[**【计算机二级C语言】卷013**](https://www.cnblogs.com/mjios/p/12452922.html)

**选择题**

**公共知识**

【1】下列叙述中正确的是

〖A〗程序执行的效率与数据的存储结构密切相关

〖B〗程序执行的效率只取决于程序的控制结构

〖C〗程序执行的效率只取决于所处理的数据量

〖D〗以上说法均错误

程序执行的效率与数据的存储结构、数据的逻辑结构、程序的控制结构、所处理的数据量等有关。

【2】下列与队列结构有关联的是

〖A〗函数的递归调用

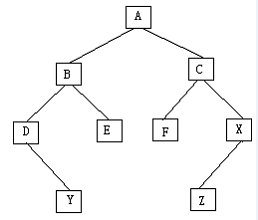
〖B〗数组元素的引用

〖C〗多重循环的执行

〖D〗先到先服务的作业调度

队列的修改是依先进先出的原则进行的, D正确。

【3】对下列二叉树



进行前序遍历的结果为

〖A〗DYBEAFCZX

〖B〗YDEBFZXCA

〖C〗ABDYECFXZ

〖D〗ABCDEFXYZ

前序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中, 首先访问根结点, 然后遍历左子树, 最后遍历右子树; 并且, 在遍历左右子树时, 仍然先访问根结点, 然后遍历左子树, 最后遍历右子树。  
前序遍历描述为：若二叉树为空, 则执行空操作。  
否则：①访问根结点; ②前序遍历左子树; ③前序遍历右子树, C正确。

【4】一个栈的初始状态为空。现将元素1, 2, 3, A, B, C依次入栈, 然后再依次出栈, 则元素出栈的顺序是

〖A〗1, 2, 3, A, B, C

〖B〗C, B, A, 1, 2, 3

〖C〗C, B, A, 3, 2, 1

〖D〗1, 2, 3, C, B, A

栈的修改是按后进先出的原则进行的, 所以顺序应与入栈顺序相反, 故选C。

【5】下面属于白盒测试方法的是

〖A〗等价类划分法

〖B〗逻辑覆盖

〖C〗边界值分析法

〖D〗错误推测法

白盒测试法主要有逻辑覆盖、基本路径测试等。  
逻辑覆盖测试包括语句覆盖、路径覆盖、判定覆盖、条件覆盖、判断-条件覆盖, 选择B。  
其余三项为黑盒测试法。

【6】下面对对象概念描述正确的是

〖A〗对象间的通信靠消息传递

〖B〗对象是名字和方法的封装体

〖C〗任何对象必须有继承性

〖D〗对象的多态性是指一个对象有多个操作

对象之间进行通信的构造叫做消息, A正确。  
多态性是指同一个操作可以是不同对象的行为, D错误。  
对象不一定必须有继承性, C错误。  
封装性是指从外面看只能看到对象的外部特征, 而不知道也无须知道数据的具体结构以及实现操作, B错误。

【7】软件设计中模块划分应遵循的准则是

〖A〗低内聚低耦合

〖B〗高耦合高内聚

〖C〗高内聚低耦合

〖D〗以上说法均错误

根据软件设计原理提出如下优化准则：①划分模块时, 尽量做到高内聚、低耦合, 保持模块相对独立性, 并以此原则优化初始的软件结构。

②一个模块的作用范围应在其控制范围之内, 且判定所在的模块应与受其影响的模块在层次上尽量靠近。

③软件结构的深度、宽度、扇入、扇出应适当。

④模块的大小要适中。

故C项正确。

【8】下面描述中不属于数据库系统特点的是

〖A〗数据共享

〖B〗数据完整性

〖C〗数据冗余度高

〖D〗数据独立性高

数据库系统的特点为高共享、低冗余、独立性高、具有完整性等, C错误。

【9】数据库设计过程不包括

〖A〗概念设计

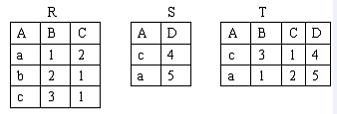
〖B〗逻辑设计

〖C〗物理设计

〖D〗算法设计

数据库设计过程主要包括需求分析、、概念结构设计、逻辑结构分析、数据库物理设计、数据库实施、数据库运行和维护阶段。  
答案为D选项。

【10】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〗投影

〖D〗并

关系R和关系S有公共域, 关系T是通过公共域的等值进行连接的结果, 符合自然连接, 选A。

**专业知识**

【11】下列叙述中正确的是

〖A〗C程序中所有函数之间都可以相互调用

〖B〗在C程序中main函数的位置是固定的

〖C〗在C程序的函数中不能定义另一个函数

〖D〗每个C程序文件中都必须要有一个main函数

C程序中主函数不能被其他函数调用, 所以A选项错误。  
main函数可以放在程序开始, 也可以放在中间, 也可以放在最后, 位置不固定, 但程序执行时必须从main函数开始, 所以B选项错误。  
在C程序的函数中不能定义另一个函数, 可以声明或调用另一个函数, 所以C选项正确。  
每个C程序中必须包含一个main函数, 但不一定是每个C程序文件中必须有, 用户单独编写的某个函数也可以存储为一个C程序文件, 所以D选项错误。

【12】以下叙述正确的是

〖A〗C语言函数可以嵌套调用, 例如：fun(fun(x))

〖B〗C语言程序是由过程和函数组成的

〖C〗C语言函数不可以单独编译

〖D〗C语言中除了main函数, 其他函数不可作为单独文件形式存在

C语言程序是由函数组成的, 所以B错误。  
C语言函数可以单独进行编译, 所以C错误。  
每个C程序中必须包含一个main函数, 但不一定是每个C程序文件中必须有, 用户单独编写的某个函数也可以存储为一个C程序文件, 所以D错误。

【13】以下选项中合法的标识符是

〖A〗1\_ \_

〖B〗1-1

〖C〗1\_1

〖D〗\_11

C语言中的标识符由下划线、字母和数字组成, 且必须以下划线或字母开始, 所以只有D选项正确。

【14】表达式3.6 - 5 / 2 + 1.2 + 5 % 2的值是

〖A〗4.3

〖B〗4.8

〖C〗3.3

〖D〗3.8

计算5 / 2, 结果取整数值2。  
计算3.6 - 2, 结果与高精度数据保持一致, 即为1.6。  
计算1.6 + 1.2, 结果为2.8。  
计算5 % 2, 结果为1。  
计算2.8 + 1结果为3.8。  
所以选择D选项。

【15】有以下定义：

复制代码

C

|  |  |
| --- | --- |
|  | int a; |
|  | long b; |
|  | double x, y; |

则以下选项中正确的表达式是

〖A〗a = x<>y

〖B〗a % (int) (x - y)

〖C〗(a \* y) % b

〖D〗y = x + y = x

C语言中没有"< >"运算符, 所以A选项错误。  
"%"运算符的两个操作值必须为整型数据, 所以B选项正确。  
a \* y的结果为double型, 所以C选项错误。  
不能将值赋给像x + y这样的表达式, 所以D选项错误。

【16】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int a = 0, b = 0; |
|  | /\*给a赋值 a = 10; |
|  | b = 20; |
|  | 给b赋值\*/ |
|  | printf("a+b=%d\n", a + b); |
|  | /\*输出计算结果\*/ |
|  | } |

程序运行后的输出结果是

〖A〗出错

〖B〗a+b=30

〖C〗a+b=10

〖D〗a+b=0

注释语句可以在一行也可以在多行, 用/\*……\*/ 或//表示, 。  
它不参与程序的运行, 所以a与b的值仍然为0, 所以打印的结果为a + b = 0。

【17】以下选项中, 值为1的表达式是

〖A〗1 - '0'

〖B〗1 - '\0'

〖C〗'1' - 0

〖D〗'\0' - '0'

\0, 空值, 其ASCII码值为0, 所以B选项的值为1。  
字符0的ASCII码值为048, 所以A、C、D的值均不为1。

【18】当变量c的值不为2、4、6时, 值为"真"的表达式是

〖A〗(c >= 2 && c <= 6) && (c % 2 != 1)

〖B〗(c == 2) || (c == 4) || (c == 6)

〖C〗(c >= 2 && c <= 6) && !(c % 2)

〖D〗(c >= 2 && c <= 6) || (c != 3) || (c != 5)

逻辑或表达式中有一个值不为0, 结果即为真, 逻辑与表达式中只有两个运算对象均非零, 结果才为真。  
所以只有选项D满足要求。

【19】下列条件语句中, 输出结果与其他语句不同的是

〖A〗if ( a!=0 ) printf("%d\n", x); else printf("%d\n", y);

〖B〗if ( a==0 ) printf("%d\n", y); else printf("%d\n", x);

〖C〗if ( a==0 ) printf("%d\n", x); else printf("%d\n", y);

〖D〗if ( a ) printf("%d\n", x); else printf("%d\n", y);

A, B, D选项的含义均为：a的值如果为0时打印y的值, 否则打印x的值, 但是C选项的含义是：a的值为0时打印x的值, 不为0时打印y的值, 与其他选项正好相反。

【20】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int x = 1, y = 2, z = 3; |
|  | if (x > y) |
|  | if (y < z) |
|  | printf("%d", ++z); |
|  | else |
|  | printf("%d", ++y); |
|  | printf("%d\n", x++); |
|  | } |

程序的运行结果是

〖A〗1

〖B〗41

〖C〗2

〖D〗331

else 总是与离他最近的上面的if配对, 所以第一个if语句的判断条件不成立, 不再执行第二个if语句和与第二个if语句配对的else语句, 而是直接执行打印x++的操作, 打印x的值1以后再加1。

【21】要求通过while循环不断读入字符, 当读入字母N时结束循环。若变量已正确定义, 以下正确的程序段是

〖A〗while (ch = getchar() = 'N') printf("%c ", ch);

〖B〗while ((ch = getchar()) != 'N') printf("%c ", ch);

〖C〗while (ch = getchar() == 'N') printf("%c ", ch);

〖D〗while ((ch = getchar()) == 'N') printf("%c ", ch);

在C语言中等于号用"= ="表示, 一个"="表示赋值, 所以A错误。  
C、D 选项的含义都是如果输入的字符为 "N", 则打印输出, 所以正好和题意相反。

【22】以下叙述中正确的是

〖A〗continue语句的作用是：使程序的执行流程跳出包含它的所有循环

〖B〗break语句只能用在循环体内和switch语句体内

〖C〗在循环体内使用break语句和continue语句的作用相同

〖D〗break语句只能用于switch语句体中

continue语句的作用是：跳过循环体中剩余的语句而进行下一次循环, 所以A选项错误。  
Break语句的作用是终止正在执行的switch流程, 跳出switch结构或者强制终止当前循环, 从当前执行的循环中跳出, 所以C、D选项错误。

【23】若各选项中所用变量已正确定义, 函数fun中通过return 语句返回一个函数值, 以下选项中错误的程序是

〖A〗void main() { …… x = fun(2, 10); …… } float fun(int a, intB) { …… }

〖B〗float fun(int a, intB) { …… } void main() { …… x = fun(i, j); …… }

〖C〗float fun(int, int); void main() { …… x = fun(2, 10); …… }

〖D〗void main() { float fun(int i, int j); …… x = fun(i, j); …… }

在调用子函数时, 应对其进行说明。  
A选项中, 调用时, 没有对子函数进行说明。  
B选项、C选项中被调用函数在主调函数之前定义, 不用说明; D选项中在主函数中对被调用函数的返回值类型进行了说明。  
因此本题选择A选项。

【24】若有定义语句：

复制代码

C

int year = 2009, \*p = &year;

以下不能使变量year中的值增至2010的语句是

〖A〗(\*p)++;

〖B〗\*p++;

〖C〗++(\*p);

〖D〗\*p += 1;

由于自加运算符"++"的运算级别运算符"\*"的运算级别相同, 按照自右向左的方向结合, B选项的表达式\*p++, 因为p++是后置自加, 所以需要先取值再自加, 所以表达式\*p++相当于\*p; p++;, 即先取出p指针指向变量的值2009, 再将指针p指向下一个存储空间, 所以不能是变量year的值增加至2010。  
因此本题答案为B选项。

【25】若在定义语句：  
int a, b, c, \*p = &c;  
接着执行以下选项中的语句, 则能正确执行的语句是

〖A〗scanf("%d", a, b, c);

〖B〗scanf("%d%d%d", a, b, c);

〖C〗scanf("%d", p);

〖D〗scanf("%d", &p);

本题重点考查的知识点是标准输入函数scanf()。  
scanf()函数要求, 除了第1个参数为格式化字符串以外, 其余参数均为相应变量的地址值。  
本题中, 只有p是地址值, 因此C选项正确。

【26】以下错误的定义语句是

〖A〗int x[4][3] = { { 1, 2, 3 }, { 1, 2, 3 }, { 1, 2, 3 }, { 1, 2, 3 } };

〖B〗int x[4][] = { { 1, 2, 3 }, { 1, 2, 3 }, { 1, 2, 3 }, { 1, 2, 3 } };

〖C〗int x[][3] = { { 0 }, { 1 }, { 1, 2, 3 } };

〖D〗int x[][3] = { 1, 2, 3, 4 };

二维数组定义的一般形式是：类型说明符数组名[常量表达式1][常量表达式2]。

其中"常量表达式1"表示第一维下标的长度, "常量表达式2" 表示第二维下标的长度。

如对二维数组全部元素赋初值, 则第一维的长度可以不给出。  
但如果对x[4][]赋值, 只确定行数, 而不确定列数, 就无法正确赋值。  
因此B选项正确。

【27】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

该题中t[][3] = { 9, 8, 7, 6, 5, 4, 3, 2, 1 }; 实际上就是t[3][3] = { { 9, 8, 7 }, { 6, 5, 4 }, { 3, 2, 1 } };, 通过for循环语句, 确定需要输出t[2][0], t[1][1]和t[0][2], 即输出3、5、7。  
因此A选项正确。

【28】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void fun(int \*a, int n) /\*fun函数的功能是将a所指数组元素从大到小排序\*/ { |
|  | int t, i, j; |
|  | for (i = 0; i < n - 1; i++) |
|  | for (j = i + 1; j < n; j++) |
|  | if (a[i] < a[j]) { |
|  | t = a[i]; |
|  | a[i] = a[j]; |
|  | a[j] = t; |
|  | } |
|  | } |
|  | void main() { |
|  | int c[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 }, i; |
|  | fun(c + 4, 6); |
|  | for (i = 0; i < 10; i++) |
|  | printf("%d,", c[i]); |
|  | printf("\n"); |
|  | } |

程序的运行结果是

〖A〗1, 2, 3, 4, 9, 8, 7, 6, 5, 0,

〖B〗0, 9, 8, 7, 6, 5, 1, 2, 3, 4,

〖C〗0, 9, 8, 7, 6, 5, 4, 3, 2, 1,

〖D〗1, 2, 3, 4, 5, 6, 7, 8, 9, 0,

该题目中fun函数的功能是将a所指数组元素从大到小排序, fun(c + 4, 6); 即指排序从第5个元素开始进行从大到小排序。  
因此A选项正确。

【29】以下正确的字符串常量是

〖A〗"\\\"

〖B〗'abc'

〖C〗Olympic Games

〖D〗""

本题考查的知识点是：字符串常量。  
在C语言中, 字符串常量是以双引号括起来的字符序列。  
因此B选项和C选项不正确。  
字符序列中可包含一些转义字符, 转义字符都是以"\"开头的。  
A选项中包含了三个"\", 前两个(\\)代表了一个"\"字符,后面一个和"""一起(\")被看作一个""",所以该字符串缺少一个结束的""", 因此不正确。  
D选项的两个"""之间没有任何字符, 代表的是一个空串, 是合法的字符串常量, 因此D选项正确。

【30】下列选项中, 能够满足"只要字符串s1等于字符串s2,则执行ST"要求的是

〖A〗if (s1 - s2 == 0) ST;

〖B〗if (s1 == s2) ST;

〖C〗if (strcpy(s1, s2) == 1) ST;

〖D〗if (strcmp(s2, s1) == 0) ST;

在C语言中要对两个字符串的大小进行比较, 就需要调用字符串比较函数strcmp, 如果这个函数的返回值等于0, 说明两个字符串相等。  
因此D选项正确。

【31】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | int fun(char p[][10]) { |
|  | int n = 0, i; |
|  | for (i = 0; i < 7; i++) |
|  | if (p[i][0] == 'T') |
|  | n++; |
|  | return n; |
|  | } |
|  | void main() { |
|  | char str[][10] = { "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun" }; |
|  | printf("%d\n", fun(str)); |
|  | } |

程序执行后的输出结果是

〖A〗1

〖B〗2

〖C〗3

〖D〗0

在main函数中, 定义了字符串数组str, 分别给每个字符串赋值为每个星期英文的前三个字母。  
fun函数, 统计每个字符串首字母为"T"的个数。  
p[i][0]是字符串的首字符, 相当于二维数组第一列的所有元素。  
一共有2个"T", 所以n = 2, 故本题答案为B选项。

【32】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void fun(char \*t, char \*s) { |
|  | while (\*t != 0) |
|  | t++; |
|  | while ((\*t++ = \*s++) != 0); |
|  | } |
|  | void main() { |
|  | ss[10] = "acc", aa[10] = "bbxxyy"; |
|  | fun(ss, aa); |
|  | printf("%s,%s\n", ss, aa); |
|  | } |

程序的运行结果是

〖A〗acc, bbxxyy

〖B〗accbbxxyy, bbxxyy

〖C〗accxxyy, bbxxyy

〖D〗accxyy, bbxxyy

本题重点考察while循环语句的简单应用。  
函数fun()中, 前一个while循环的作用是：如果形参指针t所指内容不为0, 则让t增1, 直到它指向0。  
后一个 while循环的作用是：将s所指内容赋给t所指地址, 然后两者同时增1, 直到赋给t的内容为0。  
由此可见, 函数fun()的作用就是将形参s所指字符串连接到形参t所指字符串末尾, 相当于库函数中的strcat()。  
主函数中使用fun()将数组aa中的字符串连接到了数组ss中原有字符串之后, 所以执行完后, ss中的字符串为"accbbxxyy"。  
程序最后输出的结果是：accbbxxyy, bbxxyy。  
因此B选项正确。

【33】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | #include <string.h> |
|  | void fun(char \*w, int m) { |
|  | char s, \*p1, \*p2; |
|  | p1 = w; |
|  | p2 = w + m - 1; |
|  | while (p1 < p2) { |
|  | s = \*p1; |
|  | \*p1 = \*p2; |
|  | \*p2 = s; |
|  | p1++; |
|  | p2--; |
|  | } |
|  | } |
|  | void main() { |
|  | char a[] = "123456"; |
|  | fun(a, strlen(a)); |
|  | puts(a); |
|  | } |

程序运行后的输出结果是

〖A〗123456

〖B〗116611

〖C〗161616

〖D〗654321

主函数中调用fun(a, 6)后, 指针p1指向字符串中的"1"、p2指向字符串中的"6"。  
While循环中, 只要p1 < p2, 则把p1、p2所指向的字符互换, 同时p1前移, p2后移。  
最终字符串逆序存放。  
因此D选项正确。

【34】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | int b = 2; |
|  | int fun(int \*k) { |
|  | b = \*k + b; |
|  | return (b); |
|  | } |
|  | void main() { |
|  | int a[10] = { 1, 2, 3, 4, 5, 6, 7, 8 }, i; |
|  | for (i = 2; i < 4; i++) { |
|  | b = fun(&a[i]) + b; |
|  | printf("%d ", b); |
|  | } |
|  | printf("\n"); |
|  | } |

程序运行后的输出结果是

〖A〗8 10

〖B〗10 28

〖C〗10 12

〖D〗10 16

主函数中循环执行2次, fun函数被调用2次?b是全局变量, 整个程序均可使用?第1次调用：fun(&a[2]), 形参指针k获得a[2]的地址, 则\*k为a[2]的值?返回值：b = a[2] + 2 = 3 + 2 = 5; 主程序中：b = fun(&a[2]) + b = 5 + 5 = 10; 第2次调用：fun(&a[3]), 返回值：b = 4 + 10 = 14; 主程序中：b = fun(&a[3]) + b = 14 + 14 = 28。  
因此B选项正确?

【35】设有定义：

复制代码

C

|  |  |
| --- | --- |
|  | struct { |
|  | char mark[12]; |
|  | int num1; |
|  | double num2; |
|  | } t1, t2; |

若变量均已正确赋初值, 则以下语句中错误的是

〖A〗t1 = t2;

〖B〗t2.num1 = t1.num1;

〖C〗t2.mark = t1.mark;

〖D〗t2.num2 = t1.num2;

本题考查结构体变量的引用, C选项中mark为结构体中的数组, 不能直接赋值, 所以C选项错误。

【36】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | #include <string.h> |
|  | struct A { |
|  | int a; |
|  | char b[10]; |
|  | double c; |
|  | }; |
|  | void f(struct A t); |
|  | void main() { |
|  | struct A a = { 1001, "ZhangDa", 1098.0 }; |
|  | f(a); |
|  | printf("%d,%s,%6.1f\n", a.a, a.b, a.c); |
|  | } |
|  | void f(struct A t) { |
|  | t.a = 1002; |
|  | strcpy(t.b, "ChangRong"); |
|  | t.c = 1202.0; |
|  | } |

程序运行后的输出结果是

〖A〗1002,ZhangDa,1202.0

〖B〗1002,ChangRong,1202.0

〖C〗1001,ChangRong,1098.0

〖D〗1001,ZhangDa,1098.0

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a的一个副本, 而不会改变a的值, 所以a值维持原值, 选项D正确。

【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; |
|  | STU 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]; |
|  | return a; |
|  | } |
|  | void main() { |
|  | STU c = { "Qian", 'f', 95.0, 92.0 }, d; |
|  | d = f(c); |
|  | printf("%s,%c,%2.0f,%2.0f\n", d.name, d.sex, d.score[0], d.score[1]); |
|  | } |

程序的运行结果是

〖A〗Zhao, m, 85, 90

〖B〗Qian, m, 85, 90

〖C〗Qian, f, 95, 92

〖D〗Zhao, 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, 会返回记过计算的a值, 选项A正确。

【38】若程序中有宏定义行：  
#define N 100  
则以下叙述中正确的是

〖A〗宏定义行中定义了标识符N的值为整数100

〖B〗在编译程序对C源程序进行预处理时用100替换标识符N

〖C〗上述宏定义行实现将100赋给标示符N

〖D〗在运行时用100替换标识符N

本题考查预编译相关知识, 宏定义在编译程序时做了一个简单的替换, 所以选项B正确。

【39】若变量已正确定义, 则以下语句的输出结果是

复制代码

C

|  |  |
| --- | --- |
|  | s = 32; |
|  | s ^= 32; |
|  | printf("%d", s); |

〖A〗-1

〖B〗0

〖C〗1

〖D〗32

本题考查逻辑异或运算, 异或运算只有在两个比较的位不同时其结果是1, 否则结果为0, 题目中两个值相同, 所以结果为0。

【40】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | FILE \*fp; |
|  | char str[10]; |
|  | fp = fopen("myfile.dat", "w"); |
|  | fputs("abc", fp); |
|  | fclose(fp); |
|  | fp = fopen("myfile.dat", "a+"); |
|  | fprintf(fp, "%d", 28); |
|  | rewind(fp); |
|  | fscanf(fp, "%s", str); |
|  | puts(str); |
|  | fclose(fp); |
|  | } |

程序运行后的输出结果是

〖A〗abc

〖B〗28c

〖C〗abc28

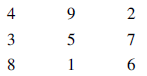
〖D〗因类型不一致而出错

本题考查文件操作函数fprintf()函数, fprintf()函数向文件输出, 将输出的内容输出到硬盘上的文件或是相当于文件的设备上执行两次fprintf后文件中有abc28, 所以C选项正确。

**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：判定形参a所指的N×N(规定N为奇数)的矩阵是否是"幻方", 若是, 函数返回值为1; 若不是, 函数返回值为0。"幻方"的判定条件是：矩阵每行、每列、主对角线及反对角线上元素之和都相等。  
例如, 以下3×3的矩阵就是一个"幻方":



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) 0 |
|  | (2) || |
|  | (3) 1 |

填空1:本题考查为变量赋初值, 在这里row += a[i][j]代表是每行的总和, colum += a[j][i]代表的是每列的总和, 因而row, colum在初始化时应该为零, 此空应该填0。  
填空2:本题考查了if条件语句, 此句if判断代表每行的总和与列是否相等, 每行的总和是否与对角线的总和相等, 两者若有一个不成立, 即返回0, 因而此空应该填写||。  
填空3:题目要求若矩阵是"幻方", 则函数返回值为1, 因而此空应该填写1。

【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：传入一个整数m, 计算如下公式的值。



例如, 若输入5, 则应输出-0.283333。  
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) t -= 1.0 / i; |
|  | (2) return t; |

(1) 变量t存放公式的和, 通过循环语句进行复合运算, 因此此处应改为t -= 1.0 / i;, 注意此处应进行变量的类型转换。  
(2) 循环结束后应将和值返回给主函数。

【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, 编写一个函数, 用来删除字符串中的所有空格。  
例如, 输入asd af aa z67, 则输出为asdafaaz67。  
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(char \*str) { |
|  | int i = 0; |
|  | char \*p = str; |
|  | while (\*p) { |
|  | if (\*p != ' ') |
|  | /\*删除空格\*/ { |
|  | str[i++] = \*p; |
|  | } |
|  | p++; |
|  | } |
|  | str[i] = '\0'; |
|  | /\*加上结束符\*/ |
|  | } |

本题要求删除所有空格, 即保留除了空格以外的其他所有字符。  
由于C语言中没有直接删除字符的操作, 所以对不需要删除的字符采用"保留"的操作。  
用指针p指向字符串中的每一个字符, 每指向到一个字符都判断其是否为空格, 若不是空格则保存到str[i]。