2020-03-10 09:27

[**【计算机二级C语言】卷020**](https://www.cnblogs.com/mjios/p/12453820.html)

**选择题隐藏全部答案**

**公共知识**

【1】某二叉树的中序遍历序列为 CBADE, 后序遍历序列为 CBADE, 则前序遍历序列为()。

〖A〗EDABC

〖B〗CBEDA

〖C〗CBADE

〖D〗EDCBA

二叉树遍历可以分为3种：前序遍历(访问根结点在访问左子树和访问右子树之前)、中序遍历(访问根结点在访问左子树和访问右子树两者之间)、后序遍历(访问根结点在访问左子树和访问右子树之后)。  
二叉树的中序遍历序列为CBADE, 后序遍历序列为CBADE, 可知该树只有左子树结点, 没有右子树结点, E为根结点。  
中序遍历序列与后序遍历序列相同说明该树只有左子树没有右子树, 因此该树有5层, 从顶向下依次为EDABC。  
故本题答案为A选项。

【2】下列叙述中正确的是()。

〖A〗在循环队列中, 队头指针和队尾指针的动态变化决定队列的长度

〖B〗在循环队列中, 队尾指针的动态变化决定队列的长度

〖C〗在带链的队列中, 队头指针与队尾指针的动态变化决定队列的长度

〖D〗在带链的栈中, 栈顶指针的动态变化决定栈中元素的个数

在栈中, 栈底保持不变, 有元素入栈, 栈顶指针增加; 有元素出栈, 栈顶指针减小。  
在循环队列中, 队头指针和队尾指针的动态变化决定队列的长度。  
在循环链表中, 前一个结点指向后一个结点, 而最后一个结点指向头结点, 只有头结点是固定的。  
线性链表中, 由于前一个结点包含下一个结点的指针, 尾结点指针为空, 要插入删除元素, 只需要改变相应位置的结点指针即可, 头指针和尾指针无法决定链表长度。  
故本题答案为A选项。

【3】设栈的存储空间为 S(1 : 60), 初始状态为 top = 61。现经过一系列正常的入栈与退栈操作后, top = 1, 则栈中的元素个数为()。

〖A〗60

〖B〗59

〖C〗0

〖D〗1

栈是一种特殊的线性表, 它所有的插入与删除都限定在表的同一端进行。  
入栈运算即在栈顶位置插入一个新元素, 退栈运算即取出栈顶元素赋予指定变量。  
栈为空时, 栈顶指针top = 0, 经过入栈和退栈运算, 指针始终指向栈顶元素。  
初始状态为top = 61, 当top = 1时, 元素依次存储在单元1 : 60中, 个数为60。  
故本题答案为A选项。

【4】设顺序表的长度为n。下列排序方法中, 最坏情况下比较次数小于n(n - 1) / 2的是()。

〖A〗堆排序

〖B〗快速排序

〖C〗简单插入排序

〖D〗冒泡排序

堆排序最坏情况下比较次数为O(nlog2n), 快速排序、简单插入排序、冒泡排序最坏情况下比较次数为n(n - 1) / 2。  
故本题答案为A选项。

【5】下面属于软件定义阶段任务的是()。

〖A〗需求分析

〖B〗软件测试

〖C〗详细设计

〖D〗系统维护

软件生命周期分为3个阶段：软件定义阶段, 任务是确定软件开发工作必须完成的目标, 确定工程的可行性; 软件开发阶段, 任务是具体完成设计和实现定义阶段所定义的软件, 通常包括总体设计、详细设计、编码和测试; 软件维护阶段, 任务是使软件在运行中持久地满足用户的需要。  
需求分析属于软件定义阶段的任务。  
故本题答案为A选项。

【6】下列选项中, 不是面向对象主要特征的是()。

〖A〗复用

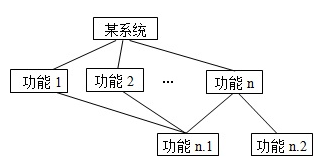
〖B〗抽象

〖C〗继承

〖D〗封装

面向对象的主要特征有抽象、继承、封装、多态等。  
故本题答案为A选项。

【7】某系统结构图如下图所示。该系统结构图的最大扇入数是()。



〖A〗3

〖B〗4

〖C〗2

〖D〗1

扇入是指调用一个给定模块的模块个数。  
图中所示功能n.1被功能1、功能2和功能3三个模块调用, 则最大扇入数为3。  
故本题答案为A选项。

【8】对数据库数据的存储方式和物理结构的逻辑进行描述的是()。

〖A〗内模式

〖B〗模式

〖C〗外模式

〖D〗用户模式

数据库系统的概念模式也称为模式, 是数据库系统中全局数据逻辑结构的描述, 全体用户的公共数据视图; 外模式也称子模式或者用户模式, 是用户的数据视图, 也就是用户所能看见和使用的局部数据的逻辑结构和特征的描述, 是与某一应用有关的数据的逻辑表示; 内模式又称物理模式, 是数据物理结构和存储方式的描述, 是数据在数据库内部的表示方式。  
可见对数据库数据的存储方式和物理结构的逻辑进行描述的是内模式。  
故本题答案为A选项。

【9】将实体-联系模型转换为关系模型时, 实体之间多对多联系在关系模型中的实现方式是()。

〖A〗建立新的关系

〖B〗建立新的属性

〖C〗增加新的关键字

〖D〗建立新的实体

将实体-联系模型转换为关系模型时, 一个m : n的联系可以转换为一个独立的关系模式, 与该联系相连的各实体的码及联系本身的属性均转换为关系的属性, 而关系的码为各实体码的组合。  
故本题答案为A选项。

【10】定义学生、教师和课程的关系模式S(S#, Sn, Sd, Dc, SA)(其属性分别为学号、姓名、所在系、所在系的系主任、年龄); C(C#, Cn, P#)(其属性分别为课程号、课程名、先修课); SC(S#, C#, G)(其属性分别为学号、课程号和成绩)。包含对非主属性部分依赖的关系是()。

〖A〗S(S#, Sn, Sd, Dc, SA)

〖B〗C(C#, Cn, P#)

〖C〗SC(S#, C#, G)

〖D〗以上三项均不是

若X→Y, 但Y不完全函数依赖于X, 则称Y对X部分函数依赖。  
关系模式S 中, (S#, Sd)→Dc, Sd →Dc, Dc不完全依赖于Sd且Sd为非主属性。  
故本题答案为A选项。

**专业知识**

【11】以下叙述中正确的是

〖A〗C语言程序所调用的函数必须放在main函数的前面

〖B〗C语言程序总是从最前面的函数开始执行

〖C〗C语言程序中main函数必须放在程序的开始位置

〖D〗C语言程序总是从main函数开始执行

C语言所调用的函数可以放在调用它的函数前, 也可以放在调用它的函数后, 所以A错误。  
C语言程序总是从main函数开始执行, 所以B错误。  
main函数的位置可以放在开始, 也可以放在中间, 也可以放在最后, 所以C错误。

【12】C语言程序中, 运算对象必须是整型数的运算符是

〖A〗&&

〖B〗/

〖C〗%

〖D〗\*

%取余运算是二目运算符, 且要求运算对象必须为整数, 所以选C。

【13】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int sum, pad, pAd; |
|  | sum = pad = 5; |
|  | pAd = ++sum, pAd++, ++pad; |
|  | printf("%d\n", pad); |
|  | } |

程序的输出结果是

〖A〗5

〖B〗6

〖C〗7

〖D〗8

C语言区分大小写, pad与pAd是两个不同的变量。  
首先将5赋值给两个变量, 由于赋值运算符的优先级高于逗号运算符, 所以计算pAd = ++sum, 此时pAd为6, sum也为6, 然后计算pAd++, pAd的值变为7, ++pad, pad的值变为6, 所以选择B。

【14】有以下程序

复制代码

C

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

程序的输出结果是

〖A〗0

〖B〗9

〖C〗3

〖D〗-12

首先计算a \* a结果为9, 执行a = a - 9, a的值为-6, 然后执行a += a, 即a = a + a = -6 + (-6), 所以值为-12。

【15】有如下程序

复制代码

C

sizeof(double)是

〖A〗一个整型表达式

〖B〗一个双精度型表达式

〖C〗一个不合法的表达式

〖D〗一种函数调用

sizeof是C语言中的一个操作符(operator), 不是函数调用, 简单的说其作用就是返回一个对象或者类型所占的内存字节数。  
所以选择A。

【16】有以下程序

复制代码

C

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

程序的输出结果是

〖A〗a=2,b=5

〖B〗a=%2,b=%5

〖C〗a=%d,b=%d

〖D〗a=%%d,b=%%d

C语言中用"%%"打印输出字符"%", 所以%%d, 输出为%d两个普通字符, 而不是格式控制符"%d"的含义, 所以打印结果为C。

【17】若有定义语句：

复制代码

C

char a = '\82';

则变量a

〖A〗说明不合法

〖B〗包含1个字符

〖C〗包含2个字符

〖D〗包含3个字符

定义语句：char a = '\82'; 定义字符变量a, '\82'转义字符错误, 三位八进制代表一个ASCII码字符, 此处只有两位数字, '\xhh'x开头的两位16进制数字代表一个ASCII码字符, 这里没有x, 因此A选项正确。

【18】有以下程序

复制代码

C

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

程序的输出结果是

〖A〗输出格式不合法,输出出错信息

〖B〗65,89

〖C〗65,90

〖D〗A,Y

本题重点考查字符变量的相关知识。  
分别给字符型变量a和b分别赋值为'A'和'Y', 即A和Y的ASSCII码, 因此输出结果为65, 89, 因此B选项正确。

【19】若变量已正确定义：for (x = 0, y = 0; (y != 99 && x < 4); x++)  
则以上for循环

〖A〗执行3次

〖B〗执行4次

〖C〗执行无限次

〖D〗执行次数不定

本题重点考查for语句的循环次数, for (x = 0, y = 0; (y != 99 && x < 4); x++) , y != 99 && x < 4即循环条件当y不等于99以及x < 4时, 执行循环语句, 当有一个条件不成立时, 均结束循环, 因此循环4次, B选项正确。

【20】对于while( !E)s;, 若要执行循环体s, 则E的取值应为：

〖A〗E等于1

〖B〗E不等于0

〖C〗E不等于1

〖D〗E等于0

while语句的一般形式为：while (表达式) 语句; 其中表达式是循环条件, 语句为循环体。  
while语句的语义是：计算表达式的值, 当值为真(非0)时, 执行循环体语句。  
该题中如要执行语句s, 则 !E应为真, 因此E应等于0, D选项正确。

【21】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int x; |
|  | for (x = 3; x < 6; x++) |
|  | printf((x % 2) ? ("\*%d") : ("#%d"), x); |
|  | printf("\n"); |
|  | } |

程序的输出结果是

〖A〗\*3#4\*5

〖B〗#3\*4#5

〖C〗\*3\*4#5

〖D〗 \*3#4#5

条件运算符组成条件表达式的一般形式为：表达式1?表达式2:表达式3

其求值规则为：如果表达式1的值为真, 则以表达式2 的值作为条件表达式的值, 否则以表达式2的值作为整个条件表达式的值。  
本题中语句printf((x % 2) ? ("%d") : ("#%d"), x);  
含义为变量x对2求余, 如果结果为真, 则按照"%d"格式输出, 否则按照#%d输出。  
因此A选项正确。

【22】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int a, b; |
|  | for (a = 1, b = 1; a <= 100; a++) { |
|  | if (b >= 20) |
|  | break; |
|  | if (b % 3 == 1) { |
|  | b = b + 3; |
|  | continue; |
|  | } |
|  | b = b - 5; |
|  | } |
|  | printf("%d\n", a); |
|  | } |

程序的输出结果是

〖A〗10

〖B〗9

〖C〗8

〖D〗7

首先注意for循环的控制条件当b >= 20或者a > 100则跳出for循环也即b < 20且a <= 100时执行for循环; 第一次进入循环a = 1, b = 1均满足循环条件, 但b % 3 == 1条件满足, 故执行b = b + 3, 得到b = 4, 注意有continue, 所以后面语句不执行, 直接跳转到a++这个语句, 所以第一次循环完之后a = 2, b = 4; 进入第二轮循环, b % 3 == 1也是满足的, 故再次b = b + 3, 此轮循环执行之后a = 3, b = 7, 进入下一轮。  
此后和前面循环同理, 都是b % 3 == 1满足, 因为每次都是加3, 而之后又去模3, 且都跳过for后面语句直接执行a++, 所以, 一直循环到b = 22跳出for循环。  
此时a为8.综合起来就是, 每次循环b增加3, a增加1, 且当b > 22时跳出循环, 结束程序。  
所以b从1增加到22, 有(22 - 1) / 3 = 7, 所以a = 1 + 7 = 8.因此C选项正确。

【23】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void fun(int x, int y, int \*c, int \*d) { |
|  | \*c = x + y; |
|  | \*d = x - y; |
|  | } |
|  | void main() { |
|  | int a = 4, b = 3, c = 0, d = 0; |
|  | fun(a, b, &c, &d); |
|  | printf("%d %d\n", c, d); |
|  | } |

程序的输出结果是

〖A〗0 0

〖B〗4 3

〖C〗3 4

〖D〗7 1

主函数中首先初始化整型变量a、b、c、d分别为4、3、0、0。  
调用函数fun, 将实参传递给形参。  
在fun函数内, \*c = 7、\*d = 1。  
返回主函数, 最后输出7和1。  
因此D选项正确。

【24】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void fun(int \*p, int \*q) { |
|  | int t; |
|  | t = \*p; |
|  | \*p = \*q; |
|  | \*q = t; |
|  | \*q = \*p; |
|  | } |
|  | void main() { |
|  | int a = 0, b = 9; |
|  | fun(&a, &b); |
|  | printf("%d %d\n", a, b); |
|  | } |

程序的输出结果是

〖A〗9 0

〖B〗0 0

〖C〗9 9

〖D〗0 9

主函数中定义整型变量a、b, 分别赋值0和9。  
将a和b的地址传递给指针变量p和q, 函数fun中将a和b的值互换后, 又将a的值赋给b。  
返回主函数, 最后输出9和9。  
因此C选项正确。

【25】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int a[] = { 2, 4, 6, 8, 10 }, x, \*p, y = 1; |
|  | p = &a[1]; |
|  | for (x = 0; x < 3; x++) |
|  | y += \*(p + x); |
|  | printf("%d\n", y); |
|  | } |

程序的输出结果是

〖A〗13

〖B〗19

〖C〗11

〖D〗15

本题通过语句"p=&a[1]"将指针变量p指向了存储单元a[1], 即使得p[0]的值为4, y的初始值为1, 然后通过3次for循环, 使得y的值分别加上p[0]、p[1]和p[2], 那么y = 1 + 4 + 6 + 8, 所以输出的y的值为19。

【26】有以下程序

复制代码

C

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

程序的输出结果是

〖A〗1 5 0

〖B〗3 5 7

〖C〗1 4 7

〖D〗3 6 9

程序首先初始化二维数组x[3][3], 然后通过3次for循环, 输出x[0][2]、x[1][1]和x[2][0]的值, 即3、5、7。  
因此B选项正确。

【27】设有某函数的说明为：int \* func(int a[10], int n);  
则下列叙述中, 正确的是

〖A〗形参a对应的实参只能是数组名

〖B〗说明中的a[10]写成a[]或\*a效果完全一样

〖C〗func的函数体中不能对a进行移动指针(如a++)的操作

〖D〗只有指向10个整数内存单元的指针, 才能作为实参传给a

本题重点考查返回指针值的函数相关知识, 返回指针值的函数, 一般定义形式为：类型名\*函数名(参数列表)。  
func函数被定义为指针型函数, 它的形参int a[10]和 int n, 分别指一维数组a[0]和整型变量n。

【28】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | char fun(char \*c) { |
|  | if (\*c <= 'Z' && \*c >= 'A') |
|  | \*c -= 'A' - 'a'; |
|  | return \*c; |
|  | } |
|  | void main() { |
|  | char s[81], \*p = s; |
|  | gets(s); |
|  | while (\*p) { |
|  | \*p = fun(p); |
|  | putchar(\*p); |
|  | p++; |
|  | } |
|  | printf("\n"); |
|  | } |

若运行时从键盘上输入  
OPEN THE DOOR<回车>  
程序的输出结果是

〖A〗OPEN THE DOOR

〖B〗oPEN tHE dOOR

〖C〗open the door

〖D〗Open The Door

字符串输入函数gets的功能是从标准输入设备键盘上输入一个字符串。  
首先使指针变量p指向字符串的首字母, while循环语句中对字符串的每个字符进行fun函数操作。  
fun函数的功能是, 将字符串中的大写字母变小写字母。  
将处理结果返回主函数, 主函数通过putchar()字符输出函数进行输出。  
因此C选项正确。

【29】设有定义语句：

复制代码

C

char \*aa[2] = { "abcd", "ABCD" };

则以下叙述正确的是

〖A〗aa[0]存放了字符串"abcd"的首地址

〖B〗aa数组的两个元素只能存放含有4个字符的一维数组的首地址

〖C〗aa数组的值分别是字符串"abcd"和"ABCD"

〖D〗aa是指针变量, 它指向含有两个元素的字符型数组

本题中定义了一个指针数组, 数组中存放的是字符指针, aa数组的两个元素分别包含了5个字符, 所以可以存放5个字符的一维数组的首地址, 所以B选项错误。  
aa数组的值是指向那两个字符串的字符指针, 而不是那两个字符串, 所以C选项错误。  
aa本身是一个指向两个字符指针的指针, 所以D选项错误。

【30】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | int fun(char \*s) { |
|  | char \*p = s; |
|  | while (\*p != 0) |
|  | p++; |
|  | return (p - s); |
|  | } |
|  | void main() { |
|  | printf("%d\n", fun("goodbey!")); |
|  | } |

程序的输出结果是

〖A〗0

〖B〗6

〖C〗7

〖D〗8

本题主要考查了函数定义及调用, 以及while循环语句, 执行fun函数后, 首先是找到最后一个字符的地址, 然后减去第一个地址得到的就是这个字符串的长度, 所以本题答案为8。  
答案为D选项。

【31】有以下程序

复制代码

C

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

程序的输出结果是

〖A〗9

〖B〗14

〖C〗10

〖D〗15

考查函数的递归调用, 只需要一步一步递归调用fun函数即可, 执行fun(5) = 5 + (4 + (3 + (2 + fun(1)))) = 15, 所以答案为D选项。

【32】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | int d = 1; |
|  | void fun(int p) { |
|  | int d = 5; |
|  | d += p++; |
|  | printf("%d ", d); |
|  | } |
|  | void main() { |
|  | int a = 3; |
|  | fun(a); |
|  | d += a++; |
|  | printf("%d\n", d); |
|  | } |

程序的输出结果是

〖A〗8 4

〖B〗9 6

〖C〗9 4

〖D〗8 5

本题主要考查局部变量和全局变量的区别, 题目中函数之外定义的d为全局变量, 函数fun内定义的d为局部变量, 所以执行fun函数后, 打印出8, 执行完d += a++; 后d变为4, 所以答案为A。

【33】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | int fun(int a) { |
|  | int b = 0; |
|  | static int c = 3; |
|  | a = (c++, b++); |
|  | return (a); |
|  | } |
|  | void main() { |
|  | int a = 2, i, k; |
|  | for (i = 0; i < 2; i++) |
|  | k = fun(a++); |
|  | printf("%d\n", k); |
|  | } |

程序的输出结果是

〖A〗4

〖B〗0

〖C〗1

〖D〗2

题目中fun函数被执行了两次, 两次执行后, 都是返回了函数内部的b值, 所以答案为0, 即B选项正确。

【34】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | char c[2][5] = { "6934", "8254" }, \*p[2]; |
|  | int i, j, s = 0; |
|  | for (i = 0; i < 2; i++) |
|  | p[i] = c[i]; |
|  | for (i = 0; i < 2; i++) |
|  | for (j = 0; p[i][j] > 0 && p[i][j] <= '9'; j += 2) |
|  | s = 10 \* s + p[i][j] - '0'; |
|  | printf("%d\n", s); |
|  | } |

程序的输出结果是

〖A〗693825

〖B〗69825

〖C〗63825

〖D〗6385

本题的意图在于将c中两个字符串的偶数位置的数字组成一个新数, 找到的四个数字分别为6、3、8、5, 组成的数为6385, 所以答案为D选项。

【35】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | #define SQR(X) X \* X |
|  | void main() { |
|  | int a = 10, k = 2, m = 1; |
|  | a /= SQR(k + m) / SQR(k + m); |
|  | printf("%d\n", a); |
|  | } |

程序的输出结果是

〖A〗0

〖B〗1

〖C〗9

〖D〗10

本题考查宏定义, 宏定义只是做个简单的替换, 执行SQR(k + m) / SQR(k + m) = k + m \* k + m / k + m \* k + m = 15 / 2, a /= SQR(k + m) / SQR(k + m)的结果为1, 选项B正确。

【36】有以下程序

复制代码

C

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

程序的输出结果是

〖A〗1

〖B〗0

〖C〗4

〖D〗8

本题考查位运算<<、&。  
y << 1为4, x >> 1为1, 二者与一下为0, 所以B选项正确。

【37】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | struct S { |
|  | int a; |
|  | int b; |
|  | }; |
|  | void main() { |
|  | struct S a, \*p = &a; |
|  | a.a = 99; |
|  | printf("%d\n", \_\_\_\_\_\_\_\_\_\_); |
|  | } |

程序要求输出结构体中成员a的数据, 以下不能填入横线处的内容是

〖A〗a.a

〖B〗\*p.a

〖C〗p->a

〖D〗(\*p).a

本题考查结构体变量的引用, 题目中要求输出结构体中成员a的数据, p为指针, \*p为结构体变量, 那么\*p.a可以表示为结构体中成员a的数据。

【38】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | #include <stdlib.h> |
|  | void fun(double \*p1, double \*p2, double \*s) { |
|  | s = (double \*) calloc(1, sizeof(double)); |
|  | \*s = \*p1 + \*(p2 + 1); |
|  | } |
|  | void main() { |
|  | double a[2] = { 1.1, 2.2 }, b[2] = { 10.0, 20.0 }, \*s = a; |
|  | fun(a, b, s); |
|  | printf("%5.2f\n", \*s); |
|  | } |

程序的输出结果是

〖A〗21.10

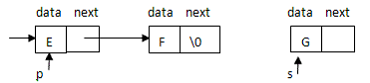
〖B〗11.10

〖C〗12.10

〖D〗 1.10

本题考查把数组名作为函数参数, 执行fun函数后, s的值并没有发生变化, 仍然是指向a, 所以输出结果为1.10, 选项D正确。

【39】若已建立以下链表结构, 指针p、s分别指向如图所示结点



则不能将s所指向结点插入到链表末尾的语句是

〖A〗p = p->next; s->next = p; p->next = s;

〖B〗s->next = '\0'; p = p->next; p->next = s;

〖C〗p = p->next; s->next = p->next; p->next = s;

〖D〗p = (\*p).next; (\*s).next = (\*p).next; (\*p).next = s;

本题考查向链表中插入节点, 选项B中, "s ->next='\0';"语句将s结点的next指针域赋值为NULL, "p=p->next; p->next=s;"这两条语句将所有结点链接, 并将s结点链接在链表的尾部。  
选项C和D表达方式相同, 只是将C选项的指针通过(\*p)转换为变量访问。  
都可以在链表末尾插入结点s。  
A选项中, "s ->next=p;"将s的next指针域赋值为p, 不是链表的结尾, 尾指针的next域为NULL。  
故本题答案为A选项。

【40】若fp已定义为指向某文件的指针, 且没有读到该文件的末尾, 则C语言函数feof(fp)的函数返回值是

〖A〗EOF

〖B〗非0

〖C〗-1

〖D〗0

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D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：把形参s所指字符串中最右边的n个字符复制到形参t所指字符数组中, 形成一个新串。若s所指字符串的长度小于n, 则将整个字符串复制到形参t所指字符数组中。  
例如, 形参s所指的字符串为：abcdefgh, n的值为5, 程序执行后t所指字符数组中的字符串应为：defgh。  
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) t, s |
|  | (2) s[i] |
|  | (3) 0或'\0' |

填空1:当给定的长度n大于该字符串s的长度, 那么把该字符串直接拷贝到t就可以了, 所以应填t, s。  
填空2:使用for循环语句, 把最右边n个字符依次添加到t中, 所以应填s[i]。  
填空3:字符串操作结束, 需要给t加一个字符串结束符, 所以应填0或'\0'。

【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：找出一个大于给定整数m且紧随m的素数, 并作为函数值返回。  
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) if (i % k == 0) |
|  | (2) if (k == i) |

(1) 判断当前数是否为素数, 若存在一个数(除1和其自身)能整除当前数, 则跳出本次循环, 所以if条件应为i % k == 0。  
(2) 如果i是素数, 则循环结束时k == i, 将该值返回。

【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void fun(int x, int pp[], int \*n), 它的功能是：求出能整除x且不是偶数的各整数, 并按从小到大的顺序放在pp所指的数组中, 这些除数的个数通过形参n返回。  
例如, 若x中的值为30, 则有4个数符合要求, 它们是1、3、5、15。  
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 x, int pp[], int \*n) { |
|  | int i, j = 0; |
|  | for (i = 1; i <= x; i = i + 2) |
|  | /\*i的初始值为1, 步长为2, 确保i为奇数\*/ |
|  | if (x % i == 0) |
|  | /\*将能整除x的数存入数组pp中\*/ |
|  | pp[j++] = i; |
|  | \*n = j; |
|  | /\*传回满足条件的数的个数\*/ |
|  | } |

本题考查：偶数的判定方法; 整除的实现。  
本题题干信息是：能整除x且不是偶数的所有整数。  
循环语句中变量i从1开始且每次增2, 所以i始终是奇数。  
整除的方法, 在前面已经讲过多次, 这里就不再赘述了。  
对于本题目要求的不是偶数的判定方法, 即该数对2求余不为0。  
除以上方法外, 还可以通过for循环语句直接把偶数筛出去, 确保参与操作的数均为奇数。