**[【计算机二级C语言】卷011](https://www.cnblogs.com/mjios/p/12452888.html)**

**选择题**

**公共知识**

【1】下列关于栈叙述正确的是()。

〖A〗栈顶元素最先能被删除

〖B〗栈顶元素最后才能被删除

〖C〗栈底元素永远不能被删除

〖D〗栈底元素最先被删除

栈是先进后出的数据结构, 所以栈顶元素最后入栈却最先被删除。  
栈底元素最先入栈却最后被删除。  
所以选择A)。

【2】下列叙述中正确的是()。

〖A〗在栈中, 栈中元素随栈底指针与栈顶指针的变化而动态变化

〖B〗在栈中, 栈顶指针不变, 栈中元素随栈底指针的变化而动态变化

〖C〗在栈中, 栈底指针不变, 栈中元素随栈顶指针的变化而动态变化

〖D〗以上说法均不正确

栈是先进后出的数据结构, 在整个过程中, 栈底指针不变, 入栈与出栈操作均由栈顶指针的变化来操作, 所以选择C)。

【3】某二叉树共有7个结点, 其中叶子结点只有1个, 则该二叉树的深度为(假设根结点在第1层)()。

〖A〗3

〖B〗4

〖C〗6

〖D〗7

根据二叉树的基本性质3:在任意一棵二叉树中, 多为0的叶子结点总比度为2的结点多一个, 所以本题中度为2的结点为1 - 1 = 0个, 所以可以知道本题目中的二叉树的每一个结点都有一个分支, 所以共7个结点共7层, 即度为7。

【4】软件按功能可以分为应用软件、系统软件和支撑软件(或工具软件)。下面属于应用软件的是()。

〖A〗学生成绩管理系统

〖B〗C语言编译程序

〖C〗UNIX 操作系统

〖D〗数据库管理系统

软件按功能可以分为：应用软件、系统软件、支撑软件。  
操作系统、编译程序、汇编程序、网络软件、数据库管理系统都属于系统软件。  
所以B)、C)、D)都是系统软件, 只有A)是应用软件。

【5】结构化程序所要求的基本结构不包括()。

〖A〗顺序结构

〖B〗GOTO跳转

〖C〗选择(分支)结构

〖D〗重复(循环)结构

1966年Boehm和Jacopini证明了程序设计语言仅仅使用顺序、选择和重复三种基本控制结构就足以表达出各种其他形式结构的程序设计方法。

【6】下面描述中错误的是()。

〖A〗系统总体结构图支持软件系统的详细设计

〖B〗软件设计是将软件需求转换为软件表示的过程

〖C〗数据结构与数据库设计是软件设计的任务之一

〖D〗PAD图是软件详细设计的表示工具

详细设计的任务是为软件结构图中而非总体结构图中的每一个模块确定实现算法和局部数据结构, 用某种选定的表达工具表示算法和数据结构的细节, 所以A)错误。

【7】负责数据库中查询操作的数据库语言是()。

〖A〗数据定义语言

〖B〗数据管理语言

〖C〗数据操纵语言

〖D〗数据控制语言

数据定义语言：负责数据的模式定义与数据的物理存取构建; 数据操纵语言：负责数据的操纵, 包括查询及增、删、改等操作; 数据控制语言：负责数据完整性、安全性的定义与检查以及并发控制、故障恢复等功能。

【8】一个教师可讲授多门课程, 一门课程可由多个教师讲授。则实体教师和课程间的联系是()。

〖A〗1 : 1联系

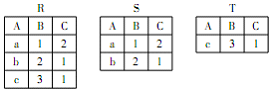
〖B〗1 : m联系

〖C〗m : 1联系

〖D〗m : n联系

因为一个教师可讲授多门课程, 而一门课程又能由多个老师讲授所以他们之间是多对多的关系, 可以表示为m : n。

【9】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〗并

〖C〗交

〖D〗差

关系T中的元组是关系R中有而关系S中没有的元组的集合, 即从关系R中除去与关系S中相同元组后得到的关系T。  
所以做的是差运算。

【10】定义无符号整数类为UInt, 下面可以作为类UInt实例化值的是()。

〖A〗-369

〖B〗369

〖C〗0.369

〖D〗整数集合 { 1, 2, 3, 4, 5 }

只有B)选项369可以用无符号整数来表示和存储。  
A)选项-369有负号, 选项C)0.369是小数都不能用无符号整数类存储。  
选项D)是一个整数集合得用数组来存储。

**专业知识**

【11】以下叙述中错误的是

〖A〗C程序在运行过程中所有计算都以二进制方式进行

〖B〗C程序在运行过程中所有计算都以十进制方式进行

〖C〗所有C程序都需要编译链接无误后才能运行

〖D〗C程序中字符变量存放的是字符的ASCII值

C程序在运行过程中的所有计算都以二进制方式进行, 所以B选项错误。

【12】以下关于C语言的叙述中正确的是

〖A〗C语言的数值常量中夹带空格不影响常量值的正确表示

〖B〗C语言中的变量可以在使用之前的任何位置进行定义

〖C〗在C语言算术表达式的书写中, 运算符两侧的运算数类型必须一致

〖D〗C语言中的注释不可以夹在变量名或关键字的中间

C语言的数值常量中不能夹带空格, 所以A错误。  
C语言中的变量都必须先定义再使用, 对变量的定义通常放在函数体内的前部, 但也可以放在函数外部或复合语句的开头, 不能随便放置, 所以B错误。  
在C语言中运算符两侧的运算数据类型可以不一致, 且结果与精度较高的保持一致, 所以C错误。

【13】以下不合法的字符常量是

〖A〗'\\'

〖B〗'\"'

〖C〗'\018'

〖D〗'\xcc'

转义字符中, 八进制的表示形式为\ddd, 但是八进制中不能包含数字8, 所以C选项不合法。

【14】以下选项中正确的定义语句是

〖A〗double, a, b;

〖B〗double a = b = 7;

〖C〗double a; b;

〖D〗double a = 7, b = 7;

A选项中变量类型符double后面不能有逗号。  
B选项中变量b不能先使用再定义, 应该写成选项D中的格式。  
C选项中变量a后面应该是逗号而不是分号。

【15】若有定义语句：

复制代码

C

int a = 3, b = 2, c = 1;

以下选项中错误的赋值表达式是

〖A〗a = (b = 4) = 3;

〖B〗a = b = c + 1;

〖C〗a = (b = 4) + c;

〖D〗a = 1 + (b = c = 4);

不能将变量或常量赋给一个表达式, 所以A选项中(b = 4) = 3是错误的。

【16】若有定义：

复制代码

C

int a, b;

通过语句  
scanf("%d;%d", &a, &b);  
能把整数3赋给变量a, 5赋给变量b的输入数据是

〖A〗3 5

〖B〗3, 5

〖C〗3; 5

〖D〗35

输入数据的格式必须与scanf()的格式控制串完全匹配, 所以输入时必须在3和5之间输入";"。

【17】已知大写字母A的ASCII码是65, 小写字母a的ASCII码是97。以下不能将变量c中的大写字母转换为对应小写字母的语句是

〖A〗c = ('A' + c) % 26 - 'a'

〖B〗c = c + 32

〖C〗c = c - 'A' + 'a'

〖D〗c = (c - 'A') % 26 + 'a'

根据题意可知, 小写字母比与之对应的大写字母的ASCII码大32。  
A选项中字符A加上c表示的大写字符再对字母个数26取余, 本身这个表达式没有任何含义所以选择A选项。

【18】在以下给出的表达式中, 与while(E)中的(E)不等价的表达式是

〖A〗(E > 0 || E < 0)

〖B〗(E == 0)

〖C〗( !E == 0)

〖D〗(E != 0)

while语句中条件表达式E的值不为0即为真, 认为满足条件, 所以与B选项中表达式的含义正好相反, 选择B选项。

【19】以下程序段中, 与语句：

复制代码

C

k = a > b ? (b > c ? 1 : 0) : 0;

功能相同的是

〖A〗if ((a > b) || (b > c)) k = 1; else k = 0;

〖B〗if ((a > b) && (b > c)) k = 1; else k = 0;

〖C〗if (a <= b) k = 0; else if (b <= c) k = 1;

〖D〗if (a > b) k = 1; else if (b > c) k = 1; else k = 0;

条件表达式的含义是, 如果表达式1成立, 结果为表达式2的值, 如果不成立为表达式3的值。  
所以题干中k的值这样取值：如果a > b且b > c, k值结果为1, 如果a > b, 且b < c则值为0。  
如果a < b, k值为0。  
也就是只要a > b与b > c中只要有一个条件不成立, k的值就为0, 所以选择B选项。

【20】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int a = 1, b = 2; |
|  | for (; a < 8; a++) { |
|  | b += a; |
|  | a += 2; |
|  | } |
|  | printf("%d,%d\n", a, b); |
|  | } |

程序运行后的输出结果是

〖A〗9,18

〖B〗8,11

〖C〗7,11

〖D〗10,14

第一次循环a的值为1, 满足条件, 执行b += a, 与a += 2, 则b的值变为3, a的值变为3。  
执行a++, a的值为4, 满足条件进入第二次循环, 执行完循环体后b的值为7, a的值为6。  
执行a++, a的值为7, 满足条件进入第三次循环, 执行完循环体后b的值为14, a的值为9。  
执行a++, a的值变为10。  
所以选择D选项。

【21】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int i, j, m = 55; |
|  | for (i = 1; i <= 3; i++) |
|  | for (j = 3; j <= i; j++) |
|  | m = m % j; |
|  | printf("%d\n ", m); |
|  | } |

程序的运行结果是

〖A〗0

〖B〗1

〖C〗2

〖D〗3

第一次执行外循环i的值为1, 满足条件, 判断第一次内循环, j的值为3, 不满足条件, 跳出内循环。  
第二次执行外循环i的值为2, 同理也不满足内循环条件, 跳出内循环。  
第三次执行外循环i的值为3, 此时进入内循环判断条件时, 条件成立, 执行m = m % j, m的值为1, 跳出内循环, 跳出外循环打印m的值。

【22】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int x = 8; |
|  | for (; x > 0; x--) { |
|  | if (x % 3) { |
|  | printf("%d,", x--); |
|  | continue; |
|  | } |
|  | printf("%d,", --x); |
|  | } |
|  | } |

程序的运行结果是

〖A〗7, 4, 2,

〖B〗8, 7, 5, 2,

〖C〗9, 7, 6, 4,

〖D〗8, 5, 4, 2,

coutinue的作用是跳出循环体中剩余的语句而进行下一次循环。  
第一次循环x的值为8, 循环体中if条件成立, 打印x的值8后将x减1, 再执行continue语句, 跳出本次循环。  
第二次判断循环条件时, x的值变为6, 不满足循环体内if条件, 执行打印--x的操作, 即打印5后跳出循环。  
第三次判断循环条件时x的值为4, 满足循环体中if条件, 执行打印x--的操作, 即打印4, 后将x值减一, 执行continue语句, 跳出本次循环。  
第四次判断循环条件时x的为2, 满足循环体中if条件, 打印x--, 即打印2后将x减一, 执行continue语句, 跳出本次循环。  
在进行for条件表达式中第三个表达式x--的操作后x的值为0, 不满足条件结束循环。  
所以打印结果为8, 5, 4, 2, 。

【23】以下叙述中错误的是

〖A〗C程序必须由一个或一个以上的函数组成

〖B〗函数调用可以作为一个独立的语句存在

〖C〗若函数有返回值, 必须通过return 语句返回

〖D〗函数形参的值也可以传回给对应的实参

形参是函数定义时由用户定义的形式上的变量, 实参是函数调用时, 主调函数为被调函数提供的原始数据。  
在C语言中, 函数形参的值不可以传回给对应的实参。  
因此D选项正确。

【24】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int a = 1, b = 3, c = 5; |
|  | int \*p1 = &a, \*p2 = &b, \*p = &c; |
|  | \*p = \*p1 \* (\*p2); |
|  | printf("%d\n", c); |
|  | } |

执行后的输出结果是

〖A〗1

〖B〗2

〖C〗3

〖D〗4

该程序中int \*p1 = &a, \*p2 = &b, \*p = &c; 指定义三个指针变量, 并赋值, 即使p1指向a; p2指向b; p指向c。  
\*p = \*p1 \* (\*p2); 该条语句是给p所指的存储单元c赋值, 就是p1所指的存储单元的值, 即a的值, 与p2所指的存储单元b的值相乘, 也就是c = a \* b, 等价于c = 1 \*3 = 3; 因此C选项正确。

【25】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void f(int \*p, int \*q); |
|  | void main() { |
|  | int m = 1, n = 2, \*r = &m; |
|  | f(r, &n); |
|  | printf("%d,%d", m, n); |
|  | } |
|  | void f(int \*p, int \*q) { |
|  | p = p + 1; |
|  | \*q = \*q + 1; |
|  | } |

程序运行后的输出结果是

〖A〗2,3

〖B〗1,3

〖C〗1,4

〖D〗1,2

在f(int \* p, int \*q)函数中, 执行p = p + 1是将p所对应的地址加1, 而\*q = \*q + 1是将q所指向的n的地址所对应的值加1, 所以m的得知所对应的值没有变, 而n的值则为3了。  
因此B选项正确。

【26】若有定义语句：int a[2][3], \*p[3];  
则以下语句中正确的是

〖A〗p = a;

〖B〗p[0] = a;

〖C〗p[0] = &a[1][2];

〖D〗p[1] = &a;

A选项错误, 因为p是指向一个指针数组, 作为数组名, 不能指向别的地方。  
B选项错误, 因为p[0]是一个int指针, 也就是int\*; 而a是一个指向指针的指针int \* \*。  
C选项正确, 因为p[0]是一个int\*, a[1][2]是int, &a[1][2]是int\*, 类型吻合。  
D选项错误, 因为a作为数组名, 不能取地址。  
即使能取, p[1]是int\*, &a是int \* \*\*, 类型不对。  
因此C选项正确。

【27】以下程序中函数f的功能是：当flag为1时, 进行由小到大排序; 当flag为0时, 进行由大到小排序。

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void f(int b[], int n, int flag) { |
|  | int i, j, t; |
|  | for (i = 0; i < n - 1; i++) |
|  | for (j = i + 1; j < n; j++) |
|  | if (flag ? b[i] > b[j] : b[i] < b[j]) { |
|  | t = b[i]; |
|  | b[i] = b[j]; |
|  | b[j] = t; |
|  | } |
|  | } |
|  | void main() { |
|  | int a[10] = { 5, 4, 3, 2, 1, 6, 7, 8, 9, 10 }, i; |
|  | f(&a[2], 5, 0); |
|  | f(a, 5, 1); |
|  | for (i = 0; i < 10; i++) |
|  | printf("%d,", a[i]); |
|  | } |

程序运行后的输出结果是

〖A〗1,2,3,4,5,6,7,8,9,10,

〖B〗3,4,5,6,7,2,1,8,9,10,

〖C〗5,4,3,2,1,6,7,8,9,10,

〖D〗10,9,8,7,6,5,4,3,2,1,

本题重点考查函数的参数传递, 函数的参数传递分为传值和传地址两种情况。  
本题就是结合数组考查参数传递的情形。  
解析：函数f完成的功能是对数据进行排序, 语句f(&a[2], 5, 0)的作用是对从a[2]开始的5个元素进行从大到小排序。  
注意：这里传递的是地址&a[2], 所以排序操作可看作是直接对数组a操作, 执行后的结果为5, 4, 7, 6, 3, 2, 1, 8, 9, 10。  
语句f(a, 5, 1)对数组a的前5个元素从小到大排序, 排序后数组为：3, 4, 5, 6, 7, 2, 1, 8, 9, 10。  
因此B选项正确。

【28】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int s[12] = { 1, 2, 3, 4, 4, 3, 2, 1, 1, 1, 2, 3 }, c[5] = { 0 }, i; |
|  | for (i = 0; i < 12; i++) |
|  | c[s[i]] ++; |
|  | for (i = 1; i < 5; i++) |
|  | printf("%d ", c[i]); |
|  | printf("\n"); |
|  | } |

程序的运行结果是

〖A〗2 3 4 4

〖B〗4 3 3 2

〖C〗1 2 3 4

〖D〗1 1 2 3

在for(i = 0; i < 12; i++)c[s[i]] ++中, 数组元素s[i]的值作为数组c的下标, 当退出循环时, 数组c的4个元素的值分别为4、3、3、2。  
因此B选项正确。

【29】以下能正确定义字符串的语句是

〖A〗char str = "\x43";

〖B〗char str[] = "\0";

〖C〗char str = ' ';

〖D〗char str[] = { '\064' };

C语言中, 字符串是用一对双引号括起来的字符序列, 并用字符型数组来存放, 故C选项和D选项不属于字符串, A选项定义的是一个字符变量str, 却用来存放字符串, 显然也不正确, 因此B选项正确。

【30】以下关于字符串的叙述中正确的是

〖A〗空串比空格打头的字符串小

〖B〗两个字符串中的字符个数相同时才能进行字符串大小的比较

〖C〗可以用关系运算符对字符串的大小进行比较

〖D〗C语言中有字符串类型的常量和变量

字符串比较大小是以第1个不相同字符的大小为标准的, 跟长度没有关系, B选项不正确; 字符串比较大小除了使用库函数strcn3()以外, 就只能靠自己写代码来实现了, 而不能通过关系运算符来比较大小。  
因为字符串在表达式中相当于const char\*, 即常字符指针, 代表的是字符串的首地址, 关系运算符会将两个字符串的首地址值比较大小, 这是毫无意义的。  
C选项不正确。  
C语言中只有字符串常量而没有字符串变量, D选项描述不正确; 空串的长度为0, 而以空格打头的字符串的长度至少为1, 因此A选项是正确。

【31】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void fun(char \*a, char \*b) { |
|  | while (\*a == '\*') |
|  | a++; |
|  | while (\*b = \*a) { |
|  | b++; |
|  | a++; |
|  | } |
|  | } |
|  | void main() { |
|  | char \*s = "\*\*\*\*\*a\*b\*\*\*\*", t[80]; |
|  | fun(s, t); |
|  | puts(t); |
|  | } |

程序的运行结果是

〖A〗ab

〖B〗a \*b

〖C〗\*\*\*\*\*a \* b

〖D〗a \*b \* \*\*\*

在函数fun(char \* a, char \*b)中, while (\*a == '\*') a++的功能是：如果\*a的内容为'\*', 则a指针向后移动, 直到遇到非'\*'字符为止, 退出循环进入下一个while循环, 在while(\*b = \*a) { b++; a++; } 中, 把字符数组a中的字符逐个赋给字符数组b。  
所以在主函数中, 执行fun(s, t)语句后, 字符数组t中的内容为"a\*b\*\*\*\*"。  
因此D选项正确。

【32】下列函数的功能是

复制代码

C

|  |  |
| --- | --- |
|  | void fun(char \*a, char \*b) { |
|  | while ((\*b = \*a) != '\0') { |
|  | a++; |
|  | b++; |
|  | } |
|  | } |

〖A〗将a所指字符串赋给b所指空间

〖B〗使指针b指向a所指字符串

〖C〗将a所指字符串和b所指字符串进行比较

〖D〗检查a和b所指字符串中是否有'\0'

While循环条件为：(\*b = \*a) != '\0', 执行时先把指针a所指向的字符赋给指针b所在内存单元, 如果该字符不是结束标识"\0", 执行循环体a++; b++;, 指针a、b分别指向下一个字符单元。  
再判断循环条件, 如果成立, 继续把指针a所指向的字符赋给指针b所在内存单元, 直到遇到结束标识为止。  
因此A选项正确。

【33】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | int fun(int x) { |
|  | int p; |
|  | if (x == 0 || x == 1) |
|  | return (3); |
|  | p = x - fun(x - 2); |
|  | return p; |
|  | } |
|  | void main() { |
|  | printf("%d\n", fun(7)); |
|  | } |

执行后的输出结果是

〖A〗2

〖B〗3

〖C〗7

〖D〗0

因为fun(int x)是一个递归函数, 所以主函数中fun(7)经过3次递归调用, 其过程可以描述为"fun(7)=7-fun(5)=7-(5-fun(3))=7-(5-(3-fun(1)))=7-(5-(3-3))=7-5=2", 所以最后输出结果为2。  
因此A选项正确。

【34】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | int fun() { |
|  | static int x = 1; |
|  | x \*= 2; |
|  | return x; |
|  | } |
|  | void main() { |
|  | int i, s = 1; |
|  | for (i = 1; i <= 2; i++) |
|  | s = fun(); |
|  | printf("%d\n", s); |
|  | } |

程序运行后的输出结果是

〖A〗0

〖B〗1

〖C〗4

〖D〗8

fun函数中的x为静态局部变量, 占用固定的内存单元, 下一次调用时仍可保留上次调用时的值。  
也就是说, 如果多次调用fun函数, x的定义只在第一次调用时有效, 从第二次调用开始, x的定义相当于不存在, 直接使用x的值。  
主函数中调用两次fun函数：第一次调用：x = 1, x = x \*2 = 2, s = 2; 第二次调用：(直接用上次x的值)x = x \*2 = 4, s = 4。  
因此C选项正确。

【35】以下结构体类型说明和变量定义中正确的是

〖A〗struct REC; { int n; char c; }; REC t1, t2;

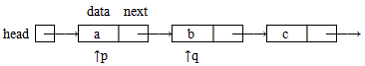
〖B〗typedef struct { int n; char c; } REC; REC t1, t2;

〖C〗typedef struct REC; { int n = 0; char c = 'A'; } t1, t2;

〖D〗struct { int n; char c; } REC; REC t1, t2;

本题考查结构体的相关知识, 选项A中struct REC后面不能有分号, C选项中typedef struct REC的后面也不能有分号, 选项D中REC已经是结构体变量, 不能当做结构体类型来使用。

【36】假定已建立以下链表结构, 且指针p和q已指向如图所示的结点：



则以下选项中可将q所指结点从链表中删除并释放该结点的语句组是

〖A〗p->next = q->next; free(q);

〖B〗p = q->next; free(q);

〖C〗p = q; free(q);

〖D〗(\*p).next = (\*q).next; free(p);

本题考查删除链表中的节点操作, 其方法是将要删除节点的上个节点的下个节点指向要删除节点的下个节点, 然后释放该要删除节点, 所以选项A正确。

【37】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。

【38】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | #define N 5 |
|  | #define M N + 1 |
|  | #define f(x) (x \* M) |
|  | void main() { |
|  | int i1, i2; |
|  | i1 = f(2); |
|  | i2 = f(1 + 1); |
|  | printf("%d %d\n", i1, i2); |
|  | } |

程序的运行结果是

〖A〗12 7

〖B〗12 12

〖C〗11 11

〖D〗11 7

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(2) = (2 \* N + 1) = 11, 执行f(1 + 1) = (1 + 1 \* N + 1) = 7。  
选项D正确。

【39】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int a = 5, b = 1, t; |
|  | t = (a << 2) | b; |
|  | printf("%d\n", t); |
|  | } |

程序运行后的输出结果是

〖A〗1

〖B〗11

〖C〗6

〖D〗21

本题考查位运算, 以及按位或操作, 将a左移一位相当于是将a乘以了4, 所以a << 2等于20, 二进制表示为00010100, 与b按位或得到00010101, 即21, 所以选项D正确。

【40】设fp为指向某二进制文件的指针, 且已读到此文件末尾, 则函数feof(fp)的返回值为

〖A〗0

〖B〗'\0'

〖C〗非0值

〖D〗NULL

本题考查文件的定位, feof函数的用法是从输入流读取数据, 如果到达文件末尾(遇文件结束符), eof函数值为非零值, 否则为0, 所以选项C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：找出N×N矩阵中每列元素中的最大值, 并按顺序依次存放于形参b所指的一维数组中。  
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) a[0][i] |
|  | (2) < |
|  | (3) x, y |

填空1:数组b用于存放每列元素中的最大值, 首先将第i列的第一个数赋给b[i], 然后用b[i]与其他数进行比较, 因此此空应填a[0][i]。  
填空2 : if条件表达式表示当b[i]小于a[j][i]时, 就把a[j][i]的值赋给b[i], 因此此空应该填<。  
填空3 : fun函数的调用, 通过fun(int(\*a)[N], int \*b)可知, 此空应该填x, y。

【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：将主函数中两个变量的值进行交换。例如, 若变量a中的值为8, b中的值为3, 则程序运行后, a中的值为3, b中的值为8。  
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) void fun(int \* x, int \*y) |
|  | (2) t = \*x; |
|  | \*x = \*y; |
|  | \*y = t; |

(1) 本题考查指针变量作函数参数。  
一般变量作参数时, 不能改变实参的值, 采用指针变量作为参数则能够改变实参的值。  
主函数中fun函数的调用方式表明fun函数的参数应当为指针类型。  
(2) 此处是一个变量交换操作错误, 可知x、y是指针类型, 变量类型不同, 因而x、y不能与t进行值的交换。

【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, 编写函数int fun(int lim, int aa[MAX]), 其功能是求出小于或等于lim的所有素数并放在aa数组中, 并返回所求出的素数的个数。  
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

|  |  |
| --- | --- |
|  | int fun(int lim, int aa[MAX]) { |
|  | int i, j, k = 0; |
|  | for (i = 2; i <= lim; i++) |
|  | /\*求出小于或等于lim的全部素数\*/ { |
|  | for (j = 2; j < i; j++) |
|  | if (i % j == 0) |
|  | break; |
|  | if (j >= i) |
|  | aa[k++] = i; |
|  | /\*将求出的素数放入数组aa中\*/ |
|  | } |
|  | return k; |
|  | /\*返回所求出的素数的个数\*/ |
|  | } |

本程序使用for循环语句查找小于lim的所有数, 使用内嵌的循环判断语句判断该数是否为素数。  
在做这道题时, 需要重点掌握素数的判定方法：  
for (j = 2; j < i; j++)  
if (i % j == 0) break;  
…