**[【计算机二级C语言】卷017](https://www.cnblogs.com/mjios/p/12453797.html)**

**选择题**

**公共知识**

【1】某带链栈的初始状态为 top = bottom = NULL, 经过一系列正常的入栈与退栈操作后, top = 10, bottom = 20。该栈中的元素个数为()。

〖A〗不确定

〖B〗10

〖C〗1

〖D〗0

带链的栈是具有栈属性的链表。  
线性链表的存储单元是不连续的, 为把存储空间中一些离散的空闲存储结点利用起来, 把所有空闲的结点组织成一个带链的栈, 称为可利用栈。  
线性链表执行删除操作运算时, 被删除的结点可以"回收"到可利用栈, 对应于可利用栈的入栈运算; 线性链表执行插入运算时, 需要一个新的结点, 可以在可利用栈中取栈顶结点, 对应于可利用栈的退栈运算。  
可利用栈的入栈运算和退栈运算只需要改动top指针即可。  
因为是不连续的存储空间, 所以top指针将不会有规律地连续变化, 因此无法据此判断栈中的元素个数。  
故本题答案为A选项。

【2】设表的长度为15。则在最坏情况下, 快速排序所需要的比较次数为()。

〖A〗105

〖B〗55

〖C〗15

〖D〗75

快速排序最坏情况下蜕化为冒泡排序, 在最坏情况下, 对长度为n的线性表排序, 冒泡排序需要比较的次数为n(n - 1) / 2。  
本题中n = 15, 15×(15 - 1) / 2 = 105。  
故本题答案为A选项。

【3】设循环队列的存储空间为Q(1 : 100), 初始状态为空。现经过一系列正常操作后, front = 49, 则循环队列中的元素个数为()。

〖A〗不确定

〖B〗49

〖C〗51

〖D〗50

循环队列是队列的一种顺序存储结构, 用队尾指针rear指向队列中的队尾元素, 用排头指针front指向排头元素的前一个位置。  
入队运算时, 队尾指针进1(即rear + 1), 然后在rear指针指向的位置插入新元素。  
退队运算时, 排头指针进1(即front + 1), 然后删除front指针指向的位置上的元素。  
只知道front的位置, 不知道rear的位置, 无法判断队列里有几个元素。  
故本题答案为A选项。

【4】某完全二叉树按层次输出(同一层从左到右)的序列为ABCDEFGH。该完全二叉树的中序序列为()。

〖A〗HDBEAFCG

〖B〗HDEBFGCA

〖C〗ABDHECFG

〖D〗ABCDEFGH

完全二叉树是指除最后一层外, 每一层上的结点数均达到最大值, 在最后一层上只缺少右边的若干结点。  
更确切地说, 如果从根结点起, 对二叉树的结点自上而下, 自左至右用自然数进行连续编号, 则深度为m且有n个结点的二叉树, 当且仅当其每一个结点都与深度为m的满二叉树中编号从1到n的结点一一对应时, 称之为完全二叉树。

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。  
即先遍历左子树, 然后访问根结点, 最后遍历右子树;  
并且在遍历左子树和右子树时, 仍然首先遍历左子树, 然后访问根结点, 最后遍历右子树。

因此, 该完全二叉树的中序序列为HDBEAFCG。  
故本题答案为A选项。

【5】下面属于整数类I的实例的是()。

〖A〗229

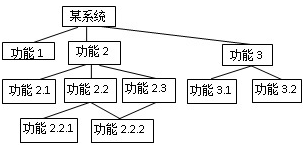
〖B〗0.229

〖C〗229E - 2

〖D〗"229"

类是具有共同属性、共同方法的对象的集合。  
229是整数, 0.229是浮点数, 229E - 2是指数, "229"是字符串。  
故本题答案为A选项。

【6】某系统总体结构如下图所示



该系统结构图的最大扇出数是()。

〖A〗3

〖B〗5

〖C〗2

〖D〗1

模块的扇出是指本模块的直属下层模块的个数。  
或者说是由一个模块直接调用的其他模块数。  
题干中某系统为一个模块, 其扇出数目为3, 功能2模块扇出数为3, 功能3模块扇出数为2, 功能2.2扇出数目为2, 则该系统结构图的最大扇出数是3。  
故本题答案为A选项。

【7】下面属于软件工程三要素的是()。

〖A〗方法、工具和过程

〖B〗方法、工具和平台

〖C〗方法、工具和环境

〖D〗工具、平台和过程

软件工程三要素是方法、工具和过程。  
故本题答案为A选项。

【8】在数据库中存储的是()。

〖A〗数据以及数据之间的联系

〖B〗数据模型

〖C〗操作信息

〖D〗数据的操作

数据库就是存放数据的仓库, 只不过数据库存放数据是按数据所提供的数据模式存放的。  
故本题答案为A选项。

【9】将数据库的结构划分成多个层次, 是为了提高数据库的()。

〖A〗逻辑独立性和物理独立性

〖B〗数据处理并发性

〖C〗管理规范性

〖D〗数据共享

数据库领域公认的标准结构是三级模式结构, 它包括外模式、概念模式、内模式, 有效地组织、管理数据, 提高了数据库的逻辑独立性和物理独立性。  
用户级对应外模式, 概念级对应概念模式, 物理级对应内模式, 使不同级别的用户对数据库形成不同的视图。  
所谓视图, 就是指观察、认识和理解数据的范围、角度和方法, 是数据库在用户"眼中"的反映, 很显然, 不同层次(级别)用户所"看到"的数据库是不相同的。  
故本题答案为A选项。

【10】定义学生选修课程的关系模式如下：

复制代码

C

|  |  |
| --- | --- |
|  | S(S#, Sn, Sd, SA)(其属性分别为学号、姓名、所在系、年龄); |
|  | C(C#, Cn, P#)(其属性分别为课程号、课程名、先选课); |
|  | SC(S#, C#, G)(其属性分别学号、课程号和成绩)。 |

检索选修了课程号为2且成绩不及格的学生的姓名的表达式是()。

〖A〗



〖B〗



〖C〗



〖D〗



π表示投影运算, 针对属性; σ表示选择运算, 针对元组; ∧为关系交; ∞表示自然连接。  
首先在SC表中选择课程号为2且成绩不及格的记录σC#=2∧G < 60(SC), 并找出这些记录中学生的学号πS#(σC#=2∧G < 60(SC)); 将检索结果与表S进行自然连接生成中间表πS#(σC#=2∧G < 60(SC))∞S, 最后在中间表中检索学生的姓名πSn(πS#(σC#=2∧G < 60(SC))∞S)。  
故本题答案为A选项。

**专业知识**

【11】以下不能用于描述算法的是

〖A〗文字叙述

〖B〗程序语句

〖C〗伪代码和流程图

〖D〗E - R图

算法的描述有伪代码、流程图、N - S结构图等。  
E - R是实体联系模型。  
所以选择D。

【12】以下选项中合法的实型常量是

〖A〗0

〖B〗3.13e - 2.1

〖C〗.914

〖D〗2.0 \* 10

A选项为整型数据。  
B选项中e后面必须为整数。  
D选项是表达式, 不是常量。  
所以选择C。

【13】以下叙述中正确的是

〖A〗a是实型变量, a = 10在C语言中是允许的, 因此可以说：实型变量中可以存放整型数

〖B〗若有int a = 4, b = 9; 执行了a = b; b = a; 之后, a的值为9, b的值为9

〖C〗若有int a = 4, b = 9; 执行了a = b后, a的值已由原值改变为b的值, b的值变为0

〖D〗若有int a = 4, b = 9; 执行了a = b; b = a; 之后, a的值为9, b的值为4

A选项中a是实型变量, a = 10在C语言中是允许的, 但是实型变量中不可以存放整型数。  
C选项中执行a = b后, a的值变为b的值, 为9, b的值不变, 仍为9。  
D选项中执行a = b后, a的值为9, b的值为9, 执行b = a后, a的值为9, b的值也为9, 所以选择B。

【14】以下选项中合法的变量是

〖A〗5a

〖B〗\_10\_

〖C〗A%

〖D〗sizeof

变量的命名要符合标识符的命名规则, 即由下划线、字母或数字组成, 且必须以下划线或字母开始, 也不能和关键字相同, 所以选择B。

【15】不能正确表示数学式



的表达式是

〖A〗a / c \* b

〖B〗a \*b / c

〖C〗a / b \* c

〖D〗a \*(b / c)

/号和\*号的优先级处于平级, 结合性都是自左至右?所以选择C项, C项式子表示的是



。

【16】有以下程序

复制代码

C

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

程序运行后的输出结果是

〖A〗3

〖B〗9

〖C〗0

〖D〗-12

首先计算a \* a, 结果为9, 然后执行a = a - 9, 即3 - 9, 结果为-6, 然后执行a = (-6) + (-6), 即a=的值为-12。  
所以选择D。

【17】若变量已正确定义, 以下选项中非法的表达式是

〖A〗a != 4 || 'b'

〖B〗'a' = 1 / 2 \* (x = y = 20, x \* 3)

〖C〗'a' % 4

〖D〗'A' + 32

B选项中'a'即表示a为字符常量, 再给'a'赋值1 / 2 \* (x = y = 20, x \* 3), 是不正确的。  
因此B选项正确。

【18】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int x = 35, B; |
|  | char z = 'B'; |
|  | B = ((x) && (z < 'b')); |
|  | printf("%d\n", B); |
|  | } |

程序运行后的输出结果是

〖A〗1

〖B〗0

〖C〗35

〖D〗66

本题重点考查逻辑运算符和关系运算符的相关知识, 已知变量x为整型变量, 并赋值为35, 变量z为字符型变量, 并赋值为'B'。  
语句B = ((x) && (z < 'b')); 中, (x)的值为1, 'B'的asscii码小于'b'的asscii码, 所以(z < 'b')的值也为1, 1 && 1结果为1。  
因此A选项正确。

【19】与数学表达式x≥y≥z对应的C语言表达式是

〖A〗(x >= y) || (y >= x)

〖B〗(x >= y >= z)

〖C〗(x >= y) !(y >= z)

〖D〗(x >= y) && (y >= z)

本题考查关系表达式和逻辑表达式, y大于等于x同时y小于等于z, 因此x <= y和y <= z是逻辑与的关系。  
因此D选项正确。

【20】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | double x = 2.0, y; |
|  | if (x < 0.0) |
|  | y = 0.0; |
|  | else if ((x < 5.0) && ( !x)) |
|  | y = 1.0 / (x + 2.0); |
|  | else if (x < 10.0) |
|  | y = 1.0 / x; |
|  | else |
|  | y = 10.0; |
|  | printf(" %f\n ", y); |
|  | } |

程序运行后的输出结果是

〖A〗0.000000

〖B〗0.250000

〖C〗0.500000

〖D〗1.000000

本题重点考查if语句, 变量x和y均为double类型, 由于变量x赋值为2.0, 因此执行if((x < 5.0) && ( !x)), 即条件为0, 继续执行if(x < 10.0), 条件为1, 即执行y = 1.0 / x; 。  
依据printf()函数输出类型中f是以小数形式输出单、双精度实数。  
因此C选项正确。

【21】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int i, sum; |
|  | for (i = 1; i < 6; i++) |
|  | sum += i; |
|  | printf("%d\n", sum); |
|  | } |

程序运行后的输出结果是

〖A〗随机值

〖B〗0

〖C〗15

〖D〗16

本题重点考查for语句, 变量没有初始化的情况下, 默认值与编译器有关系, 不同的编译器有不同的值。  
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 = 3; |
|  | do { |
|  | printf("%d,", a -= 2); |
|  | } while ( !(--a)); |
|  | printf("\n"); |
|  | } |

程序运行后的输出结果是

〖A〗1

〖B〗1, -2,

〖C〗3,0,

〖D〗1,0

本题重点考查do……while语句, 该循环语句的特点是, 先执行循环中的语句, 然后再判断表达式是否为真, 如果为真则继续循环; 如果为假, 则终止循环。  
因此, do - while循环至少要执行一次循环语句。  
变量a赋初值为3, 每循环一次需执行语句a -= 2一次, 同时循环条件使变量a自减1, 所以第一次循环后变量a变为1, 第2次循环后变量a变为-2。  
因此B选项正确。

【23】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void fun(int a, int b, int c) { |
|  | a = b; |
|  | b = c; |
|  | c = a; |
|  | } |
|  | void main() { |
|  | int a = 10, b = 20, c = 30; |
|  | fun(a, b, c); |
|  | printf("%d,%d,%d\n", c, b, a); |
|  | } |

程序运行后的输出结果是

〖A〗10,20,30

〖B〗30,20,10

〖C〗20,30,10

〖D〗 0,0,0

函数调用中发生的数据传送是单向的。  
即只能把实参的值传送给形参, 而不能把形参的值反向地传送给实参。  
因此在函数调用过程中, 形参的值发生改变, 而实参中的值不会变化。  
因此B选项正确。

【24】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int c[6] = { 10, 20, 30, 40, 50, 60 }, \*p, \*s; |
|  | p = c; |
|  | s = &c[5]; |
|  | printf("%d\n", s - p); |
|  | } |

程序运行后的输出结果是

〖A〗5

〖B〗50

〖C〗6

〖D〗60

首先初始化一维数组c[6], 语句p = c; 指将c[0]元素的地址赋给指针变量p; 语句s = &c[5]; 指将c[5]元素的地址赋给指针变量s。  
程序最后输出s - p, 即结果为5。  
因此A选项正确。

【25】有以下程序

复制代码

C

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

程序运行后的输出结果是

〖A〗0

〖B〗2

〖C〗1

〖D〗3

该题首先初始化一维数组a[3]所有元素都为0; 执行嵌套循环for语句, 当i = 0、j = 0时, a[0] = a[0] + 1 = 1; 当i = 0、j = 1时, a[1] = a[0] + 1 = 2; 当i = 1、j = 0时, a[0] = a[1] + 1 = 3; 当i = 1、j = 1时, a[1] = a[1] + 1 = 3; 因此D选项正确。

【26】有以下程序

复制代码

C

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

程序运行后的输出结果是

〖A〗4 6

〖B〗4 4

〖C〗2 2

〖D〗2 4

首先通过p = a使p指向数组的第1个元素, 所以输出2; 在输出2以后, 由于p++, 即p就指向数组的第2个元素, \*\*k就是取出p所指向元素的值, 而p指向的是数组的第2个元素, 即输出4。  
因此D选项正确。

【27】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int aa[3][3] = { { 2 }, { 4 }, { 6 } }, i, \*p = &aa[0][0]; |
|  | for (i = 0; i < 2; i++) { |
|  | if (i == 0) |
|  | aa[i][i + 1] = \*p + 1; |
|  | else |
|  | ++p; |
|  | printf("%d ", \*p); |
|  | } |
|  | } |

程序运行后的输出结果是

〖A〗2 6

〖B〗2 3

〖C〗4 3

〖D〗4 6

观察题目, 可以发现, \*p = &aa[0][0]语句实际是将数组aa的首地址赋给了指针变量p, 将i的值带入for循环中, i = 0时, aa[0][1] = 3, \*p = 2; \*p输出的是指针所指向的数组值, 此时p所指向的是aa[0][0], 输出2, 而i = 1时执行了++p操作使指针向后移动指向了aa[0][1], 所以在输出\*p时应为3。  
因此B选项正确。

【28】以下fun函数返回数组中最大值的下标

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | int fun(int \*s, int n) { |
|  | int i, k; |
|  | for (i = 0, k = i; i < n; i++) |
|  | if (s[i] > s[k]) |
|  | \_\_\_\_\_\_\_\_\_\_; |
|  | return (k); |
|  | } |

在横线处应填入的内容是

〖A〗++k

〖B〗i = k

〖C〗k++

〖D〗k = i

fun函数的功能是返回数组中最大值的下标。  
通过for循环语句, 每次将最大的数给k, 即k = i。  
因此D选项正确。

【29】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("0ABCDEF")); |
|  | } |

程序运行后的输出结果是

〖A〗1

〖B〗6

〖C〗7

〖D〗0

本题考查字符串作为函数参数以及while语句, 本题中传入字符串0ABCDEF, 然后执行fun后, 就得到了该字符串中字符的个数, 不包括结尾符, 该字符串有7个字符, 所以答案为C。

【30】若有以下程序段

复制代码

C

|  |  |
| --- | --- |
|  | char str[4][12] = { "aa", "bbb", "ccccc", "d" }, \*strp[4]; |
|  | int i; |
|  | for (i = 0; i < 4; i++) |
|  | strp[i] = str[i]; |

不能正确引用字符串的选项是

〖A〗str[0]

〖B〗strp

〖C〗strp[3]

〖D〗\*strp

本题考查对字符串数组的引用。  
通过for循环将str数组每个元素的首地址赋给strp指针数组的每个元素。  
指针数组strp表示每一个数组元素都存放了一个字符串的首地址。  
选项A中, 引用的是字符串数组的第一个字符串; 选项B中, strp是指针的指针, 不能引用数组元素; C选项中str[3], 引用的是str中的第4个字符串; 选项D中, \*strp = strp[0]引用的是数组str中的第一个字符串。  
故本题答案为B选项。

【31】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | #include <string.h> |
|  | void fun(char \*s) { |
|  | char a[10]; |
|  | strcpy(a, "STRING"); |
|  | s = a; |
|  | } |
|  | void main() { |
|  | char \*p = "PROGRAM"; |
|  | fun(p); |
|  | printf("%s\n", p); |
|  | } |

程序运行后的输出结果是(此处□代表空格)

〖A〗STRING

〖B〗STRING□□□□

〖C〗STRING□□□

〖D〗PROGRAM

本题考查字符串指针作为函数参数, 本题中p作为字符串指针传入fun中, p指向的内容并没有发生变化, 所以选项D正确。

【32】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void fun(int n, int \*s) { |
|  | int f; |
|  | if (n == 1) |
|  | \*s = n + 1; |
|  | else { |
|  | fun(n - 1, &f); |
|  | \*s = f; |
|  | } |
|  | } |
|  | void main() { |
|  | int x = 0; |
|  | fun(4, &x); |
|  | printf("%d\n", x); |
|  | } |

程序运行后的输出结果是

〖A〗3

〖B〗1

〖C〗2

〖D〗4

本题考查函数的递归调用, 将函数参数带入函数中, 一步步递归即可得到结果, 结果为2, 选项C正确。

【33】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | int a = 1, b = 0; |
|  | printf("%d,", b = a + b); |
|  | printf("%d\n", a = 2 \* b); |
|  | } |

程序运行后的输出结果是()。

〖A〗1,2

〖B〗1,0

〖C〗3,2

〖D〗0,0

本题考查复合的赋值运算符, b = a + b 代表将a、b的值相加再赋给b, 此时b的值为1, a = 2 \* b代表将b的当前值1乘以2赋给a, 此时a的值为2, 程序输出1, 2 答案选A。

【34】有以下程序

复制代码

C

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

程序运行后的输出结果是

〖A〗34756

〖B〗5745

〖C〗5795

〖D〗35745

本题主要考查do…while语句, 执行fun函数后, a数组里面的元素变为35745, 输出后四个字符, 所以结果为5745, 选项B正确。

【35】有以下程序段

复制代码

C

|  |  |
| --- | --- |
|  | int \*p; |
|  | p = \_\_\_ malloc(sizeof(int)); |

若要求使p指向一个int 型的动态存储单元, 在横线处应填入的是

〖A〗(int \*)

〖B〗int

〖C〗int\*

〖D〗(\*int)

本题考查malloc函数, 题目中要求p指向一个int型的动态存储单元, 那么就应该将分配的存储单元转化为int, 所以选项A正确。

【36】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | #define SUB(X, Y) (X + 1) \* Y |
|  | void main() { |
|  | int a = 3, b = 4; |
|  | printf("%d\n", SUB(a++, b++)); |
|  | } |

程序运行后的输出结果是

〖A〗25

〖B〗20

〖C〗12

〖D〗16

本题考查宏定义, 宏定义只是做个简单的替换, 执行SUB(a++, b++) = (a++ + 1) \*b++ = 16, 选项D正确。

【37】有以下程序

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | char c[2][5] = { "6938", "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; j += 2) |
|  | s = 10 \* s + p[i][j] - '0'; |
|  | printf("%d\n", s); |
|  | } |

程序运行后的输出结果是

〖A〗6938

〖B〗9824

〖C〗4528

〖D〗6385

本题中首先是将二维字符数组c赋值指针数组p, 然后遍历p找到两个字符串中的偶数位置的字符并组成一个数值输出, 根据代码执行即可得到结果为6385。

【38】若有以下程序段

复制代码

C

|  |  |
| --- | --- |
|  | struct st { |
|  | int n; |
|  | struct st \*next; |
|  | }; |
|  | struct st a[3] = { 5, &a[1], 7, &a[2], 9, '\0' }, \*p; |
|  | p = &a[0]; |

则以下选项中值为6的表达式是

〖A〗p->n

〖B〗(\*p).n

〖C〗p->n++

〖D〗++(p->n)

本题考查结构体变量的引用, 其中a为定义的结构体数组, D选项中p->n为5, ++(p->n)为6, 所以D选项正确。

【39】有以下程序

复制代码

C

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

程序运行后的输出结果是

〖A〗1

〖B〗-1

〖C〗0

〖D〗7

本题考查位运算中按位异或运算符, 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, i ^ i运算符两边相同, 所以结果为0, 选项C正确。

【40】以下程序用来统计文件中字符的个数(函数feof用以检查文件是否结束, 结束时返回非零)

复制代码

C

|  |  |
| --- | --- |
|  | #include <stdio.h> |
|  | void main() { |
|  | FILE \*fp; |
|  | long num = 0; |
|  | fp = fopen("fname.dat", "r"); |
|  | while (\_\_\_\_\_\_\_\_\_\_) { |
|  | fgetc(fp); |
|  | num++; |
|  | } |
|  | printf("num=%d\n", num); |
|  | fclose(fp); |
|  | } |

下面选项中, 填入横线处不能得到正确结果的是

〖A〗feof(fp) == NULL

〖B〗!feof(fp)

〖C〗feof(fp)

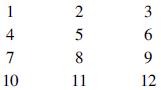
〖D〗feof(fp) == 0

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, while判断条件应是如果没有到达文件末尾, 所以选项C不能得到正确的结果。

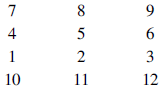
**编程题**

[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：将a所指4×3矩阵中第k行的元素与第0行元素交换。  
例如, 有下列矩阵：



若k为2, 程序执行结果为：



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) k |
|  | (2) N |
|  | (3) a[k][i] |

填空1:变量k在函数体fun中已经使用, 但在函数体中没有定义, 肯定是在函数的形参中定义的, 所以应填k。  
填空2:数组共N列, 所以应填N。  
填空3:这部分语句实现的功能是变量值的交换, 所以应填a[k][i]。

【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：读入一个字符串(长度<20), 将该字符串中的所有字符按ASCII码值升序排序后输出。  
例如, 若输入"edcba", 则应输出"abcde"。  
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) for (i = strlen(t) - 1; i; i--) |
|  | 或 for (i = strlen(t) - 1; i > 0; i--) |
|  | (2) if (t[j] > t[j + 1]) |

(1) 本题是利用选择法对数组元素进行比较排序。  
所谓选择法, 是依次用当前取得的元素和其后面的元素进行比较, 在第一个元素和其后面的元素顺次比较时, 可以借助中间变量来对两个数进行交换, 要保证第一个元素始终存放数组中的最大数, 以后依次挑选出次大数, 这样最终的数组就是有序的。  
strlen函数所求得的字符串长度, 数组最大下标为字符长度减1, 所以要减1。  
(2) 这里是一个分析逻辑错误, 题中要求按升序排序, 所以应改为if(t[j] > t[j + 1])。

【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fun, 它的功能是：将ss所指字符串中所有下标为奇数位置的字母转换为大写(若该位置上不是字母, 则不转换)。  
例如, 若输入"abc4Efg", 则应输出"aBc4EFg"。  
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 \*ss) { |
|  | int i; |
|  | for (i = 0; ss[i] != '\0'; i++) |
|  | /\*将ss所指字符串中所有下标为奇数位置的字母转换为大写\*/ |
|  | if (i % 2 == 1 && ss[i] >= 'a' && ss[i] <= 'z') |
|  | ss[i] = ss[i] - 32; |
|  | } |

该题要求将给定字符串中奇数位置的字母转换为大写, 需要先判断奇数位置以及是否是小写字母, 如果是再通过其转换方法进行转换。  
从C语言的学习中知道, 只要将小写字母减去32即转成大写字母, 将大写字母加上32即转成小写字母。  
本程序用if语句实现转换功能。