[**【计算机二级C语言】卷005**](https://www.cnblogs.com/mjios/p/12452574.html)

**选择题**

**公共知识**

【1】一个栈的初始状态为空。现将元素1、2、3、4、5、A、B、C、D、E依次入栈, 然后再依次出栈, 则元素出栈的顺序是()。

〖A〗12345ABCDE

〖B〗EDCBA54321

〖C〗ABCDE12345

〖D〗54321EDCBA

栈是先进后出的原则组织数据, 所以入栈最早的最后出栈, 所以选择B)。

【2】下列叙述中正确的是()。

〖A〗循环队列有队头和队尾两个指针, 因此, 循环队列是非线性结构

〖B〗在循环队列中, 只需要队头指针就能反映队列中元素的动态变化情况

〖C〗在循环队列中, 只需要队尾指针就能反映队列中元素的动态变化情况

〖D〗循环队列中元素的个数是由队头指针和队尾指针共同决定

循环队列有队头和队尾两个指针, 但是循环队列仍是线性结构的, 所以A)错误; 在循环队列中只需要队头指针与队尾两个指针来共同反映队列中元素的动态变化情况, 所以B)与C)错误。

【3】在长度为n的有序线性表中进行二分查找, 最坏情况下需要比较的次数是()。

〖A〗



〖B〗



〖C〗



〖D〗



当有序线性表为顺序存储时才能用二分法查找。可以证明的是对于长度为n的有序线性表, 在最坏情况下, 二分法查找只需要比较



次, 而顺序查找需要比较n次。

【4】下列叙述中正确的是()。

〖A〗顺序存储结构的存储一定是连续的, 链式存储结构的存储空间不一定是连续的

〖B〗顺序存储结构只针对线性结构, 链式存储结构只针对非线性结构

〖C〗顺序存储结构能存储有序表, 链式存储结构不能存储有序表

〖D〗链式存储结构比顺序存储结构节省存储空间

链式存储结构既可以针对线性结构也可以针对非线性结构, 所以B)与C)错误。  
链式存储结构中每个结点都由数据域与指针域两部分组成, 增加了存储空间, 所以D)错误。

【5】数据流图中带有箭头的线段表示的是()。

〖A〗控制流

〖B〗事件驱动

〖C〗模块调用

〖D〗数据流

数据流图中带箭头的线段表示的是数据流, 即沿箭头方向传送数据的通道, 一般在旁边标注数据流名。

【6】在软件开发中, 需求分析阶段可以使用的工具是()。

〖A〗N - S图

〖B〗DFD图

〖C〗PAD图

〖D〗程序流程图

在需求分析阶段可以使用的工具有数据流图DFD图, 数据字典DD, 判定树与判定表, 所以选择B)。

【7】在面向对象方法中, 不属于"对象"基本特点的是()。

〖A〗一致性

〖B〗分类性

〖C〗多态性

〖D〗标识唯一性

对象有如下一些基本特点：标识唯一性、分类性、多态性、封装性、模块独立性好。  
所以选择A)。

【8】一间宿舍可住多个学生, 则实体宿舍和学生之间的联系是()。

〖A〗一对一

〖B〗一对多

〖C〗多对一

〖D〗多对多

因为一间宿舍可以住多个学生即多个学生住在一个宿舍中, 但一个学生只能住一间宿舍, 所以实体宿舍和学生之间是一对多的关系。

【9】在数据管理技术发展的三个阶段中, 数据共享最好的是()。

〖A〗人工管理阶段

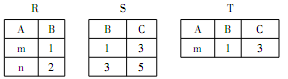
〖B〗文件系统阶段

〖C〗数据库系统阶段

〖D〗三个阶段相同

数据管理发展至今已经历了三个阶段：人工管理阶段、文件系统阶段和数据库系统阶段。  
其中最后一个阶段结构简单, 使用方便逻辑性强物理性少, 在各方面的表现都最好, 一直占据数据库领域的主导地位, 所以选择C)。

【10】有三个关系R、S和T如下：  
由关系R和S通过运算得到关系T, 则所使用的运算为()。



〖A〗笛卡尔积

〖B〗交

〖C〗并

〖D〗自然连接

自然连接是一种特殊的等值连接, 它要求两个关系中进行比较的分量必须是相同的属性组, 并且在结果中把重复的属性列去掉, 所以根据T关系中的有序组可知R与S进行的是自然连接操作。

**专业知识**

【11】下列叙述中错误的是()。

〖A〗C程序可以由多个程序文件组成

〖B〗一个C语言程序只能实现一种算法

〖C〗C程序可以由一个或多个函数组成

〖D〗一个C函数可以单独作为一个C程序文件存在

在一个C语言程序中可以进行多种算法的实现, 对算法的个数没有规定, 所以B)错误。

【12】以下选项中, 能用作数据常量的是()。

〖A〗115L

〖B〗0118

〖C〗1.5e1.5

〖D〗o115

A)选项中115L表示115是长整型数据, 合法。  
B)选项是八进制常量的表示方法, 但是在八进制中不能含有数字8, 所以B)错误。  
C)选项中e后面应该是整数不能是小数1.5, 所以C)错误。  
D)选项中八进制常量应该是数字"0"开始, 而不是字母"o"开始。

【13】按照C语言规定的用户标识符命名规则, 不能出现在标识符中的是()。

〖A〗大写字母

〖B〗下划线

〖C〗数字字符

〖D〗连接符

C语言中标识符由字母、下划线、数字组成, 且开头必须是字母或下划线。  
所以D)选项中的连接符不合法。

【14】设变量已正确定义并赋值, 以下正确的表达式是()。

〖A〗x = y + z + 5, ++y

〖B〗int(15.8 % 5)

〖C〗x = y \*5 = x + z

〖D〗x = 25 % 5.0

B)与D)选项中取模运算符%的左右两个操作数均应为整数, 所以B)、D)错误。  
C)选项中不能将x + z的值赋给表达式y \* 5, 所以C)错误。

【15】设有定义：int x = 2;, 以下表达式中, 值不为6的是()。

〖A〗2 \*x, x += 2

〖B〗x++, 2 \*x

〖C〗x \*= (1 + x)

〖D〗x \*= x + 1

A)选项中逗号表达式先计算第一表达式2 \* x, 然后计算表达式x += 2的值, 即x = x + 2即4, 整个逗号表达式为第二个表达式的值4, 所以选A)。  
B)选项中首先计算逗号表达式中第一个表达式x++, 此时x为3, 再执行第二个表达式2 \*x = 2 \*3 = 6, 所以逗号表达式为第二个表达式的值6。  
C)选项的赋值表达式可以表示为x = x \* (1 + x) = 2 \* (1 + 2) = 6。  
D)选项中的表达式可以表示为x = x \* (x + 1) = 2 \*3 = 6。

【16】有以下程序：

复制代码

C

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

程序运行后的输出结果是()。

〖A〗2,3,3

〖B〗2,3,2

〖C〗2,3,1

〖D〗2,2,1

z = x++, y++, ++y; 因为赋值运算符的优先级高于逗号运算符的优先级, 所以可以将上式表示成(z = x++), (y++), (++y)。  
然后从左向右先计算表达式z = x++, 因为x++先使用后自增, 所以z的值为1, x的值为2。  
再计算逗号表达式第二个表达式y++, 此时y的值为2, 最后计算第三个表达式++y, y的值为3。

【17】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | char c1, c2; |
|  | c1 = 'A' + '8' - '4'; |
|  | c2 = 'A' + '8' - '5'; |
|  | printf("%c,%d\n", c1, c2); |
|  | } |

已知字母A的ASCII码为65, 程序运行后的输出结果是()。

〖A〗E,68

〖B〗D,69

〖C〗E,D

〖D〗输出无定值

本题目中字符变量c1是字符'A'的ASCII码加上4, 即69所对应的字符'E'。  
字符变量c2是字符'A'的ASCII码加上3, 即68所对应的字符'D'。  
但是打印输出时, c1以%c的格式输出, 所以是E, c2以%d的格式输出, 所以是68。

【18】若有定义int x, y; 并已正确给变量赋值, 则以下选项中与表达式(x - y) ? (x++) : (y++)中的条件表达式(x - y)等价的是()。

〖A〗(x - y < 0 || x - y > 0)

〖B〗(x - y < 0)

〖C〗(x - y > 0)

〖D〗(x - y == 0)

条件表达式：x=表达式1?表达式2:表达式3 的含义是：先求解表达式1, 若为非0(真), 则求解表达式2, 将表达式2的值赋给x。  
若表达式1的值为0(假), 则求解表达式3, 将表达式3的值赋给x。  
在本题中与表达式1 : (x - y)等价的是(x - y < 0 || x - y > 0)。

【19】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int x = 1, y = 0; |
|  | if ( !x) |
|  | y++; |
|  | else if (x == 0) |
|  | if (x) |
|  | y += 2; |
|  | else |
|  | y += 3; |
|  | printf("%d\n", y); |
|  | } |

程序运行后的输出结果是()。

〖A〗3

〖B〗2

〖C〗1

〖D〗0

在if else 语句中 else 总是与离它最近的if配对。  
本题目中x为1所以 !x为0, 所以执行else if语句中的内容, 判断(x == 0)是否成立, 因为x为1所以条件不成立, 所以else if内部的if…else 语句不再执行, 所以y的值还是初始值0。

【20】若有定义：float x = 1.5; int a = 1, b = 3, c = 2; 则正确的switch语句是()。

〖A〗switch (a + b) { case 1: printf("\*\n"); case 2 + 1 : printf("\*\*\n"); }

〖B〗switch ((int) x); { case 1: printf("\*\n"); case 2: printf("\*\*\n"); }

〖C〗switch (x) { case 1.0: printf("\*\n"); case 2.0: printf("\*\*\n"); }

〖D〗switch (a + b) { case 1: printf("\*\n"); case c: printf("\*\*\n"); }

B)选项中switch((int) x); 语句中不应该有最后的分号。  
switch (expr1) , 中的expr1不能用浮点类型或long类型, 也不能为一个字符串, 所以C)错误。  
case后面常量表达式的类型必须与switch后面表达式的类型一致, 所以D)错误。

【21】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int y = 9; |
|  | for (; y > 0; y--) |
|  | if (y % 3 == 0) |
|  | printf("%d", --y); |
|  | } |

程序的运行结果是()。

〖A〗852

〖B〗963

〖C〗741

〖D〗875421

第一次for循环, y的值为9, y % 3的值为0, 满足条件打印--y, 即先减一后打印, 所以打印8; 第二次for循环, y的值为7, y % 3的值为1, 不执行打印语句; 第三次for循环, y的值为6, y % 3的值为0, 满足条件打印--y, 即先减一后打印, 所以打印5; 第四次for循环, y的值为4, 不满足if条件, 不执行打印语句; 第五次for循环, y的值为3, 满足if条件, 打印输出2; 第六次for循环, y的值为1, 不满足条件, 不执行打印语句。

【22】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int i, j, m = 1; |
|  | for (i = 1; i < 3; i++) { |
|  | for (j = 3; j > 0; j--) { |
|  | if (i \*j > 3) |
|  | break; |
|  | m \*= i \* j; |
|  | } |
|  | } |
|  | printf("m=%d\n", m); |
|  | } |

程序运行后的输出结果是()。

〖A〗m=4

〖B〗m=2

〖C〗m=6

〖D〗m=5

第一次外循环i的值为1, 第一次内循环j的值为3, 不满足条件执行m \*= i \* j即m的值为3; 第二次j的值为2, 不满足条件执行m \*= i \* j, 即m的值为6; 第三次j的值为1, 不满足条件执行m \*= i \* j, 即m的值仍为6。  
第二次外循环i的值为2, j的值为3, 满足条件, 执行break语句, 跳出循环。

【23】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | int fun(int x, int y) { |
|  | if (x != y) |
|  | return ((x + y) / 2); |
|  | else |
|  | return (x); |
|  | } |
|  | void main() { |
|  | int a = 4, b = 5, c = 6; |
|  | printf("%d\n", fun(2 \*a, fun(b, c))); |
|  | } |

〖A〗6

〖B〗3

〖C〗8

〖D〗12

本题中第一次调用为fun(8, fun(5, 6)), 因为fun(5, 6)返回值为5, 所以第二次调用为fun(8, 5) = 6。  
所以选择A)。

【24】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | int add(int a, int b) { |
|  | return (a + b); |
|  | } |
|  | void main() { |
|  | int k, (\*f)(), a = 5, b = 10; |
|  | f = add; |
|  | … |
|  | } |

则以下函数调用语句错误的是()。

〖A〗k = f(a, b);

〖B〗k = add(a, b);

〖C〗k = (\*f)(a, b);

〖D〗k = \*f(a, b);

D)选项中\*f(a, b)表示调用后返回一个指向整型数据的地址指针, 即该函数的返回值为指针类型, 所以不能将其赋值给整形变量k。

【25】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int n, \*p = NULL; |
|  | \*p = &n; |
|  | printf("Input n:"); |
|  | scanf("%d", &p); |
|  | printf("output n:"); |
|  | printf("%d\n", p); |
|  | } |

该程序试图通过指针p为变量n读入数据并输出, 但程序有多处错误, 以下语句正确的是()

〖A〗int n, \*p = NULL;

〖B〗\*p = &n;

〖C〗scanf("%d", &p)

〖D〗printf("%d\n", p);

选项B)的正确写法应为p = &n; 选项C)的正确写法应为scanf("%d", p); 选项D)的正确写法应为 printf("%d\n", \*p)。

【26】下列定义数组的语句中, 正确的是()。

〖A〗#define N 10 int x[N];

〖B〗int N = 10; int x[N];

〖C〗int x[0..10];

〖D〗int x[];

数组说明的一般形式为：类型说明符 数组名[常量表达式]。  
B)中N是变量, 不能用变量定义数组长度。  
C)选项中数组长度是非法的一串数字。  
定义数组时必须为其指明长度, D)选项中数组长度为空, 所以非法。

【27】有以下程序：

复制代码

C

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

程序运行后的输出结果是()。

〖A〗0 3 0 4

〖B〗2 0 5 0

〖C〗3 3 4 4

〖D〗3 0 4 0

第一次循环i为0, i % 2为0, 执行switch(a[0] % 2)中的case 0语句后内容即a[0] ++, a[0]的值变成3; 第二次循环i的值为1, i % 2为1, 执行case 1 : a[i] = 0; 所以a[1]的值变成0; 第三次循环i的值为2, i % 2为0, 执行switch(a[2] % 2)中的case 1语句后的内容 a[2] --, a[2]的值变成4; 第四次循环i的值为3, i % 2为1, 执行case 1 : a[i] = 0; 所以a[3]的值变成0。

【28】有以下程序：

复制代码

C

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

程序运行后的输出结果是()。

〖A〗3

〖B〗4

〖C〗1

〖D〗9

每次内循环只循环一次就结束, 第一次外循环时t = t + b[0][b[0][0]] = 1 + b[0][0] = 1 + 0 = 1; 第二次外循环时t = t + b[1][b[1][1]] = 1 + b[1][1] = 1 + 1 = 2; 第三次外循环时t = t + b[2][b[2][2]] = 2 + b[2][2] = 2 + 2 = 4。

【29】以下语句中存在语法错误的是()。

〖A〗char ss[6][20]; ss[1] = "right?";

〖B〗char ss[][20] = { "right?" };

〖C〗char \*ss[6]; ss[1] = "right?";

〖D〗char \*ss[] = { "right?" };

数组定义后, 不可以对数组整体赋值, s是二维数组, 因ss[1]是一维字符数组, 即字符串, 字符串赋值可以使用strcpy(ss[1], "right"); 这样的形式, 而选项A)中对二维数组中的第"1"维(相当于一个一维数组)赋值, 是不可以的。  
选项B)和D)是定义时对数组初始化, 这是可以的。  
选项C)中, 将字符串在内存中的首地址赋给指针数组的一个元素, 这是可以的。

【30】以下不能将s所指字符串正确复制到t所指存储空间的是()

〖A〗do { \*t++ = \*s++; } while (\*s);

〖B〗for (i = 0; t[i] = s[i]; i++);

〖C〗while (\*t = \*s) { t++; s++ }

〖D〗for (i = 0, j = 0; t[i++] = s[j++]; );

do { \*t++ = \*s++; } while (\*s); 不能因为当\*s = '\0'时, while (\*s) 跳出循环, 这样字符串结束标志'\0'没有复制给\*t, 造成\*t不完整。

注意, \*t++ = \*s++是先执行t = \*s, 然后才进行t = t + 1, s = s + 1。  
B), C), D)三项都能将'\0'复制过去。

【31】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void swap(char \*x, char \*y) { |
|  | char t; |
|  | t = \*x; |
|  | \*x = \*y; |
|  | \*y = t; |
|  | } |
|  | void main() { |
|  | char \*s1 = "abc", \*s2 = "123"; |
|  | swap(s1, s2); |
|  | printf("%s,%s\n", s1, s2); |
|  | } |

程序执行后的输出结果是()。

〖A〗321,cba

〖B〗abc,123

〖C〗123,abc

〖D〗1bc,a23

字符串是一个特殊的数组, 所以按照数组的规则, s1应该指向的是数组的首地址, 即"abc"的第一个字符的地址。  
s2指向的是"123"的第一个字符的地址。  
调用swap函数之后交换的是两个字符串的第一个字符'a'和'1'的内容, 所以打印输出为D)。

【32】有以下函数：

复制代码

C

|  |  |
| --- | --- |
|  | int fun(char \*x, char \*y) { |
|  | int n = 0; |
|  | while ((\*x == \*y) && \*x != '\0') { |
|  | x++; |
|  | y++; |
|  | n++; |
|  | } |
|  | return n; |
|  | } |

函数的功能是()。

〖A〗将y所指字符串赋给x所指存储空间

〖B〗查找x和y所指字符串中是否有'\0'

〖C〗统计x和y所指字符串中最前面连续相同的字符个数

〖D〗统计x和y所指字符串中相同的字符个数

本题中由循环条件可知遇到'\0'或x与y所指的字符的值不等中的一个条件时就结束, 所以功能是统计x和y所指字符串中最前面连续相同的字符个数。

【33】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 <= 3; i++) |
|  | s \*= fun(); |
|  | printf("%d\n", s); |
|  | } |

程序运行后的输出结果是()。

〖A〗10

〖B〗30

〖C〗0

〖D〗64

本题目中静态局部变量x, 在静态存储区内分配存储单元。  
在程序整个运行期间都不释放。  
所以第一次循环s的值为2, 第二次循环中, 返回的x的值为4, 所以s的值为8, 第三次循环, 返回的x的值为8, 所以s的值为64。

【34】在一个C源程序文件中所定义的全局变量, 其作用域为()。

〖A〗由具体定义位置和extern说明来决定范围

〖B〗所在程序的全部范围

〖C〗所在函数的全部范围

〖D〗所在文件的全部范围

全局变量的作用域是从声明处到文件的结束。  
所以选择A)

【35】以下叙述中错误的是()。

〖A〗可以通过typedef增加新的类型

〖B〗可以用typedef将已存在的类型用一个新的名字来代表

〖C〗用typedef定义新的类型名后, 原有类型名仍有效

〖D〗用typedef可以为各种类型起别名, 但不能为变量起别名

关键字typedef的作用只是将C语言中的已有的数据类型作了置换, 并不是增加新的类型, 所以A)错误。

【36】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | struct S { |
|  | int a, b; |
|  | } data[2] = { 10, 100, 20, 200 }; |
|  | void main() { |
|  | struct S p = data[1]; |
|  | printf("%d\n", ++(p.a)); |
|  | } |

程序运行后的输出结果是()。

〖A〗10

〖B〗11

〖C〗20

〖D〗21

声明 data 是结构 S 数组。  
初始化 data[0].a = 10; data[0].b = 100; data[1].a = 20; data[1].b = 200。  
主函数中p = data[1]; 即 p.a = data[1].a; p.b = data[1].b; 执行语句printf("%d\n", ++(p.a)); 打印输出时 p.a 先增1再打印。  
p.a = data[1].a = 20, 先增1 等于21。

【37】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | #define PT 3.5; |
|  | #define S(x) PT \* x \* x; |
|  | void main() { |
|  | int a = 1, b = 2; |
|  | printf("%4.1f\n", S(a + b)); |
|  | } |

程序运行后的输出结果是()。

〖A〗7.5

〖B〗31.5

〖C〗程序有错无输出结果

〖D〗14.0

宏定义不是C语句, 末尾不需要有分号。  
所以语句printf("%4.1f\n", S(a + b)); 展开后为printf("%4.1f\n", 3.5; \*a + b \* a + b; ); 所以程序会出现语法错误。

【38】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | unsigned char a = 2, b = 4, c = 5, d; |
|  | d = a | b; |
|  | d &= c; |
|  | printf("%d\n", d); |
|  | } |

程序运行后的输出结果是()。

〖A〗3

〖B〗4

〖C〗5

〖D〗6

&按位与, 如果两个相应的二进制位都为1, 则该位的结果值为1, 否则为0。  
|按位或, 两个相应的二进制位中只要有一个为1, 该位的结果值为1。  
2的二进制为00000010, 4的二进制为00000100, 所以做或运算结果为00000110, 该数与5即00000101做与操作结果为00000100, 即4。

【39】有以下程序：

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | #include <stdlib.h> |
|  | void main() { |
|  | int \*a, \*b, \*c; |
|  | a = b = c = (int \*) malloc(sizeof(int)); |
|  | \*a = 1; |
|  | \*b = 2, \*c = 3; |
|  | a = b; |
|  | printf("%d,%d,%d\n", \*a, \*b, \*c); |
|  | } |

程序运行后的输出结果是()。

〖A〗1,1,3

〖B〗2,2,3

〖C〗1,2,3

〖D〗3,3,3

malloc函数动态分配一个整型的内存空间, 然后把函数返回的地址用(int \*)强制类型转换为整型指针, 再把它赋给a, b, c, 即让指针变量a, b, c都指向刚申请的内存空间。  
所以只有最后一个赋值语句\*c = 3的值保留在了该空间内, 因为a, b, c三个指针变量均指向该空间, 所以打印该空间内的数值为3。

【40】以下叙述中正确的是()。

〖A〗当对文件的读(写)操作完成之后, 必须将它关闭, 否则可能导致数据丢失

〖B〗打开一个已存在的文件并进行了写操作后, 原有文件中的全部数据必定被覆盖

〖C〗在一个程序中当对文件进行了写操作后, 必须先关闭该文件然后再打开, 才能读到第1个数据

〖D〗C语言中的文件是流式文件, 因此只能顺序存取数据

B)选项中打开一个已存在的文件并进行了写操作后, 原有文件中的全部数据不一定被覆盖, 也可以对源文件进行追加操作等。  
C)选项中在一个程序中当对文件进行了写操作后, 不用先关闭该文件然后再打开, 才能读到第1个数据, 可以用fseek()函数进行重新定位即可。  
D)选项中, C语言中的文件可以进行随机读写。

**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：计算形参x所指数组中N个数的平均值(规定所有数均为正数), 作为函数值返回, 并将大于平均值的数放在形参y所指数组中, 在主函数中输出。  
例如, 有10个正数：46、30、32、40、6、17、45、15、48、26, 其平均值为30.500000。  
主函数中输出：46 32 40 45 48。  
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) 0.0 |
|  | (2) x[i] / N |
|  | (3) j++ |

填空1:通过读上面的程序可以看出此空考的是给变量赋初值, av代表的是平均值, 因此av的初值应该是0.0。  
填空2:通过for循环可知, 此空代表求平均值, 因此应该填写x[i] / N。  
填空3:先把大于平均值的数放在形参y所指数组中, 然后使下标值加1, 因此此空应该填j++。

【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：根据整型参数m, 计算如下公式的值。



例如, 若m = 2000, 则应输出0.000160。  
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) double fun(int m) |
|  | (2) for (i = 100; i <= m; i += 100) |
|  |  |

(1) 题目要求在函数fun中求级数前m项和, 可用循环语句, 每次计算级数中的一项, 然后累加。  
第一处错误在于定义fun(int m), 由函数的返回值可知应该定义为double fun(int m)。  
(2) for (i = 100, i <= m, i += 100) 中是一个简单的语法错误。  
for循环语句的形式为for(表达式1; 表达式2; 表达式3), 其表达式之间应以";"相隔。

【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, 已知学生的记录由学号和学习成绩构成, N名学生的数据已存入a结构体数组中。请编写函数fun, 该函数的功能是：找出成绩最低的学生记录, 通过形参返回主函数(规定只有一个最低分)。已给出函数的首部, 请完成该函数。  
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

|  |  |
| --- | --- |
|  | fun(STU a[], STU \*s) { |
|  | int i; |
|  | \*s = a[0]; |
|  | /\*先认为第1个值最小\*/ |
|  | for (i = 0; i < N; i++) |
|  | /\*如果在循环的过程中再发现比第1个值更小的则赋给\*s\*/ |
|  | if (s->s > a[i].s) |
|  | \*s = a[i]; |
|  | } |

找出结构体数组元素中的最小值。  
先认为第1个值最小, 即\*s = a[0];, 如果在循环的过程中发现比第1个值更小的, 就将指针s指向该元素, 直到找到最小元素。  
另外, 本题还涉及结构体中的指向运算符, 请考生注意。