后的字符串按ASCII码值从小到大排序，并删去相同的字符。

```
#include <stdio.h>
strmerge(char  *a, char *b, char *c)
{ char  t,  *w;
  w=c;
  while ( *a!= '\0'  【1】  *b!= '\0' )
  { t=【2】 ? *a++ : *b < *a ? *b++ : (【3】);
        /* 将*a 、*b 的小者存入 t */
    if ( *w 【4】 '\0' )  *w=t;
    else if(t【5】 *w)  *++w=t;  /* 将与*w 不相同的 t 存入 w */
    }
  while ( *a!= '\0' )          /* 以下将 a 或 b 中剩下的字符存入 w */
    if ( *a!= *w ) *++w= *a++;
    else  a++;
  while ( *b!= '\0' )
    if ( *b!= *w ) *++w= *b++;
    else  b++;
  *++w=【6】;
}
strsort( char  *s)
{ int i, j, n;
  char t, *w;
  【7】;
  for ( n=0;*w!= '\0' ;【8】)
    w++;
```

```
    for ( i=0; i<n-1; i++)
      for ( j=i+1; j<n; j++)
        if ( s[i]>s[j])
          { 【9】 t=s[i]; s[i]=s[j]; s[j]=t;
  }
  main()
  { char s1[100] ,s2[100], s3[200];
    printf (" \n Enter First String:" );
    scanf( "%s" ,s1);
    printf (" \n Enter Second String:" );
    scanf( "%s" ,s2);
    strsort(s1);
    strsort(s2);
    【10】=' \0' ;
    strmerge (s1,s2,s3);
    printf ( "\n Result: %s", s3);
  }
```

六、编写程序(共10分)

编写函数void strcmb(char *s1，char *s2)，函数的功能是将主函数中输入的两行字符串进行连接操作，然后将串中全部空格移到串尾。主函数如下所示。

```
main( )
{ char  str1[200], str2[100];
  void  strcmb( char *s1, char *s2 );
  printf( "Input the first string" );
  scanf( "%s" ,str1);
  printf( "Input the second string" );
  scanf( "%s" ,str2);
  strcmb( str1, str2 );
```

```
    printf( "%s" , str1);
}
```

第二部分：数据库系统（共75分）

一、单项选择题（每题1分，共10分）

1. 数据库的三级模式结构中，最接近外部存储器的是【1】

A）外模式 B）内模式 C）模式 D）概念模式

2. 对数据库的操作要以【2】的内容为依据

A）数据模型 B）数据库管理系统

C）数据字典 D）运行日志

3. 设关系R和S具有相同的目，且相应属性取自同一个域，各有10个元组，则R∪S的元组个数为【3】

A）10 B）20 C）100 D）小于等于20

4. 设属性A是关系R的主属性，则属性A不能取空值NULL，这是【4】

A）实体完整性规则 B）参照完整性规则

C）用户定义完整性规则 D）域完整性规则

5. 查询学生刘江所选修的所有课程的成绩，原始关系代数表达式如下：

$\pi_G(\sigma_{S.SNO=SC.SNO\ AND\ S.SN='刘江'}(S\times SC))$

优化后的关系代数表达式第1步应做【5】

A）$\sigma_{S.SNO=SC.SNO}(S\times SC)$ B）$\sigma_{S.SN='刘江'}(S\times SC)$

C）$\sigma_{S.SN='刘江'}(S)$ D）$S\times SC$

6. 一个关系模式R{X1, X2, X3, X4}，假定该关系存在如下函数依赖：X1→X2, X1→X3, X3→X4，则该关系属于【6】

A）2NF B）3NF C）4NF D）BCNF

7. 以下函数依赖中属于平凡函数依赖的是【7】

A）Sno Cname→Cname Grade B）Sno Cname Grade→Cname Grade

C）Sno Cname→Sname Grade D）Sno Grade→Sname

8. 如果事务T对数据D已加S锁，则其他事务对数据D【8】

A）可以加S锁，不能加X锁 B）可以加S锁，也可以加X锁

C）不能加S锁，可以加X锁 D）不能加任何锁

9. 数据库后备副本的主要用途是【9】

A）数据转储　　B）历史档案　　C）故障恢复　　D）安全性控制

10. 数据库恢复时，对尚未做完的事务执行【10】

A）REDO 处理　　B）UNDO 处理　　C）ABORT 处理　　D）ROLLBACK 处理

二、填空题（每题 1 分，共 10 分）

1. 事务的四个特性是：原子性、隔离性、持久性和_____【1】_____。

2. 传统的集合运算是从关系的_____【2】_____方向进行。

3. SQL 是一种介于关系代数和_____【3】_____之间的结构化查询语言。

4. 关系模式的操作异常问题往往是由_____【4】_____引起的。

5. 数据库设计过程中，对视图的设计属于_____【5】_____设计阶段。

6. _____【6】_____称为封锁粒度。

7. 可信计算机系统评测标准 TCSEC 中，系统可信程度最低级别是_____【7】_____级。

8. 数据库安全性控制主要通过数据库系统的_____【8】_____机制实现。

9. 完整性约束条件作用的对象可以是_____【9】_____、元组、列三种。

10. 数据库系统中，由_____【10】_____负责全面管理和控制数据库系统。

三、判断题（认为表述正确，填写“T”，否则填“F”，每题 1 分，共 5 分）

F 1. 从 E-R 模型向关系模型转换时，一个 1：n 联系可以转换为一个独立的关系模式，也可以与联系的任意一端实体所对应的关系模式合并。

F 2. 使用视图可以加快查询语句的执行速度。

F 3. 两端锁协议是使并发操作实现可串行化调度的必要条件。

T 4. 在视图中插入一个元组，该元组会同时插入到基本表中。

F 5. 静态 SQL 语句是指主变量的个数、数据类型和值在预编译时都已确定的 SQL 语句。

四、名词解释：（每题 3 分，共 12 分）

1、数据库

2、聚簇

3、非主属性

4、静态转储

五、综合题（共4题，共38分）

1. 已知关系R、S如下，求R÷S的结果。（4分）

R	A	B	C	D	E
	x	a	x	a	1
	x	a	z	a	1
	x	a	z	b	1
	y	a	z	a	1
	y	a	y	b	3
	y	a	z	b	1
	z	a	z	a	1
	z	a	y	b	1
	z	a	z	b	1

S	D	E
	a	1
	b	1

R÷S

A	B	C
x	a	z
y	a	z
z	a	z

2. 设有一个记录各个球队队员每场比赛进球数的关系模式：（8分）

R（队员编号，比赛场次，进球数，球队名，队长名）

如果规定每个队员只能属于一个球队，每个球队只有一个队长。

① 试写出关系模式R的基本函数依赖和键码。

② 将R分解成3NF模式集，并说明理由。

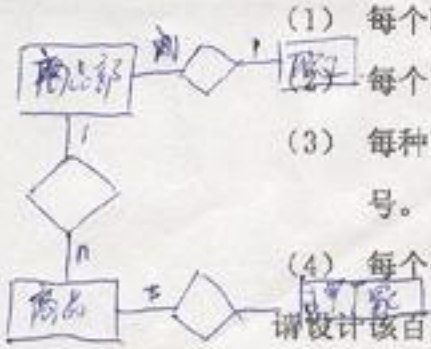
3. 设有一家百货商店，已知如下信息：（8分）

（1）每个职工的数据包括：职工号、姓名、住址和所在的商品部。

（2）每个商品部的数据包括：部门号、部门名、职工、经理、经销的商品。

（3）每种商品的数据包括：商品名、生产厂家、价格、型号、内部商品代号。

（4）每个生产厂家的数据有：厂名、地址、向商店提供的商品价格。



请设计该百货商店的概念模型，再进一步转换为关系模型，并用下划线标出主码。

4. 设有关系模式如下：（每小题3分，共18分）

学生 S（SNO, SNAME, SEX, AGE）

课程 C（CNO, CNAME, PCNO, TEACHER）

选课 SC（SNO, CNO, GR）

对应的中文含义为：学生S（学号，姓名，性别，年龄）

课程C（课程号，课程名，先修课程号，授课教师名）

选课SC（学号，课程号，课程成绩）

写出满足下列各题要求的SQL查询语句。

（1） 找出张老师所授课程的课程号和课程名。

（2） 找出年龄小于20岁的女学生的学号和姓名。

（3） 将选修课程“DB”的学生学号、姓名建立视图SDB。

（4） 找出未选“PROG”课程的学生姓名。

（5） 统计选修各门课的学生人数，输出课程号和人数，查询结果按课程号降序排列。

（6） 从学生选课关系SC中删除“张军”同学的所有元组。

(1)
```
SELECT C.CNO, C.CNAME
FROM C
WHERE C.TEACHER='张老师';
```

(2)
```
SELECT S.SNO, S.SNAME
FROM S
WHERE S.SEX='女' and
      S.AGE<20;
```

(3)
```
CREATE VIEW SDB
AS
    SELECT S.SNO, S.SNAME
    FROM S, SC, C
    WHERE S.SNO=SC.SNO and SC.CNO='DB' C.CNO and C.CNAME='DB';
```

(4)
```
SELECT S.SNAME
FROM S
WHERE S.SNO NOT IN
      SELECT S.SNO
      FROM S, C, SC
      WHERE S.SNO=SC.SNO and
            SC.CNO=C.CNO and
            C.CNAME='PROG';
```

(5)
```
SELECT CNO, COUNT(SNO)
FROM SC
GROUP BY CNO DESC;
```

(6)
```
DELECT
FROM SC
WHERE SC.SNO =
      SELECT S.SNO
      FROM S
      WHERE S.SNAME='张军';
```

```
#include <stdio.h>
main()
{int  a=1,b=0;
  do
    { switch ( a )
      { case  1 : b=1; break;
        case  2 : b=2; break;
       default : b=0;
       }
      a=a+b ;
    } while(b);
    printf( "%d,%d" ,a,b)
}
```

A) 4,0　　B) 4,2　　C) 1,0　　D) 2,2

6. 若有以下定义和语句：

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

则以下正确的语句是【6】

A) a=5;　　B) a={2,'a',1,2}

C) printf("%d\n" ,a);　　D) n=a;

7. 与表达式“(n) ? (c++) : (c--)”中的表达式（n）等价的表达式是【7】

A) (n==0)　　B) (n==1)　　C) (n!=0)　　D) (n!=1)

8. 调用 strlen(“\\t\\0ef\n”) 的返回值是【8】

A) 13　　B) 8　　C) 7　　D) 3

9. 已知 int a[]={1,2,3,4},y,*p=a;则执行语句 y=(*++p)--;之后，数组 a 各

元素的值变为【9】

A) 0, 1, 3, 4 B) 1, 1, 3, 4 C) 1, 2, 2, 4 D) 1, 2, 3, 3

变量 y 的值是【10】

A) 1 B) 2 C) 3 D) 4

10. 下面程序的输出结果是【11】

```
int w=3;
main()
{int w=10;
printf( "%d\n" , fun(5)*w );
}
 fun( int k )
 {if (k==0) return(w);
  return (fun(k-1)*k);
  }
```

A) 360 B) 3600 C) 1080 D) 1200

11. 下面程序的输出结果是：

```
#include <stdio.h>
int k=1;
main()
{int i=4;
 fun(i);
 printf( "\n%d,%d" ,i,k);      /*输出结果为 【12】*/
}
 fun( int m )
 { m+=k; k+=m;
   {char  k= 'b' ;
    printf( "\n%d" , k- 'A' );   /*输出结果为 【13】*/
    }
  printf( "\n%d,%d" , m,k);     /*输出结果为 【14】*/
```

```
}
```

【12】A) 4, 6　　B) 5, 6　　C) 4, 1　　D) A、B、C 都不对

【13】A) 1　　B) −59　　C) −64　　D) A、B、C 都不对

【14】A) 5, 66　　B) 1, 66　　C) 5, 6　　D) A、B、C 都不对

12. 下面程序的运行结果是【15】

```
void f (int v, int w)
{ int t;
  t=v; v=w; w=t;}
main()
{ int x=1, y=3, z=2;
  if (x>y)  f( x, y ) ;
  else if (y>z) f( y, z ) ;
  else      f( x, z ) ;
  printf ( "%d,%d,%d " , x, y, z);
}
```

A) 1,2,3　　B) 3,1,2　　C) 2,3,1　　D) 1,3,2

13. 有定义 int a[3][10], (*p)[10]=a; 以下不等价的一组表达式是【16】

A) a[1][1]、*(*(a+1)+1)、*(a[1]+1)

B) *a+1、*p+1、a[0]+1

C) &a[1][1]、*p+1、a[1]+1

D) a、p、&a[0]

14. 下面程序的运行结果是【17】

```
struct STU {
    char  name[10];
    int   num;
    };
void f1 (struct STU  c)
{ struct STU  b={ "SunDan" ,2042};
  c=b;c.num=2043;
```

```
}
void f2 (struct STU  *c)
{ struct STU  b={"Lili",2044};
  *c=b;
}
main()
{ struct STU  a={"Yangyang",2041},b={"Wangli",2043};
  f1(a);
  f2(&b);
  printf ( "%d,%d\n" ,a.num,b.num);
}
```

A) 2041, 2044　　B) 2041, 2043　　C) 2043, 2044　　D) 2042, 2043

15. 下面程序的运行结果是【18】

```
char fun (char x, char y)
  { if (x<y)  return x;
    return y;
}
main()
{ int  a='9', b='8', c='7';
  printf ( "%c\n" ,fun(fun(a,b),fun(b,c)));
}
```

A) 函数调用出错　　B) 8　　C) 9　　D) 7

16. 以下程序的输出结果是【19】

```
void f (int a[], int i, intj)
  { int t;
    if (i<j)
    { t=a[i]; a[i]=a[j];a[j]=t;
      f(a, i+1, j-1);
    }
```

```
}
main()
{ int  i,aa[5]={1,2,3,4,5};
  f(aa,0,4);
  for(i=0;i<5;i++) printf ("%d," ,aa[i]);
  printf ("\n");
}
```

A)5,4,3,2,1　　B) 5,2,3,4,1　　C) 1,2,3,4,5　　D) 1,5,4,3,2

17. 下面程序的输出结果是【20】

```
#include <stdio.h>
main()
{ FILE  *fp;
  int i,k=0,n=0;
  fp= fopen( "d1.dat", "w" );
  for (i=1;i<4;i++) fprintf ( fp, " %d", i);
  fclose(fp);
  fp= fopen( "d1.dat", "r" );
  fscanf (fp, "%d%d",&k,&n);
  printf( "%d,%d",k,n);
  fclose (fp);
}
```

A) 123,0　　B) 1,23　　C) 0,0　　D) 1,2

二、阅读程序写结果（共10分）

1. 以下程序的输出结果是【1】（3分）

```
#include<stdio.h>
char  *ss(char  *s)
{ return  s+strlen(s)/2;}
main()
{ char  *p, *str=" abcdefg" ;
```

```
p=ss(str); printf("%s\n",p); }
```

2. 下面程序段的输出结果是【2】(3 分)

```
int f(int a[], int n)
{ if (n>1) return a[0]+f(&a[1],n-1);
  else return a[0];}
main()
{ int aa[3]={1,2,3}, s;
  s = f( &aa[0],3 ) ;
  printf ( " %d\n" , s);
}
```

3. 下面程序段的输出结果是【3】(4 分)

```
main()
{ int sub( ) ;
  int a=2,b=2, n, arr[10]={1,2,3,4,5,6,7,8,9,10};
  sub(a,&b,arr,10);
  printf("a=%d,b=%d\n",a,b);
  printf("arr[]=");
  for(n=0;n<10;n++)
    printf("%c ",*(arr+n));
  printf("\n");
}
int sub(int x, int *z, int suba[],int i)
{ int n;
  x+=2;
  *z=x+60;
  for(n=0;n<i;n++)
   *(suba+n)+=*z;
}
```

三、阅读程序写结果并回答问题（共6分）

执行下述程序(程序1)时，键盘输入2个整数：5（回车），3（回车），其输出结果为【1】（2分）。

程序1：

```
main()
{ int num1,num2;
  int *num1_p=&num1, *num2_p=&num2, *pointer;
  printf("Input the first number: "); scanf("%d",num1_p);
  printf("Input the second number: "); scanf("%d",num2_p);
  printf("num1=%d, num2=%d\n", num1, num2);
  if( *num1_p > *num2_p )
    { pointer= num1_p,  num1_p=num2_p,  num2_p=pointer;}
  printf("*num1_p=%d, *num2_p=%d\n", *num1_p, *num2_p);
  printf("num1=%d, num2=%d\n", num1, num2);
}
```

若将程序稍加修改，如程序2所示，运行时仍输入：5（回车），3（回车），则运行结果为【2】(2分)；分析程序1和程序2的异同之处【3】（2分）

程序2：

```
main()
{ int num1,num2;
  int *num1_p=&num1, *num2_p=&num2, pointer;
  printf("Input the first number: "); scanf("%d",num1_p);
  printf("Input the second number: "); scanf("%d",num2_p);
  printf("num1=%d, num2=%d\n", num1, num2);
  if( *num1_p > *num2_p )
    {pointer= *num1_p, *num1_p=*num2_p,  *num2_p=pointer;}
  printf("*num1_p=%d, *num2_p=%d\n", *num1_p, *num2_p);
  printf("num1=%d, num2=%d\n", num1, num2);
}
```

四、程序选择填空(每空3分,共9分)

以下程序的功能是将一个整数字符串转换为一个整数，如“-1234”转换为-1234，请选择填空。

```
#include <stdio.h>
#include <string.h>
main()
{ char s[6];
  int n;
  gets(s);
  if( *s== '-' ) n = -chnum(s+1);
  else n=chnum(s);
  printf( "%d\n" ,n);
}
chnum( char  *p)
{ int num=0, k, len, j;
  len=strlen(p);
  for ( ; 【1】 ; p++ )
  { k=【2】 ;
    j=--len;
    while (【3】) {k=k*10;}
    num=num+k;
  }
  return (num);
}
```

【1】 A) *p!= '\0'　　B) *(++p)!= '\0'

C) *(p++)!= '\0'　　D) len!=0

【2】 A) *p　　B) *p+ '0'

C) *p- '0'　　D) *p-32