**[【计算机二级C语言】卷010](https://www.cnblogs.com/mjios/p/12452746.html)**

**选择题**

**公共知识**

【1】下列叙述中正确的是()。

〖A〗循环队列是队列的一种链式存储结构

〖B〗循环队列是队列的一种顺序存储结构

〖C〗循环队列是非线性结构

〖D〗循环队列是一种逻辑结构

在实际应用中, 队列的顺序存储结构一般采用循环队列的形式。

【2】下列关于线性链表的叙述中, 正确的是()。

〖A〗各数据结点的存储空间可以不连续, 但它们的存储顺序与逻辑顺序必须一致

〖B〗各数据结点的存储顺序与逻辑顺序可以不一致, 但它们的存储空间必须连续

〖C〗进行插入与删除时, 不需要移动表中的元素

〖D〗以上说法均不正确

一般来说, 在线性表的链式存储结构中, 各数据结点的存储序号是不连续的, 并且各结点在存储空间中的位置关系与逻辑关系也不一致。  
线性链表中数据的插入和删除都不需要移动表中的元素, 只需改变结点的指针域即可。

【3】一棵二叉树共有25个结点, 其中5个是叶子结点, 则度为1的结点数为()

〖A〗16

〖B〗10

〖C〗6

〖D〗4

根据二叉树的性质3:在任意一棵二叉树中, 度为0的叶子结点总是比度为2的结点多一个, 所以本题中度为2的结点是5 - 1 = 4个, 所以度为1的结点的个数是25 - 5 - 4 = 16个。

【4】在下列模式中, 能够给出数据库物理存储结构与物理存取方法的是()。

〖A〗外模式

〖B〗内模式

〖C〗概念模式

〖D〗逻辑模式

数据库系统的三级模式是概念模式、外模式和内模式。  
概念模式是数据库系统中全局数据逻辑结构的描述, 是全体用户公共数据视图。  
外模式也称子模式或用户模式, 它是用户的数据视图, 给出了每个用户的局部数据描述。  
内模式又称物理模式, 它给出了数据库物理存储结构与物理存取方法, 所以选择B项。

【5】在满足实体完整性约束的条件下()。

〖A〗一个关系中应该有一个或多个候选关键字

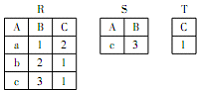
〖B〗一个关系中只能有一个候选关键字

〖C〗一个关系中必须有多关键字个候选

〖D〗一个关系中可以没有候选关键字

实体完整性约束要求关系的主键中属性值不能为空值, 所以选择A)。

【6】有三个关系R、S和T如下：



则由关系R和S得到关系T的操作是()。

〖A〗自然连接

〖B〗交

〖C〗除

〖D〗并

如果S = T / R, 则S称为T除以R的商。  
在除运算中S的域由T中那些不出现在R中的域所组成, 对于S中的任一有序组, 由它与关系R中每个有序组所构成的有序组均出现在关系T中。  
所以本题选择C)。

【7】下面描述中, 不属于软件危机表现的是()。

〖A〗软件过程不规范

〖B〗软件开发生产率低

〖C〗软件质量难以控制

〖D〗软件成本不断提高

软件危机主要表现在：软件需求的增长得不到满足; 软件开发成本和进度无法控制; 软件质量难以保证; 软件不可维护或维护程度非常低; 软件的成本不断提高; 软件开发生产率的提高赶不上硬件的发展和应用需求的增长。  
所以选择A)。

【8】下面不属于需求分析阶段任务的是()。

〖A〗确定软件系统的功能需求

〖B〗确定软件系统的性能需求

〖C〗需求规格说明书评审

〖D〗制定软件集成测试计划

需求分析阶段的工作有：需求获取; 需求分析; 编写需求规格说明书; 需求评审, 所以选择D)。

【9】在黑盒测试方法中, 设计测试用例的主要根据是()。

〖A〗程序内部逻辑

〖B〗程序外部功能

〖C〗程序数据结构

〖D〗程序流程图

黑盒测试是对软件已经实现的功能是否满足需求进行测试和验证, 黑盒测试完全不考虑程序内部的逻辑结构和内部特性, 只根据程序的需求和功能规格说明, 检查程序的功能是否符合它的功能说明, 所以本题选择B)。

【10】在软件设计中不使用的工具是()。

〖A〗系统结构图

〖B〗PAD图

〖C〗数据流图(DFD图)

〖D〗程序流程图

系统结构图是对软件系统结构的总体设计的图形显示。  
在需求分析阶段, 已经从系统开发的角度出发, 把系统按功能逐次分割成层次结构, 是在概要设计阶段用到的。  
PAD图是在详细设计阶段用到的。  
程序流程图是对程序流程的图形表示, 在详细设计过程中用到。  
数据流图是结构化分析方法中使用的工具, 它以图形的方式描绘数据在系统中流动和处理的过程, 由于它只反映系统必须完成的逻辑功能, 所以它是一种功能模型, 是在可行性研究阶段用到的而非软件设计时用到, 所以选择C)。

**专业知识**

【11】以下选项中关于程序模块化的叙述错误的是

〖A〗可采用自底向上、逐步细化的设计方法把若干独立模块组装成所要求的程序

〖B〗把程序分成若干相对独立、功能单一的模块, 可便于重复使用这些模块

〖C〗把程序分成若干相对独立的模块, 可便于编码和调试

〖D〗可采用自顶向下、逐步细化的设计方法把若干独立模块组装成所要求的程序

程序模块化思想中, 可以采用自顶向下、逐步细化的方法。  
所以选项A中"自底向上"的说法是错误的。

【12】以下叙述中正确的是

〖A〗用C语言编写的程序只能放在一个程序文件中

〖B〗C程序书写格式严格, 要求一行内只能写一个语句

〖C〗C程序中的注释只能出现在程序的开始位置和语句的后面

〖D〗C程序书写格式自由, 一个语句可以写在多行上

C语言编写的程序可以放置于多个程序文件中, 所以A错误。  
C程序中的一行可以有多条语句, 所以B错误。  
C语言中的注释语句可以与原语句放在一行也可以不放在一行, 所以C错误。

【13】以下不合法的数值常量是

〖A〗8.0E0.5

〖B〗1e1

〖C〗011

〖D〗0xabcd

A选项中E后面的指数必须为整型数据, 所以A错误。  
C选项中011表示的是八进制常量, 0xabcd表示的是十六进制常量。

【14】以下关于C语言数据类型使用的叙述中错误的是

〖A〗若要处理如"人员信息"等含有不同类型的相关数据, 应自定义结构体类型

〖B〗若要保存带有多位小数的数据, 可使用双精度类型

〖C〗若只处理"真"和"假"两种逻辑值, 应使用逻辑类型

〖D〗整数类型表示的自然数是准确无误差的

C语言中没有逻辑类型, 所以C错误。  
若要保存带有多位小数的数据, 可以用单精度类型也可以用双精度类型。  
处理包含不同类型的相关数据可以定义为结构体类型。  
整数类型可以无误差的表示自然数。

【15】设有定义：

复制代码

C

int k = 0;

以下选项的四个表达式中与其他三个表达式的值不相同的是

〖A〗++k

〖B〗k += 1

〖C〗k++

〖D〗k + 1

++k, 先使得k的值自增1后再使用, k++是先取得k的值再将k的值自增1, 所以C选项中表达式的值为0, 而其他三个表达式的值均为1, 所以选择C。

【16】有如下程序段：

复制代码

C

|  |  |
| --- | --- |
|  | int x = 12; |
|  | double y = 3.141593; |
|  | printf("%d%8.6f", x, y); |

其输出结果是

〖A〗123.141593

〖B〗12 3.141593

〖C〗12, 3.141593

〖D〗123.1415930

printf函数中格式字符"%8.6"的含义是：变量y的输出宽度为8(包括小数点), 小数点后面保留6位小数, 且右对齐。  
所以选择A。

【17】已知字符'A'的ASCII代码值是65, 字符变量c1的值是'A', c2的值是'D'。  
则执行语句

复制代码

C

printf("%d,%d", c1, c2 - 2);

的输出结果是

〖A〗65,68

〖B〗A,68

〖C〗A,B

〖D〗65,66

打印时以%d整型格式打印输出, 所以字符标量c1的值打印出来就是65, 从c2 - 2的值打印出来就是68 - 2, 即66。  
所以选择D。

【18】以下选项中, 当x为大于1的奇数时, 值为0的表达式是

〖A〗x % 2 == 0

〖B〗x / 2

〖C〗x % 2 != 0

〖D〗x % 2 == 1

算术运算符的优先级高于关系运算符的优先级, 所以当x为大于1的奇数时, A选项的值为假, 即0。  
选项的值为不定值, 但绝对不是0 ? C选项的值为真, 即1。  
D选项的值为真, 即1。

【19】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int x; |
|  | scanf("%d", &x); |
|  | if (x <= 3); |
|  | else if (x != 10) |
|  | printf("%d\n", x); |
|  | } |

程序运行时, 输入的值在哪个范围才会有输出结果

〖A〗小于3的整数

〖B〗不等于10的整数

〖C〗大于3或等于10的整数

〖D〗大于3且不等10的整数

根据代码可知如果x的值小于3, 执行空语句, 不进行打印操作。  
如果x的值大于3, 再判断x的值是否不等于10, 如果不等于10就打印x的值, 所以选择D选项。

【20】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int a = 7; |
|  | while (a--); |
|  | printf("%d\n", a); |
|  | } |

程序运行后的输出结果是

〖A〗0

〖B〗-1

〖C〗1

〖D〗7

a--的含义是先用后减1, 所以当a的值为0时, 跳出循环, 但是仍要执行a--, a的值变为-1。

【21】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | char b, c; |
|  | int i; |
|  | b = 'a'; |
|  | c = 'A'; |
|  | for (i = 0; i < 6; i++) { |
|  | if (i % 2) |
|  | putchar(i + b); |
|  | else |
|  | putchar(i + c); |
|  | } |
|  | printf("\n"); |
|  | } |

程序运行后的输出结果是

〖A〗abcdef

〖B〗ABCDEF

〖C〗aBcDeF

〖D〗AbCdEf

函数的含义是如果i的值能被2整除, 以字符格式打印输出i + c的值, 如果i的值不能被2整除打印i + b的值。  
第一次i值为0, 执行else语句打印字符'A', 第二次i值为1, 打印1 + b的值即字符'b', 第三次i值为2, 打印字符'C', 依此类推选择D选项。

【22】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int i, j, x = 0; |
|  | for (i = 0; i < 2; i++) { |
|  | x++; |
|  | for (j = 0; j <= 3; j++) { |
|  | if (j % 2) |
|  | continue; |
|  | x++; |
|  | } |
|  | x++; |
|  | } |
|  | printf("x=%d\n", x); |
|  | } |

程序执行后的输出结果是

〖A〗x=4

〖B〗x=8

〖C〗x=6

〖D〗x=12

coutinue的作用是跳出循环体中剩余的语句而进行下一次循环。  
第一次执行外循环i的值为0, 执行x++, x的值变为1, 第一次执行内层循环j的值为0, 不满足if条件, 执行x++, x的值变为2, 第二次内循环j的值为1, if条件成立, 跳出本次循环, 第三次执行内循环j的值为2, 不满足if条件, x的值变为3, 第四次执行内循环j的值为3满足条件跳出本次内循环, x的值加1, 即为4, 第一次外循环结束。  
第二次执行外循环时, 同理, i的值被加了, 4次, 变为8, 所以选择B选项。

【23】若函数调用时的实参为变量时, 以下关于函数形参和实参的叙述中正确的是

〖A〗函数的形参和实参分别占用不同的存储单元

〖B〗形参只是形式上的存在, 不占用具体存储单元

〖C〗同名的实参和形参占同一存储单元

〖D〗函数的实参和其对应的形参共占同一存储单元

本题重点考查的是函数的形参和实参。  
在定义函数时函数名后面括弧中的变量名称为"形式参数"(简称形参), 在主调函数中调用一个函数时, 函数名后面括弧中的参数(可以是一个表达式)称为"实际参数"(简称实参)。  
C语言规定, 实参变量对形参变量的数据传递是"值传递", 即单向传递, 只由实参传给形参, 而不能由形参传回来给实参。  
在内存中, 实参单元与形参单元是不同的单元。  
因此A选项正确。

【24】设有以下函数：

复制代码

C

|  |  |
| --- | --- |
|  | void fun(int n, char \*s) { |
|  | …… |
|  | } |

则下面对函数指针的定义和赋值均正确的是

〖A〗void (\*pf)(int , char); pf = &fun;

〖B〗void \*pf(); pf = fun;

〖C〗void \*pf(); \*pf = fun;

〖D〗void (\*pf)(int , char \*); pf = fun;

函数的参数可以是指针类型。  
它的作用是将一个变量的地址传送到另一个函数中。  
函数名代表函数的入口地址, 指向函数的指针应该定义为void(\*pf)()。  
如果定义为void \* pf(), 则表示函数pf返回值为一个基类型为void的指针。  
因此D选项正确。

【25】若要求定义具有10个int型元素的一维数组a, 则以下定义语句中错误的是

〖A〗#define n 5 int a[2 \* n];

〖B〗int n = 10, a[n];

〖C〗int a[5 + 5];

〖D〗#define N 10 int a[N];

一维数组的定义方式为：类型说明符 数组名[常量表达式]; 注意定义数组时, 元素个数不能是变量。  
因此应该选B选项。

【26】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int i, t[][3] = { 9, 8, 7, 6, 5, 4, 3, 2, 1 }; |
|  | for (i = 0; i < 3; i++) |
|  | printf("%d ", t[2 - i][i]); |
|  | } |

程序执行后的输出结果是

〖A〗3 5 7

〖B〗7 5 3

〖C〗3 6 9

〖D〗7 5 1

二维数组t[][3]实际上指t[3][3] = { { 9, 8, 7 }, { 6, 5, 4 }, { 3, 2, 1 } }, 通过循环语句for语句可以得到i = 0, t[2][0] = 3、i = 1, t[1][1] = 5、i = 2, t[0][2] = 7、i = 3时循环结束即t[2][0] = 3、t[1][1] = 5、t[0][2] = 7。  
因此A选项正确。

【27】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void fun(int \*s, int n1, int n2) { |
|  | int i, j, t; |
|  | i = n1; |
|  | j = n2; |
|  | while (i < j) { |
|  | t = s[i]; |
|  | s[i] = s[j]; |
|  | s[j] = t; |
|  | i++; |
|  | j--; |
|  | } |
|  | } |
|  | void main() { |
|  | int a[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 }, k; |
|  | fun(a, 0, 3); |
|  | fun(a, 4, 9); |
|  | fun(a, 0, 9); |
|  | for (k = 0; k < 10; k++) |
|  | printf("%d", a[k]); |
|  | printf("\n"); |
|  | } |

程序的运行结果是

〖A〗4321098765

〖B〗5678901234

〖C〗0987654321

〖D〗0987651234

该程序首先给一维数组赋值, 然后三次调用fun函数, 其中fun(a, 0, 3); 功能是将一维数组中第1个元素和第4个元素互换, 第2个元素和第3个元素互换; 其中fun(a, 4, 9); 功能是将一维数组中第5个元素和第10个元素互换, 第6个和第9个元素互换, 第7个元素和第8个元素互换; 其中fun(a, 0, 9); 功能是将将一维数组中第1个元素和第10个元素互换, 第2个元素和第9个元素互换……依次类推。  
因此B选项正确。

【28】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int a[4][4] = { { 1, 4, 3, 2 }, { 8, 6, 5, 7 }, { 3, 7, 2, 5 }, { 4, 8, 6, 1 } }, i, k, t; |
|  | for (i = 0; i < 3; i++) |
|  | for (k = i + 1; k < 4; k++) |
|  | if (a[i][i] < a[k][k]) { |
|  | t = a[i][i]; |
|  | a[i][i] = a[k][k]; |
|  | a[k][k] = t; |
|  | } |
|  | for (i = 0; i < 4; i++) |
|  | printf("%d,", a[0][i]); |
|  | } |

程序运行后的输出结果是

〖A〗1,1,2,6,

〖B〗6,2,1,1,

〖C〗6,4,3,2,

〖D〗2,3,4,6,

首先对二维数组进行赋值操作, a[0][0]、a[0][1]、…、a[3][2]、a[3][3]的值为1、4、…、6、1。  
通过for嵌套循环语句和if条件语句, 对二维数组对角线元素进行由大到小的排序操作, 程序最后通过for语句输出二维数组第1行的4个元素。  
因此C选项正确。

【29】以下选项中正确的语句组是

〖A〗char \*s; s = { "BOOK!" };

〖B〗char \*s; s = "BOOK!";

〖C〗char s[10]; s = "BOOK!";

〖D〗char s[]; s = "BOOK!";

A选项去掉大括号就正确了; C选项和D选项应在定义时赋初值。  
因此B选项正确。

【30】若有以下定义和语句：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | char s1[10] = "abcd!", \*s2 = "\n123\\"; |
|  | printf("%d %d\n", strlen(s1), strlen(s2)); |

则输出结果是

〖A〗10 7

〖B〗10 5

〖C〗5 5

〖D〗5 8

strlen函数返回字符串的长度, 求字符串长度时, 遇到结束标志'\0'为止, 但是长度不包括结束标识。  
字符数组s1的后5个元素没有赋值, 都为'\0', 即"abcd!"后为'\0'。  
所以strlen(s1)的值为5。  
字符指针s2所指向的字符串中, \n为转义字符换行符, 表示1个字符, \\也为转义字符, 代表\, 也是1个字符, 其后为字符串结束标识'\0'; 所以strlen(s2)的值也为5。  
因此C选项正确。

【31】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void fun1(char \*p) { |
|  | char \*q; |
|  | q = p; |
|  | while (\*q != '\0') { |
|  | (\*q)++; |
|  | q++; |
|  | } |
|  | } |
|  | void main() { |
|  | char a[] = { "Program" }, \*p; |
|  | p = &a[3]; |
|  | fun1(p); |
|  | printf("%s\n", a); |
|  | } |

程序执行后的输出结果是

〖A〗Prphsbn

〖B〗Prohsbn

〖C〗Progsbn

〖D〗Program

在main函数中, 定义了一个字符数组a和字符指针p, 字符指针p指向数组a的第4个字符'g', 调用fun1函数。  
在fun1函数中, 定义了字符指针q, 使q = p, 即去也指向数组a的第4个字符, (\*q)++是将q所指的字符加1, 即'g' + 1 = 'h', 语句q++使指针移到下一个字符, while循环即是将后面的字符都逐个加1。  
故本题答案为B选项。

【32】有以下函数：

复制代码

C

|  |  |
| --- | --- |
|  | int aaa(char \*s) { |
|  | char \*t = s; |
|  | while (\*t++); |
|  | t--; |
|  | return (t - s); |
|  | } |

以下关于aaa函数功能叙述正确的是

〖A〗将串s复制到串t

〖B〗比较两个串的大小

〖C〗求字符串s的长度

〖D〗求字符串s所占字节数

本题重点考查的知识点while循环语句的应用。  
aaa()函数中, 首先定义了一个字符指针t指向形参s, 然后通过一个while循环让指针 t不断递增, 直到t指向字符串结束标识处。  
当t指向结束标识处时, 由于后缀++运算符的原因, 它还会被再递增1, 所以接下来的t--; 语句让它回到结束标识处。  
最后返回t - s, s还是指向字符串第1个字符处, 而t指向了字符串结尾, 故返回值为字符串的长度值。  
因此C选项正确。

【33】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | int fun(int a, int B) { |
|  | if (b == 0) |
|  | return a; |
|  | else |
|  | return (fun(--a, --b)); |
|  | } |
|  | void main() { |
|  | printf("%d\n", fun(4, 2)); |
|  | } |

程序的运行结果是

〖A〗1

〖B〗2

〖C〗3

〖D〗4

本题重点考查函数的递归调用; 题目中给出的fun()函数直接调用了自身, 所以是一个递归函数。  
其功能是：当参数b为0时, 返回参数a的值, 否则返回fun(--a, --b)的值。  
从这里可以看出, 当b不断递减时, a也不断递减, 直到b为0时返回a的值。  
那么a递减的总值就是b原来的值。  
所以整个递归函数的作用就是返回a - b的值。  
因此B选项正确。

【34】以下选项中叙述错误的是

〖A〗C程序函数中定义的自动变量, 系统不自动赋确定的初值

〖B〗在C程序的同一函数中, 各复合语句内可以定义变量, 其作用域仅限本复合语句内

〖C〗C程序函数中定义的赋有初值的静态变量, 每调用一次函数, 赋一次初值

〖D〗C程序函数的形参不可以说明为static型变量

A选项描述正确, 自动变量未赋初值, 为随机值; B选项描述正确, 变量定义除在函数开始位置外, 在复合语句开始也可以; C选项描述是错误的, 函数内的静态变量, 只在第一次调用时候赋值, 以后调用保留上次值; D选项描述也正确, 形参属于局部变量, 占用动态存储区, 而static型变量占用静态存储区。  
因此C选项正确。

【35】有以下程序段：

复制代码

C

|  |  |
| --- | --- |
|  | struct st { |
|  | int x; |
|  | int \*y; |
|  | } \* pt; |
|  | int a[] = { 1, 2 }, b[] = { 3, 4 }; |
|  | struct st c[2] = { 10, a, 20, b }; |
|  | pt = c; |

以下选项中表达式的值为11的是

〖A〗++pt->x

〖B〗pt->x

〖C〗\*pt->y

〖D〗(pt++)->x

本题考查结构体数组初始化以及结构体变量的引用, 题目中定义了一个结构体数组c并初始化, 指针pt指向c的第一个元素, 那么pt->x为10, ++pt->x为11, \*pt->y为1, (pt++)->x为20, 所以答案为A。

【36】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | struct S { |
|  | int n; |
|  | int a[20]; |
|  | }; |
|  | void f(int \*a, int n) { |
|  | int i; |
|  | for (i = 0; i < n - 1; i++) |
|  | a[i] += i; |
|  | } |
|  | void main() { |
|  | int i; |
|  | struct S s = { 10, { 2, 3, 1, 6, 8, 7, 5, 4, 10, 9 } }; |
|  | f(s.a, s.n); |
|  | for (i = 0; i < s.n; i++) |
|  | printf("%d,", s.a[i]); |
|  | } |

程序运行后的输出结果是

〖A〗2,3,1,6,8,7,5,4,10,9,

〖B〗3,4,2,7,9,8,6,5,11,10,

〖C〗2,4,3,9,12,12,11,11,18,9,

〖D〗1,2,3,6,8,7,5,4,10,9,

题目中定义了一个结构体类型S, 然后定义了一个结构体变量s并初始化, 执行f函数后, 将s中元素a的每个元素都加上i, i的值从0 ~ 9, 这里需要注意, 最后一个元素没有加i, 所以最终答案为C选项。

【37】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | #include <string.h> |
|  | typedef struct { |
|  | char name[9]; |
|  | char sex; |
|  | float score[2]; |
|  | } STU; |
|  | void f(STU a) { |
|  | STU b = { "Zhao", 'm', 85.0, 90.0 }; |
|  | int i; |
|  | strcpy(a.name, b.name); |
|  | a.sex = b.sex; |
|  | for (i = 0; i < 2; i++) |
|  | a.score[i] = b.score[i]; |
|  | } |
|  | void main() { |
|  | STU c = { "Qian", 'f', 95.0, 92.0 }; |
|  | f(c); |
|  | printf("%s,%c,%2.0f,%2.0f\n", c.name, c.sex, c.score[0], c.score[1]); |
|  | } |

程序的运行结果是

〖A〗Zhao, m, 85, 90

〖B〗Qian, m, 85, 90

〖C〗Zhao, f, 95, 92

〖D〗Qian, f, 95, 92

本题考查结构体的相关操作以及传值、传址的区别, 该题中调用f函数后, 会生成参数c的一个副本, 而不会改变c的值, 所以c值维持原值, 选项D正确。

【38】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | #define f(x) x \* x \* x |
|  | void main() { |
|  | int a = 3, s, t; |
|  | s = f(a + 1); |
|  | t = f((a + 1)); |
|  | printf("%d,%d\n", s, t); |
|  | } |

程序运行后的输出结果是

〖A〗10,64

〖B〗10,10

〖C〗64,10

〖D〗64,64

本题考查宏定义的用法, 宏定义只是做个简单的替换, 所以本题中执行f(a + 1) = a + 1 \* a + 1 \* a + 1 = 3 \* a + 1 = 10, f((a + 1)) = (a + 1) \* (a + 1) \* (a + 1) = 64, 所以答案为A选项。

【39】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | char a = 4; |
|  | printf("%d\n", a = a << 1); |
|  | } |

程序的运行结果是

〖A〗16

〖B〗8

〖C〗40

〖D〗4

本题考查位运算, 题目中将4向左移一位然后重新赋值给a, 4左移一位为8, 程序运行结果为8, 选项B正确。

【40】设文件指针fp已定义, 执行语句fp = fopen("file", "w"); 后, 以下针对文本文件file操作叙述的选项中正确的是

〖A〗只能写不能读

〖B〗写操作结束后可以从头开始读

〖C〗可以在原有内容后追加写

〖D〗可以随意读和写

考查文件操作函数fopen的基础知识, 以"w"方式打开文件, 只能写不能读。

**编程题**

[编程题代码下载](https://github.com/CoderMJLee/NCRE)

【41】使用VC++2010打开考生文件夹下blank1中的解决方案。此解决方案的项目中包含一个源程序文件blank1.c。在此程序中, 函数fun的功能是：调用随机函数产生20个互不相同的整数放在形参a所指数组中(此数组在主函数中已置0)。  
请在程序的下画线处填入正确的内容并把下画线删除, 使程序得出正确的结果。  
注意：部分源程序在文件blank1.c中。  
不得增行或删行, 也不得更改程序的结构 !

复制代码

C

|  |  |
| --- | --- |
|  | (1) N或20 |
|  | (2) break |
|  | (3) n |

填空1:变量n用于存储数组的下标, 要通过while语句对数组进行赋值, 数组的容量为20, 因此循环条件应为n < 20。  
填空2:通过一个for循环判断x是否与数组中已存的元素重复, 若重复则跳出for循环结束。  
填空3:若for循环是由break语句结束的, 则x与数组中的元素重复, 此时i必然小于n; 若for循环是因为循环变量i递增到某值, 而不再满足循环条件结束的, 说明x的值与数组中的元素不重复, 则此时i的值等于n。

【42】使用VC++2010打开考生文件夹下modi1中的解决方案。此解决方案的项目中包含一个源程序文件modi1.c。在此程序中, 函数fun的功能是：先从键盘上输入一个3行、3列的矩阵的各个元素的值, 然后输出主对角线元素之和。  
请改正程序中的错误, 使它能得出正确的结果。  
注意：部分源程序在文件modi1.c中。  
不要改动main函数, 不得增行或删行, 也不得更改程序的结构 !

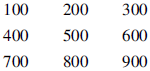
复制代码

C

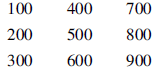
|  |  |
| --- | --- |
|  | (1) sum = 0; |
|  | (2) scanf("%d", &a[i][j]); |

该题考查对循环语句的掌握和对数组概念的理解。  
本题的解题思路为：先从键盘输入一个3×3矩阵, 然后循环累加, 执行循环语句中的sum = sum + a[i][i]; 。  
因为变量sum用来存放累加后的结果, 所以应对其初始化为0。  
第二处错误考查标准输入函数scanf的格式, 被赋值的变量前要加上取地址符"&"。

【43】使用VC++2010打开考生文件夹下prog1中的解决方案。此解决方案的项目中包含一个源程序文件prog1.c。在此程序中, 编写程序, 实现矩阵(3行、3列)的转置(即行列互换)。  
例如, 若输入下面的矩阵：



则程序输出：



注意：部分源程序在文件prog1.c中。  
请勿改动主函数main和其他函数中的任何内容, 仅在函数fun的花括号中填入你编写的若干语句。

复制代码

C

|  |  |
| --- | --- |
|  | void fun(int array[3][3]) { |
|  | int i, j, t; |
|  | for (i = 0; i < 3; i++) |
|  | /\*将右上三角和左下三角对换, 实现行列互换\*/ |
|  | for (j = i + 1; j < 3; j++) { |
|  | t = array[i][j]; |
|  | array[i][j] = array[j][i]; |
|  | array[j][i] = t; |
|  | } |
|  | } |

要实现矩阵转置, 即将右上角数组元素和左下角数组元素对换, 本题通过数组元素交换方法, 完成矩阵转置操作。  
因为对矩阵转置后仍然存回其本身, 所以只能循环矩阵中的一个角(本程序是右上半三角)。  
控制右上半三角的方法是在内层循环中循环变量j从i + 1或i开始。