

【2023 年】山西省朔州市全国计算机等级 考试数据库技术真题(含答案)

学校:_____ 班级:_____ 姓名:_____ 考号:_____

一、1.选择题(10题)

1. 有两个关系 R 和 S, 分别包含 15 个和 10 个元组, 则在 $R \cup S$, $R-S$, $R \cap S$ 中不可能出现的元组数目情况是

A. 15, 5, 10 B. 18 17, 7 C. 21, 11, 4 D. 25 15, 0

2. 采用扩展关系数据模型的方法建立的数据库系统, 称作_____。

A. 对象—关系数据库系统 B. 扩展关系数据库系统 C. 拓展关系数据库系统 D. 以上都不正确

3. 下列哪一个关键码序列不符合堆的定义? (。)

A. A、C、D、G、H、M、P、Q、R、X

B. A、C、M、D、H、P、X、G、O、R

C. A、D、P、R、C、Q、X、M、H、G

D. A、D、C、M、P、G、H、X、R、Q

4. 有一个网络数据库应用系统, 其中一台计算机 A 存有 DBMS 软件、所有用户数据和应用程序, 其余各节点作为终端通过通信线路向 A 发出数据库应用请求, 这种方式属于

A. 集中式数据库系统 B. 并行数据库系统 C. 客户机/服务器数据库系统

D. 分布式数据库系统

5. 下面关于外码的描述中，不正确的是()

A. 外码体现了关系间的联系手段

B. 外码是某个关系的码

C. 外码不是任何关系的码

D. 外码是一个关系的码同时是另一个关系的属性

6. 在数据库 3 级模式结构中，外模式的个数()

A. 只有一个 B. 可以有任意多个 C. 与用户个数相同 D. 由设置的系统参

数决定

7. 计算机软件分为系统软件和应用软件，属于系统软件的是

A. 数据库管理系统 B. 操作系统 C. 财务管理系统 D. 程序语言系统

8. 关系数据库中，标识元组的属性或最小属性组称为()

A. 主键 B. 关系 C. 分量 D. 字段

9. 下列哪一个是输入设备？

A. 绘图仪 B. 打印机 C. 光笔 D. 数/模转换器

10. 在逻辑结构设计中，关系虽已达到规范化，但因某些属性过多时，

可将它分为两个或多个关系模式，这叫做

A. 模式评价 B. 优化模式 C. 合并 D. 分解

二、填空题(10题)

11. 使用 PowerDesigner 的 DataArchitect 可以对已有数据库生成物理数据模型(PDM)，这种功能通常称为_____。

12. 若一棵二叉树中度为 2 的结点数为 11，则该二叉树的叶结点数为_____。

13. 一个功能完备的网络系统应该提供一些基本的安全服务功能，这些功能包括保密性、_____、数据完整性、防抵赖和访问控制。

14.

由于视图是虚表，因此对视图的修改，最终要转换为对 **【】** 的修改。

15. 虚电路的工作方式的工作过程分为 3 个阶段：虚电路建立阶段、_____阶段与虚电路拆除阶段。

16. SMTP 的通信过程可以分成三个阶段，它们是连接_____阶段、邮件传递阶段和连接关闭阶段。

17. 服务器响应并发请求有两种实现方案：重复服务器方案和

_____。

18.组播路由协议分为_____组播路由协议和域间组播路由协议。

19. 用二维表结构表示实体以及实体间联系的数据模型称为_____数据模型。

20.死锁产生的原因是系统资源不足和 **【】**

三、单选题(10题)

21.下列不是 TCP/IP 参考模型中的应用层协议 () 。

A. A. 电子邮件协议 SMTP B. 网络文件系统 NFS C. 简单网络管理协议 SNMP D. 文件传输协议 FTP

22.下列计算机表示的信息中，不能称为“数据”的是()

A. 人的姓名 B. 图形符号 C. 计算机程序 D. 文件名

23. SQL 语言的 GRANT 和 REVOKE 语句主要用来维护数据库的 ()。

A. 安全性 B. 完整性 C. 可靠性 D. 一致性

24.在关系中选择满足某个条件的元组的关系代数运算称之为()

A. 选中运算 B. 选择运算 C. 投影运算 D. 搜索运算

25.在对全局 E-R 模型进行优化时，通常利用规范化理论中的()概念消除冗余关系。

A. 无损联接 B. 函数依赖 C. 范式 D. 逻辑蕴涵

26.概念设计的结构是得到一个与()

A. DBMS 有关的逻辑结构 B. DBMS 无关的概念模式 C. DBMS 和硬件有关的逻辑结构 D. DBMS 和硬件无关的数据结构

27.关系数据模型有许多优点，但下面所列的条目中哪一条不是它的优点()。

A. A. 结构简单

B. 高度非过程化

C. 有标准语言

D. 可表示复杂的语义

28.下列关于数据库重构的说法中，错误的是_____。

A. 数据库的重构就是对数据库的逻辑结构作必要的小范围改变

B. 当存储空间的分配过于零散，导致存取效率下降，所以必须重构数据库

C. 数据库重构包括重新设计数据库及其应用程序

D. 如果能够重构数据库，尽量不要重新设计数据库及相应的应用程序

29. 下面哪些是参数调整中需要关注的对象。

A. 内存分配 B. 磁盘 I/O C. 资源竞争 D. 以上全部

30. 下列存储管理方案中，() 允许动态扩充内存容量。

A. 固定分区 B. 可变分区 C. 页式 D. 虚拟页式

四、C 语言程序设计题(10题)

31. 程序 test C 的功能是：利用以下所示的简单迭代方法求方程

$\cos(x) - x = 0$ 的一个实根。迭代公式： $X_{n+1} = \cos(x_n)$ (n是迭代次数)

迭代步骤如下：

(1)取 X_1 初值为 0. 0;

(2) $X_0 = X_1$ ，把 X_1 的值赋给 x_0 ;

(3) $X_1 = \cos(x_0)$ 求出一个新的 x_1 ;

(4)若 x_0, X_1 的绝对值小于 0. 000001, 执行步骤(5), 否则执行步骤(2);

(5)所求 X_1 就是方程 $\cos(X) - X = 0$ 的一个实根，作为函数值返回。

请编写函数 countValue 实现程序的功能，最后 main 函数调用函数 writeDAT 把结果输出到文件 Out. dat 中。注意：部分源程序存放在 testC 文件中。

请勿改动主函数 main 和输出数据函数 writeDAT 的内容。

```

1 | #include <math.h> //表示这个程序中要引用 vc6.0 的数学库函数
2 | #include <stdio.h> //表示这个程序要引用 vc6.0 的标准输入输出函数
3 | void writeDAT(); //writeDAT() 函数的说明语句
4 | double countValue()
5 | {
6 |
7 | }
8 | main()
9 | {

```

```

10 | printf("实根=%f\n", countValue()); //在屏幕上显示方程的一个实根
11 | printf(" %f\n", cos(countValue()) -countValue()); //在屏幕显示求得这个实根的精度
12 | writeDAT(); //把结果输出到文件 out.dat 中
13 | }
    /* 函数 writeDAT()把结果输出到文件 out.dat 中*/
14 | void writeDAT()
15 | {
16 | FILE *wf; //定义文件指针变量 wf
17 | wf=fopen("out.dat","w"); //以只写的方式打开 out.dat,并使 wf 指向这个文件
18 | fprintf(wf, "%f\n", countValue()); //把求得的实根写入到文件 out.dat
19 | fclose(wf); //关闭文件 out.dat
20 | }

```

32.函数 ReadDat 实现从文件 in. dat中读取一篇英文文章存入到字符串数组 XX 中；请编制函数 SortCharD, 其函数的功能是：以行为单位对字符按 ASCII 码从大到小的顺序进行排序，排序后的结果仍按行重新存入字符串数组 XX 中。最后 main 函数调用函数 WriteDat把结果 XX 输出到文件 out. dat中。

例：原文：dAe , BfC .

CCbbAA 结果：fedCBA ., bbCCAA 原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

注意：部分源程序存放在 test C 文件中。

请勿改动主函数 main、读数据函数 ReadDat 和输出数据函数 WriteDat 的内容。

```

1  #include <stdio.h>
2  #include <string.h>
3  char xx[50][80];
4  unsigned int maxline = 0; /* 文章的总行数 */
5  int ReadDat(void);
6  void WriteDat(void);
7  void SortCharD(void)
8  {
9
10 }
11 void main()
12 {
13     if(ReadDat())
14     {
15         printf("数据文件 in.dat 不能打开! \n\007");
16         return;
17     }
18     SortCharD();
19     WriteDat();
20 }
    /*从文件 in.dat 中读取一篇英文文章存入到字符串数组 xx 中*/
21 int ReadDat(void)
22 {
23     FILE *fp;

```

```

24     int i = 0;
25     char *p;
26     if((fp=fopen("in.dat","r"))==NULL)
27         return 1; //如果文件 in.dat 内容为空, 则返回1
    /*当从文件 in.dat 中读取 79 个字符存入 xx[i] 没有错误发生时, 执行循环体中的语句*/
28     while(fgets(xx[i],80,fp) != NULL)
29     {
30         p = strchr(xx[i], '\n'); //在字符串 xx[i] 中查找回车换行符第一个匹配的位置
31         if(p) *p = 0; //如果 p 的值不为零, 则将其值设置为 0
32         i++; //行变量加 1
33     }
34     maxline = i; //把文章的总行数赋给变量 maxline
35     fclose(fp);
36     return 0;
37 }
    /*把结果 xx 输出到文件 out.dat 中*/
38 void WriteDat(void)
39 {
40     FILE *fp;
41     unsigned int i;
42     fp = fopen("out.dat", "w");
43     for(i = 0; i < maxline; i++)
44     {
45         printf("%s\n", xx[i]);
46         fprintf(fp, "%s\n", xx[i]);
47     }
48     fclose(fp);
49 }

```


33.

函数 ReadDat 实现从文件 in. dat 中读取一篇英文文章存入到字符串数组 XX 中，请编制函数 ConvertCharD，其函数的功能是：以行为单位把字符串中的所有小写字母改写成该字母的上一个字母，如果是字母 a，则改写成字母 z，其他字符保持不变。把已处理的字符串仍按行重新存入字符串数组 xx 中。最后 main 函数调用函数 WriteDat 把结果 xx 输出到文件 out. dat 中。例：原文：Adb . Bcdza

abck. LLhj 结果：Aca. Bbcyzzabj LLgi 原始数据文件存放的格式是：每行的宽度均小于 80 个字符，含标点符号和空格。

注意：部分源程序存放在 test.c 文件中。

请勿改动主函数 main、读函数 ReadDat0 和输出函数 WriteDat 的内容。

```
1  #include <stdio.h>
2  #include <string.h>
3  char xx[50][80];
```

```

4  int maxline = 0; /* 文章的总行数 */
5  int ReadDat(void);
6  void WriteDat(void);
7  /*以行为单位把字符串中的所有小写字母改写成该字母的上一个字母, 如果是字母 a, 则改写成字母 z. 其他字符保持不变*/
8  void ConvertCharD(void)
9  {
10 }
11 void main()
12 {
13     if(ReadDat())
14     {
15         printf("数据文件 in.dat 不能打开! \n\007");
16         return;
17     }
18     ConvertCharD();
19     WriteDat();
20 }
21 /*从文件 in.dat 中读取一篇英文文章存入到字符串数组 xx 中*/
22 int ReadDat(void)
23 {
24     FILE *fp;
25     int i = 0;
26     char *p;
27     if((fp=fopen("in.dat", "r"))==NULL)
28         return 1;
29     while(fgets(xx[i], 80, fp) != NULL)
30     {
31         p = strchr(xx[i], '\n');
32         if(p) *p = 0;
33         i++;
34     }
35     maxline = i;
36     fclose(fp);
37     return 0;
38 }
39 /*把结果 xx 输出到文件 out.dat 中*/
40 void WriteDat(void)
41 {
42     FILE *fp;
43     int i;
44     fp = fopen("out.dat", "w");
45     for(i = 0; i < maxline; i++)
46     {
47         printf("%s\n", xx[i]);
48         fprintf(fp, "%s\n", xx[i]);
49     }
50     fclose(fp);
51 }

```

34. 已知在文件 in.dat 中存有 100 个产品销售记录, 每个产品销售记录由产品代码 dm(字符型 4 位), 产品名称 me(字符型 10 位), 单价 dj 整

型), 数量 `sl`(整型), 金额 `je`(长整型)五部分组成。其中: 金额=单价*数量。函数 `ReadDat` 读取这 100 个销售记录并存入结构数组 `sell`中。请编制函数 `SortDat`, 其功能要求: 按产品名称从大到小进行排列, 若产品名称相同, 则按金额从大到小进行排列, 最终排列结果仍存入结构数组 `sell`中。最后 `main`函数调用函数 `WriteDat`把结果输出到文件 `out.dat`中。

提示: 若中间变量为 `PRO temp`, 则可以直接使用结构赋值语句进行解

题: 产品名称比较请用函数 `strcmp`进行解题。例如: `sell[i]=temp`

注意: 部分源程序存放在 `test.c` 文件中。

请勿改动主函数 `main`、读函数 `ReadDat` 和输出函数 `WriteDat`的内容。

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #define MAX 100
5  typedef struct
6  {
7      char dm[5]; /* 产品代码 */
8      char mc[11]; /* 产品名称 */
9      int dj; /* 单价 */
10     int sl; /* 数量 */
11     long je; /* 金额 */
12 } PRO;
13 PRO sell[MAX];
14 void ReadDat();
15 void WriteDat();
16 void SortDat()
17 {
18 }
19 }
20 void main()
21 {
22     memset(sell, 0, sizeof(sell));
23     ReadDat();
24     SortDat();
25     WriteDat();
26 }
27 void ReadDat()
28 {
29     FILE *fp;
30     char str[80], ch[11];
31     int i;
32     fp = fopen("in.dat", "r");
33     for(i = 0; i < 100; i++)
34     {
35         fgets(str, 80, fp);
36         memcpy(sell[i].dm, str, 4);
```

```

37     memcpy(sell[i].mc, str + 4, 10);
38     memcpy(ch, str+14, 4); ch[4]=0;
39     sell[i].dj = atoi(ch);
40     memcpy(ch, str+18, 5); ch[5]=0;
41     sell[i].sl = atoi(ch);
42     sell[i].je = (long)sell[i].dj * sell[i].sl;
43 }
44 fclose(fp);
45 }
46 void WriteDat()
47 {
48     FILE *fp;
49     int i;
50     fp = fopen("out.dat", "w");
51     for(i = 0; i < 100; i++)
52     {
53         fprintf(fp, "%s %s %4d %5d %10ld\n", sell[i].dm, sell[i].mc, sell[i].dj, sell[i].sl,
54             sell[i].je);
55     }
56     fclose(fp);
57 }

```

35. 将文件 in.dat 中的 200 个整数读至数组 XX 中。请编制 jsValue 函数，要求：求出数组 XX 中的数值为奇数的个数 cnt1 和数值为偶数的个数 cnt2 以及数组 XX 下标为奇数的元素值的算术平均值 pj。

结果 cnt1, cnt2, pj 输出到 out.dat 中。注意：部分源程序存在 test.c 文件中。请勿改动数据文件 in.dat 中的任何数据、主函数 main、读函数 read_da 和输出函数 writeDat 的内容。

```

1  #include <stdio.h>
2  #define N 200
3  int cnt1, cnt2, xx[N];
4  float pj; //平均值
5  void writeDat();
6  void jsValue()
7  {
8
9  }
/* read_dat(int xx[N])函数将文件in.dat中的200个整数读至数组xx中*/
10 void read_dat(int xx[N])
11 {
12     int i,j;
13     FILE *fp;
14     fp = fopen("in.dat","r");
15     for(i = 0; i < 20; i++)
16     {
17         for(j = 0; j < 10; j++)
18         {
19             fscanf(fp, "%d,", &xx[i*10+j]); //从文件in.dat中读取一个整数存入数组元素xx[i*10+j]中
20             printf("%d ", xx[i*10+j]);
21         }
22         printf("\n");
23     }
24     fclose(fp);
25 }
26 void main()
27 {
28     read_dat(xx);

```

```

29     jsValue();
30     printf("\n\ncnt1=%d, cnt2=%d, pj=%6.2f\n", cnt1, cnt2, pj);
31     writeDat();
32 }
/* 将计算结果cnt1, cnt2, pj输出到out.dat中*/
33 void writeDat()
34 {
35     FILE *fw;
36     fw = fopen("out.dat","w");
37     fprintf(fw, "%d\n%d\n%6.2f\n", cnt1, cnt2, pj); //cnt1, cnt2, pj写入到out.dat中
38     fclose(fw);
39 }

```

36. 在文件 in.dat 中已存有 200 个正整数，且每个数均在 1 000 至 9999 之间。函数 readDat 是读取这 200 个数存放到数组 aa 中。请编制函数 jsSort 其函数的功能是：要求按每个数的后三位的大小进行降序排列，如果出现后三位相等的数值，则对这些数值按原始 4 位数据进行升序排

列，将排序后的前 10 个数存入数组 bb 中。

最后调用函数 writeDat 把结果 bb 输出到文件 out. dat 中。

```
例：处理前 9012 5099 6012 7025 8088
      处理后 5099 8088 7025 6012 9012
注意：部分源程序存在 test.c 文件中。
请勿改动数据文件 in.dat 中的任何数据、主函数 main()、读函数 readDat()和写函数 writeDat()的内容。
1  #include <stdio.h>
2  #include <string.h>
3  void readDat();
4  void writeDat();
5  int aa[200], bb[10];
6  void jsSort()
7  {
8
9  }
10 void main()
11 {
12     readDat();
13     jsSort();
14     writeDat();
15 }
16 void readDat()
17 {
18     FILE *in;
19     int i;
20     in = fopen("in.dat", "r");
21     for(i = 0; i < 200; i++)
22         fscanf(in, "%d,", &aa[i]);
23     fclose(in);
24 }
25 void writeDat()
26 {
27     FILE *out;
28     int i;
29     out = fopen("out.dat", "w");
30     for(i = 0; i < 10; i++)
31     {
32         printf("i=%d,%d\n", i + 1, bb[i]);
33         fprintf(out, "%d\n", bb[i]);
34     }
35     fclose(out);
36 }
```

37. 已知数据文件 in. dat 中存有 200 个四位数，并已调用读函数 readDat 把这些数存入数组 a 中，请考生编制一函数 jsVal，其功能是：依次从数

组 a 中取出一个四位数，如果该四位数小于该四位数以后的连续五个数且该数是偶数 (该四位数以后不满五个数，则不统计)，则统计出满足此条件的个数 cnt，并把这些四位数存入数组 b 中，而后对数组 b 进行升序排序，最后调用写函数 writeDat 把结果 cnt 以及数组 b 中符合条件的四位数输出到 Out. dat 文件中。

注意：部分源程序存在 test c 文件中。

程序中已定义数组：a[200] b[200]，已定义变量：cnt

请勿改动数据文件 in. dat 中的任何数据、主函数 main、读函数 readDat 和写函数 writeDat 的内容。

```
1  #include <stdio.h>
2  #define MAX 200 //定义宏变量MAX 其值等于 200
3  int a[MAX], b[MAX], cnt = 0; //其初始值为 0, 定义全局整型一维数组 a[MAX], b[MAX] 和变量 cnt
4  void writeDat(); //writeDat() 函数的说明语句
5  void jsVal()
6  {
7
8  }
9  /* readDat () 从数据文件 in.dat 中读取 200 个四位数存入数组 a 中 */
10 void readDat()
11 {
12     int i;
13     FILE *fp;
14     fp = fopen("in.dat", "r");
```



```

14     for(i = 0; i < MAX; i++)
15         fscanf(fp, "%d", &a[i]);
16     fclose(fp);
17 }
18 void main()
19 {
20     int i;
21     readDat();
22     jsVal(); //调用 jsVal()函数,实现题目要求的功能
23     printf("满足条件的数=%d\n", cnt);
24     for(i = 0; i < cnt; i++)
25         printf("%d ", b[i]);
26     printf("\n");
27     writeDat();
28 }
/*函数 writeDat()把结果 cnt 以及数组 b 中符合条件的四位数输出到 out.dat 文件中*/
29 void writeDat()
30 {
31     FILE *fp;
32     int i;
33     fp = fopen("out.dat", "w");
34     fprintf(fp, "%d\n", cnt);
35     for(i = 0; i < cnt; i++)
36         fprintf(fp, "%d\n", b[i]);
37     fclose(fp);
38 }

```

38.程序 test c 的功能是：计算出自然数 SIX 和 NINE 满足条件 $SIX+SIX+SIX=NINE+NINE$ 的个数 cnt，以及满足此条件的所有 SIX 与 NINE 的和 SUM。请编写函数 countValue实现程序的要求，最后 main 函数调用函数 writeDAT 把结果 cnt和 sum，输出到文件 Out. dat中。其中的 S, I, X, N, E 各代表一个十进制数字，允许代表的数字相同，但 s 和 N 不能为 0。例如：944+944+944=1416+1416 注意：部分源程序存放在 test c 文件中。

请勿改动主函数 main 和输出数据函数 writeDAT 的内容。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/465240143302011220>