

# 程序员

## 2009年下半年试题

本试卷为：样式2

样式1：适用于模拟考试，所有答案在最后面。

样式2：适用于复习，每道题的题目和答案在一起。

本试卷由**跨步软考**提供

我们目前提供的免费服务有：

- 手机APP刷题
- 网页版刷题
- 真题pdf版下载
- 视频课程下载
- 其他资料下载

更多免费服务请访问我们的官网：<http://kuabu.xyz>

你也可以关注我们的微信公众号：**跨步软考**





如果您发现试题有错误，您可以通过以下方式联系我们

- 客服邮箱：service@kuabu.xyz
- 备用邮箱：kuabu@outlook.com
- 您也可以在微信公众号后台留言

本文档所有权归**跨步软考**(kuabu.xyz)，您可以传播甚至修改本文档，但是必须标明出自“**跨步软考 (kuabu.xyz)**”

## 上午综合试卷

第1题: 在Word中, 利用 (1) 功能按钮, 可以在画好的图形内填充颜色; 在单击 “绘图” 工具栏上的 “椭圆” 按钮后, 再按住 (2) 键进行绘制可确保绘出一个圆形

- A. 
- B. 
- C. 
- D. 

参考答案: D

解析:

Word 2003编辑中可以对字体、图形以及线条进行着色, 而这些着色的工具也是各不相同。A选项的功能是对选定的文字颜色进行设置, B选项的功能是更改线条的颜色, C选项的功能是对选定的文字进行突出显示, D选项的功能是对图像进行颜色的填充。这些按钮的后面都有一个下拉按钮, 其功能就是更改当前颜色。

第2题: 在Word中, 利用 (1) 功能按钮, 可以在画好的图形内填充颜色; 在单击 “绘图” 工具栏上的 “椭圆” 按钮后, 再按住 (2) 键进行绘制可确保绘出一个圆形

- A. Shift
- B. Ctrl
- C. Alt
- D. Tab

参考答案: A

解析:

使用Word中 “绘图” 工具栏上的各个工具按钮可以绘制多种图形。如果要绘制一个圆, 只能使用椭圆按钮, 但是这样绘制是很难绘制出一个圆的, 若要绘制圆, 应当在单击 “椭圆” 按钮后, 再按住Shift键。当绘制了多个图形对象时, 先绘制的将会被覆盖。可以对多个图形进行叠放层次的设置和组合, 对绘制的图形进行任意角度的旋转, 除了向左旋转90°、向右旋转90°和水平翻转、垂直翻转这些特殊的角度外, 还可以进行任意角度的旋转。如果要绘制一个正方形, 操作方法同上。

第3题: 在下图所示的工作表中, 若学生的最终成绩等于平时成绩的30%加上考试成绩的70%, 那么应在E2单元格中填写 (3), 并(4)拖动填充柄至E7单元格, 则可自动算出这些学生的最终成绩。

	A	B	C	D	E
1	学号	姓名	平时成绩	考试成绩	最终成绩
2	100201	张明明	89	91	
3	100202	李俊	78	75	
4	100203	王晓华	92	85	
5	100204	赵丽萍	85	80	
6	100205	黎明	70	72	
7	100206	任芳	82	69	

- A.  $C2*30\%, D2*70\%$
- B.  $C2*30\%:D2*70\%$

C. SUM(C2\*30%,D2\*70%)

D. SUM(C2\*70%:D2\*30%)

参考答案：C

解析：

在Excel中，公式必须以等号=开头，系统将=号后面的字符串识别为公式。计算求和的函数是SUM,根据题意，若要计算表中每个学生的最终成绩，那么可在E2单元格中填写“=SUM(C2\*30%, D2\*70%)”，如下图所示。

E2		=SUM(C2*30%, D2*70%)				
	A	B	C	D	E	F
1	学号	姓名	平时成绩	考试成绩	最终成绩	
2	100201	张明明	89	91	90.4	
3	100202	李俊	78	75		
4	100203	王晓华	92	85		
5	100204	赵丽萍	85	80		
6	100205	黎明	70	72		
7	100206	任芳	82	69		

第4题：在下图所示的工作表中，若学生的最终成绩等于平时成绩的30%加上考试成绩的70%，那么应在E2单元格中填写(3),并(4)拖动填充柄至E7单元格,则可自动算出这些学生的最终成绩。

	A	B	C	D	E
1	学号	姓名	平时成绩	考试成绩	最终成绩
2	100201	张明明	89	91	
3	100202	李俊	78	75	
4	100203	王晓华	92	85	
5	100204	赵丽萍	85	80	
6	100205	黎明	70	72	
7	100206	任芳	82	69	

A. 向水平方向

B. 向垂直方向

C. 按住Shift键向水平方向

D. 按住Shift键向垂直方向

参考答案：B

解析：

在Excel中有规律数据可以进行快速填充，灵活地使用自动填充功能可以避免重复输入数据。若用户需要对某个Excel工作表的E2: E7区域快速计算最终成绩，可以采用的方法是在E2单元格中填写“=SUM(C2\*30%, D2\*70%)”或“=SUM(C2\*30%:D2\*70%)”，并将鼠标移到E2单元格的右下角，并拖至E7,此时则可自动算出这些学生的最终成绩，如下图所示。

E2		=SUM(C2*30%, D2*70%)				
	A	B	C	D	E	F
1	学号	姓名	平时成绩	考试成绩	最终成绩	
2	100201	张明明	89	91	90.4	
3	100202	李 俊	78	75	75.9	
4	100203	王晓华	92	85	87.1	
5	100204	赵丽萍	85	80	81.5	
6	100205	黎 明	70	72	71.4	
7	100206	任 芳	82	69	72.9	

第5题：“http://www.rkb.gov.cn”中的gov代表的是（5）。

- A. 民间组织
- B. 商业机构
- C. 政府机构
- D. 高等院校

参考答案：C

解析：

因特网最高层域名分为机构性域名和地理性域名两大类。域名地址由字母或数字组成，中间以隔开，例如www.rkb.gov.cn其格式为：机器名.网络名.机构名.最高域名。Internet上的域名由域名系统DNS统一管理。域名被组织成具有多个字段的层次结构。最左边的字段表示单台计算机名，其他字段标识了拥有该域名的组；第二组表示网络名，如rkb；第三组表示机构性质，例如.gov是政府部门；而最后一个字段被规定为表示组织或国家，称为顶级域名，常见的国家或地区域名如下表所示。

表 常见的国家或地区域名

域 名	国家/地区	域 名	国家/地区
.cn	China（中国）	.gb	Great Britain（英国）
.au	Australia（澳大利亚）	.hk	HongKang（中国香港）
.ca	Canada（加拿大）	.kr	Korea-south（韩国）
.jp	Japan（日本）	.ru	Russian（俄罗斯）
.de	Germany（德国）	.it	Italy（意大利）
.fr	France（法国）	.tw	Taiwan（中国台湾）

常见的机构性域名如下表所示。

表 常见的机构性域名

域 名	机 构 性 质	域 名	机 构 性 质
.com	工、商、金融等企业	.rec	消遣机构
.net	互联网络、接入网络服务机构	.org	各种非盈利性的组织
.gov	政府部门	.edu	教育机构
.arts	艺术机构	.mil	军事机构
.info	提供信息服务的企业	.firm	商业公司
.store	商业销售机构	.nom	个人或个体

第6题：逻辑变量X、Y进行逻辑“异或”（用?表示）运算的含义是：若X、Y取值相同（都为true或都为false），则X?Y的值为false，否则X?Y的值为true。用逻辑“与”（^）、“或”（v）、“非”（~）表示X?Y的式子为（6）

- A.  $(X \wedge Y) \wedge (X \wedge \bar{Y})$

- B.  $(X \vee Y) \wedge (\bar{X} \vee Y)$
- C.  $(X \wedge \bar{Y}) \vee (\bar{X} \wedge Y)$
- D.  $(X \vee Y) \vee (\bar{X} \vee Y)$

参考答案：C

解析：

本题考查逻辑运算基础知识。

X和Y逻辑“与”运算的含义是：X和Y的值都为true，则 $X \wedge Y$ 为true，否则为false。

X和Y逻辑“或”运算的含义是：X和Y的值都为false，则 $X \vee Y$ 为false，否则为true。

X和Y逻辑“异或”运算的含义是：X和Y的值都为true或false，则 $X \oplus Y$ 为false，否则为true。

用下表检验题中各个选项，可得：

X	Y	A	B	C	D
		$(X \wedge Y) \wedge (X \wedge \bar{Y})$	$(X \vee Y) \wedge (\bar{X} \vee Y)$	$(X \wedge \bar{Y}) \vee (\bar{X} \wedge Y)$	$(X \vee Y) \vee (\bar{X} \vee Y)$
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	0	1	1
1	1	0	1	0	1

显然，与逻辑“异或”运算相符合的逻辑式为  $(X \wedge \bar{Y}) \vee (\bar{X} \wedge Y)$ 。

第7题：以下关于计算机中数据表示的叙述中，错误的是（7）。

- A. 计算机中的数值数据采用二进制表示，非数值性数据不使用二进制表示
- B. 正整数的原码和补码表示形式相同，而负整数的原码和补码表示形式不同
- C. 数值中的小数点在硬件中不明确表示，而是采用约定位置的方式
- D. 码长相同时，补码比原码可以多表示一个数

参考答案：A

解析：

本题考查数据表示方面的基础知识。

计算机语言是二进制，因此在计算机中存储和运算的所有数据（数值数据和非数值数据），无论采用哪一种编码方式，最终都会以二进制形式加以表示。因此，选项A错误。

第8题：某机器的浮点数格式如下，采用非规格化表示（即不要求尾数的绝对值大于等于0.5）。



若阶码和尾数都采用补码表示，则该格式浮点数的绝对值范围为（8）。

- A.  $2^{-6} \sim 2^6$
- B.  $2^{-63} \sim 2^{63}$
- C.  $2^{-64} \sim 2^{64}$



D.  $2^{-64} \sim 2^{63}$

参考答案 : D

解析 :

本题考查数据表示方面的基础知识。

浮点数的表示方式为  $W \times 2^M$ , 其中,  $W$  称为尾数 (包含数符),  $M$  称为阶 (包括阶符)。

阶码长度为 6 位, 补码表示方式下其值区间为  $[-2^6, 2^6 - 1]$ , 因此  $M$  的取值范围为  $-64 \sim 63$ 。尾数和数符共 9 位, 补码表示方式下其值区间为  $[-1, 1 - 2^{-8}]$ , 因此上述格式浮点数的绝对值范围为  $2^{-64} \sim 2^{63}$ 。

第9题 : 关于汉字编码的叙述, 错误的是 ( 9 )。

- A. 采用矢量法表示汉字时, 若两个汉字的笔画和字形不同, 则它们的矢量编码一定不同
- B. 采用点阵法表示汉字时, 若两个汉字的笔画和字形不同, 则它们的点阵信息量一定不同
- C. 汉字的输入、存储和输出采用不同的编码, 拼音码属于输入码
- D. 汉字在计算机内存储时, 其编码长度不能少于2个字节

参考答案 : B

解析 :

本题考查数据表示方面的基础知识。

汉字种类繁多, 编码比拼音文字困难, 而且在一个汉字处理系统中, 输入、内部处理、存储和输出对汉字代码的要求不尽相同, 所以采用的编码也不同。汉字的输入码主要分为三类: 数字编码、拼音码和字形码。

用点阵表示字形时, 若点阵大小确定, 对于不同汉字, 其点阵信息量是相同的。

汉字的矢量表示法是将汉字看作是由笔画组成的图形, 提取每个笔画的坐标值, 这些坐标值就可以决定每一笔画的位置, 将每一个汉字的所有坐标值信息组合起来就是该汉字字形的矢量信息。显然, 汉字的字形不同, 其矢量信息也就不同, 每个汉字都有自己的矢量信息。

第10题 : 在微型计算机中, 通常用主频来描述CPU的 ( 10 ): 对计算机磁盘工作影响最小的因素是 ( 11 )

- A. 运算速度
- B. 可靠性
- C. 可维护性
- D. 可扩充性

参考答案 : A

解析 :

主频是CPU的时钟频率, 简单地说也就是CPU的工作频率。一般来说, 一个时钟周期完成的指令数是固定的, 所以主频越高, CPU的速度也就越快, 故常用主频来描述CPU的运算速度。外频是系统总线的工作频率。倍频是指CPU外频与主频相差的倍数。主频=外频X倍频。

第11题 : 在微型计算机中, 通常用主频来描述CPU的 ( 10 ): 对计算机磁盘工作影响最小的因素是 ( 11 )

- A. 温度

- B. 湿度
- C. 噪声
- D. 磁场

参考答案：C

解析：

使用硬盘时应注意防高温、防潮和防电磁干扰。硬盘工作时会产生一定热量，使用中存在散热问题。温度以20?25℃为宜，温度过高或过低都会使晶体振荡器的时钟主频发生改变。温度还会造成硬盘电路元件失灵，磁介质也会因热胀效应而造成记录错误。温度过低，空气中的水分会凝结在集成电路元件上，造成短路；湿度过高，电子元件表面可能会吸附一层水膜，氧化、腐蚀电子线路，以致接触不良，甚至短路，还会使磁介质的磁力发生变化，造成数据的读写错误。湿度过低，容易积累大量因机器转动而产生的静电荷，这些静电会烧坏CMOS电路，吸附灰尘而损坏磁头、划伤磁盘片。机房内的湿度以45%?65%为宜。注意使空气保持干燥或经常给系统加电，靠自身发热将机内水汽蒸发掉。另外，尽量不要使硬盘靠近强磁场，如音箱、喇叭、电机、电台和手机等，以免硬盘所记录的数据因磁化而损坏。

第12题：MIDI数据与数字化波形声音数据（12）。

- A. 相同
- B. 不同
- C. 相近
- D. 格式一致

参考答案：B

解析：

波形声音是一个用来表示声音强弱的数据序列，它是由模拟声音经采样、量化和编码后得到的便于计算机存储和处理的数据格式。声音信号数字化后，其数据传输率（每秒位数）与信号在计算机中的实时传输有直接关系，而其总数据量又与计算机的存储空间有直接关系。数字波形声音数据量非常大，因此在编码的时候常常要采用压缩的方式来压缩数字数据以减少存储空间和提高传输效率（降低传输带宽）。而MIDI数据不是单个采样点的编码（波形编码），而是乐谱的数字描述，称为MIDI消息。乐谱由音符序列、定时、音色和音量等组成，每个消息对应一个音乐事件（如键压下、键释放等），一组MIDI消息送到MIDI音源时，音源即合成出相应的音乐。所以，MIDI数据与数字化波形声音数据不同。

第13题：话筒是向计算机提供（13）的设备。

- A. 音频数字信号
- B. 音频模拟信号
- C. 采样信号
- D. 音频数字信号和采样信号

参考答案：B

解析：

声音是一种模拟信号，计算机要对它进行处理，必须将它转换为数字声音信号，即用二进制数字的编码形式来表示声音。话筒是向计算机提供声音信号的设备，作用是将自然声音信号转换为电信号（模拟声音信号），然后送到音频卡（声卡），将话筒输入的声音信息进行模数转换（A/D）、压缩等处理。

第14题: 跨步软考 (kuabu.xyz) 没有找到题目!

参考答案:

## 下午案例分析

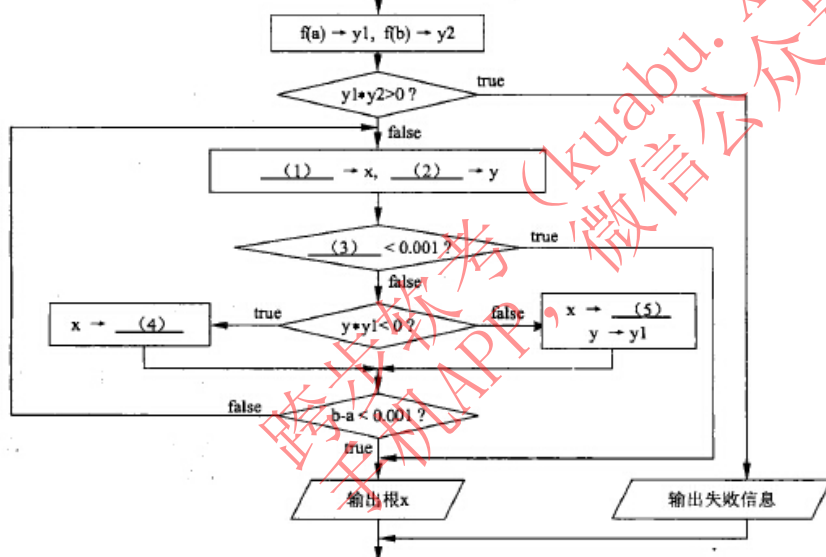
第1题: 【说明】

求连续函数 $f(x)$ 的根 (方程 $f(x)=0$ 的解) 的最简单方法是二分法。为此, 首先需要在若干点上检查函数值的符号, 如果发现 $f(a)$ 与 $f(b)$ 符号相反 ( $a < b$ ), 则在区间  $(a, b)$  中必然存在 $f(x)$ 的根。因为当 $x$ 从 $a$ 变到 $b$ 时, 连续函数的值将从正变到负 (或从负变到正), 必然要经过0。区间  $(a, b)$  就是根的初始范围。

取该区间的中点 $m$ , 如果 $f(m)=0$ , 则根就是 $m$ 。如果 $f(a)$ 与 $f(m)$ 符号相反, 则根一定在区间 $(a, m)$ 中; 如果 $f(m)$ 与 $f(b)$ 符号相反, 则根一定在区间 $(m, b)$ 中。因此, 根的范围缩小了一半。

依此类推, 将区间一半一半地分下去, 当区间的长度很小 (达到根的精度要求, 例如0.001)时, 或者当区间中点处的函数值几乎接近于0 (即绝对值小于预先规定的微小量, 例如0.001)时, 近似计算就可以结束了。

以下流程图描述了用二分法近似计算区间  $(a, b)$  中 $f(x)$ 的根的过程。问题: 1.1 【流程图】



跨步软考[www.kuabu.xyz]答案解析:

(1)  $(a+b)/2$

(2)  $f(x)$ , 或  $f((a+b)/2)$

(3)  $|y|$ , 或  $abs(y)$ , 其中  $y$  可由  $f(x)$  或  $f((a+b)/2)$  代替

(4)  $b$

本题描述了求函数根 (0点) 的二分法, 题中还详细说明了二分法的原理。

假设 $a$ 和 $b$ 是区间两端点值的变量。流程图中, 一开始就将函数两端的值分别送 $y1$ 和 $y2$ , 接着判断 $y1$ 与 $y2$ 符号是否相反 (同号时该算法不能往下进行)。若相反, 则应将 $a$ 与 $b$ 的中点值 $(a+b)/2$ 送 $x$ 。此时的函数值 $f(x)$ , 即 $f((a+b)/2)$ 应送 $y$ 。因此, (1)处应填 $(a+b)/2$ , (2)处可填 $f(x)$ 或 $f((a+b)/2)$ 。

接着需要判断新的函数值是否已经接近0, 因此, (3)处应填 $|y|$ 或 $abs(y)$ 。

如果这个新函数值已经接近0, 则可以直接输出变量 $x$ 的值 (刚取的中点值) 作为函数的近似根; 如果该函数值尚未接近0, 则需要将该区间进行二分, 即需要判断选用左半区间还是右半区间, 继续进行迭代计算。

如果 $y * y1 < 0$ , 则说明新的函数值与原区间的左端函数值符号相反, 因此应取左半区间; 否则应取右半区间。

若取左半区间, 则原来的区间左端点 $a$ 没有变化, 左端点的函数值 $y1$ 也没有变化, 只要将中点值 $x$ 送右端点变量 $b$ 就可以。因此, (4)处填 $b$ 。



若取右半区间, 则区间的右端点没有变化, 右端点的函数值 $y_2$ 也没有变化, 这时需要将中点值 $x$ 送左端点变量 $a$ 。因此, (5)处应填 $a$ 。由于每次迭代都需要判断 $y \cdot y_1$ 的符号, 因此 $y_1$ 的改变将影响下次迭代。因此, 此时还需要将中点处的函数值 $y$ 送 $y_1$ , 作为新区间的左端点函数值。

当新的区间 $(a, b)$ 长度 $ha$ 很小时, 迭代计算就可以结束, 输出已经得到的近似根 $x$ 就可以了。

## 第2题: 【说明1】

函数Counter(int n, int w[])的功能是计算整数 $n$ 的二进制表示形式中1的个数, 同时用数组 $w$ 记录该二进制数中1所在位置的权。

例如, 十进制数22的二进制表示为10110。对于该二进制数, 1的个数为3, 在 $w[0]$ 中存入2 (即 $2^1$ )、 $w[1]$ 中存入4 (即 $2^2$ )、 $w[2]$ 中存入16 (即 $2^4$ )。

## 【说明2】

函数SmOve(int A[], int n)的功能是将数组中所有的奇数都放到所有偶数之前。其过程为: 设置数组元素下标索引 $i$  (初值为0)和 $j$  (初值为 $n-1$ ), 从数组的两端开始检查元素的奇偶性。若 $A[i]$ 、 $A[j]$ 都是奇数, 则从前往后找出一个偶数, 再与 $A[j]$ 进行交换; 若 $A[i]$ 、 $A[j]$ 都是偶数, 则从后往前找出一个奇数, 再与 $A[i]$ 进行交换; 若 $A[i]$ 是偶数而 $A[j]$ 是奇数, 则交换两者, 直到将所有的奇数都排在所有偶数之前为止。

## 【C函数1】

问题: 2.1

```
int Counter(int n, int w[])
{
    int i = 0, k = 1;
    while ( (1) ) {
        if (n % 2) w[i++] = k;
        n = n / 2; (2);
    }
    return i;
}
```

## 【C函数2】

```
void Smove(int A[], int n)
{
    int temp, i = 0, j = n-1;
    if (n < 2) return;
    while (i < j) {
        if (A[i] % 2 == 1 && A[j] % 2 == 1) { (3); }
        else if (A[i] % 2 == 0 && A[j] % 2 == 0) { (4); }
        else {
            if ( (5) ) {
                temp = A[i]; A[i] = A[j]; A[j] = temp;
            }
            i++, j--;
        }
    }
}
```

跨步软考[www.kuabu.xyz]答案解析:

(1) $n!=0$ , 或其等价形式

(2) $k=k*2$ , 或 $k*=2$ , 或 $k+=k$ , 或 $k=k+k$ , 或其等价形式

(3) $i++$ , 或 $++i$ , 或 $i+=1$ , 或 $i=i+1$ , 或其等价形式

(4) $j--$ , 或 $--j$ , 或 $j-=1$ , 或 $j=j-1$ , 或其等价形式

(5) $A[i]\%2=0\&\&A[j]\%2!=1$ , 或 $A[i]\%2!=0\&\&A[j]\%2$ , 或 $!(A[i]\%2)\&\&A[j]\%2$ , 或其等价形式

本题考查c程序控制结构的应用。

根据题目描述, 函数Counter(int n, int w[])的功能是计算整数 $n$ 的二进制表示形式中1的个数, 同时用数组 $w$ 记录该二进制数中1所在位置的权。要计算 $n$ 的二进制形式中1的个数, 基本方法是计算 $n$ 的二进制数时进行计数,

题中的函数Counter采用的就是此方法。十进制数转换为二进制数的方法是辗转除以2取余数, 直到被除数为0时为止。

例如, 十进制数26的二进制形式为11010, 计算步骤为:

①被除数是26, 除数是2, 商为13, 余数为0;

②被除数是13, 除数是2, 商为6, 余数是1;

③被除数是6, 除数是2, 商为3, 余数是0;

④被除数是3, 除数是2, 商为1, 余数是1;

⑤被除数是1, 除数是2, 商为0, 余数是1。

计算结束。

从上例可知, 二进制数的各位是从低到高依次计算出来的。

因此, 函数Counter中的空(1)处应填入 $n$ 或 $n!=0$ 。显然, 代码中用 $k$ 记录二进制数各位的权值, 其初始值为1, 每算出1位, 其值要改为上一位权值的2倍。因此空(2)处应填入 $k=k*2$ 或 $k*=2$ 。

对于函数Smove(int A[], int n), 其功能是将数组中所有的奇数都放到所有偶数之前。分析题中给出的处理过程, 交换动作发生在 $A[i]$ 为偶数而 $A[j]$ 为奇数时, 因此空(5)处应填入“ $A[i]\%2=0\&\&A[j]\%2!=0$ ”(或其等价表示方式), 其他情况下都是修改 $i$ 和(或) $j$ 的值。下面举一例进行说明, 设数组A中的元素初始排列为15, 20, 23, 30, 77, 62, 45, 68, 34, 61。

初始时 $i=0, j=n-1=9$ 。 $A[0]=15, A[9]=61$ ,  $A[0]$ 是奇数, 所以其值保持不动,  $A[9]$ 也是奇数, 需要将其前移, 实质上是找出前面的一个偶数与其对调, 这就需要增加下标 $i$ 的值, 从而向后找偶数。对应的代码是 $i++, j$ 不变, 因此空(3)处应填入 $i++$ 或等价表示方式。接下来找到的偶数是 $A[1]$ , 将其与 $A[9]$ 对调, 数组中的元素排列情况为15, 61, 23, 30, 77, 62, 45, 68, 34, 20。之后 $i++, j--$ , 对应的数组元素为 $A[2]$ (值为23, 是奇数)和 $A[8]$ (值为34, 是偶数), 这种情况下显然不需要进行对调, 因此将 $i$ 增1、 $j$ 减1, 找到 $A[3]$ 和 $A[7]$ 。由于 $A[3]$ (值为30)和 $A[7]$ (值为68)都是偶数, 因此需要从后面找出一个奇数, 将其与 $A[3]$ 对调。因此, 空(4)处应填入 $j--$ (或等价表示方式)。接下来找到 $A[6]$ , 交换 $A[3]$ 和 $A[6]$ 的值, 数组元素的排列情况为15, 61, 23, 45, 77, 62, 30, 68, 34, 20。这样, 数组中的奇数全部排在偶数之前。

### 第3题: 【说明1】

函数test\_f1(int m, int n)对整数m、n进行某种运算后返回一个整数值。

【C函数1】

```
int test_f1(int m, int n)
{
    int k;
    k = m > n ? m : n;
    for (; (k%m!=0) || (k%n!=0); k++);
    return k;
}
```

问题: 3.1 (1)请写出发生函数调用test\_f1(9,6)时, 函数的返回值;

(2)请说明函数test\_f1的功能。问题: 3.2 请写出函数test\_f2()的运行结果。问题: 3.3 函数test\_f3()对返回值的处理有缺陷, 请指出该缺陷并说明修改方法。

跨步软考[www.kuabu.xyz]答案解析:

(1)18

(2)求两个整数的最小公倍数

考生解答含义与最小公倍数相同即可给分。

本题考查C语言基本运算和控制结构的应用。

对于传入函数test\_f1中的参数m和n, 运算“ $k=m>n?m:n$ ”使得k取m和n中的较大者。此后, “for;(k%m!=0)||(k%n!=0);k++;”使得k最后的取值正好能同时被m和n整除, 因此, 本函数的功能是求m和n的最小公倍数。9和6的最小公倍数是18。

跨步软考[www.kuabu.xyz]答案解析:

9 4

1 4

本题考查C程序中的数据定义和存储应用。

对于定义char str[]="NewWorld",数组str由字符串"NewWorld"初始化后,其大小为9 (包括串为结束标志字符" ' '),因此sizeof(str)的值为9。

对于定义"char\*p=str",ptr指向"NewWorld"的第一个字符"N" (即数组str的起始地址对应存储单元处存放的字符),ptr是指针变量,因此sizeof(p)的值为4。

对于定义"char i=" ",i是字符变量,其初始值为ASCII码值为0的字符,因此sizeof(i)的值为1。

对于定义"void \*ptr=malloc(50)",ptr是指向50个字节存储区域的指针,因此sizeof(ptr)的值为4。

跨步软考[www.kuabu.xyz]答案解析:

缺陷:返回了局部数据(或变量,或数组)的指针(或地址)。

修改方法:用malloc函数申请存储字符串的存储空间,令指针tstr指向该存储空间并返回tstr的值即可。

本题考查C语言数据存储类型的基础知识及常见应用错误和处理知识。

一般将内存划分为代码区、静态数据区、栈区和堆区,其中,栈区和堆区也称为动态数据区。C程序中的数据(大多以变量和数组表示)可存储在计算机系统内存区域的不同部分。全局变量(全局数组、外部变量等)和静态局部变量的存储单元在静态数据区分配,它们的生存期(即变量与其存储单元之间的绑定关系)较长,在程序运行期间始终存在。而局部变量(自动变量、局部数组等)的存储单元在栈区分配,它们的生存期较短,随着函数被调用而分配空间,函数调用结束后释放分配给它们的存储空间,这是由系统控制的。而堆区的存储空间由程序员根据需要用malloc函数申请,不用时用free函数释放,归还给系统。

在函数test\_f3中,数组tstr的空间在栈区分配,函数调用结束后该存储空间将由系统自动回收,随着程序的执行再进行分配,所以用"return tstr"返回该数组的首地址将可能使得需要用这个存储区中数据的地方出错。

第4题:【说明】

函数del\_substr(S,T)的功能是从头至尾扫描字符串S,删除其中与字符串T相同的所有子串,其处理过程为:首先从串S的第一个字符开始查找子串T,若找到,则将后面的字符向前移动将子串T覆盖掉,然后继续查找子串T;否则从串S的第二个字符开始查找,依此类推,重复该过程,直到串S的结尾为止。该函数中字符串的存储类型SString定义如下:

```
typedef struct {  
    char *ch;                /*串空间的首地址*/  
    int length;              /*串长*/  
}SString;
```

问题:4.1

## 【C 函数】

```
void del_substr(SString *S, SString T)
{
    int i, j;
    if ( S->length < 1 || T.length < 1 || S->length < T.length )
        return;
    i = 0;                                /* i 为串 S 中字符的下标 */
    for ( ; ; ) {
        j = 0;                            /* j 为串 T 中字符的下标 */
        while ( i < S->length && j < T.length ) {
            /* 在串 S 中查找与 T 相同的子串 */
            if ( S->ch[i]==T.ch[j] ) {
                i++; j++;
            }
            else {
                i = (1); j = 0;           /* i 值回退, 为继续查找 T 做准备 */
            }
        }
        if ( (2) ) {                      /* 在 S 中找到与 T 相同的子串 */
            i = (3);                      /* 计算 S 中子串 T 的起始下标 */
            for(k = i+T.length; k<S->length; k++) /* 通过覆盖子串 T 进行删除 */
                S->ch[(4)] = S->ch[k];
            S->length = (5);               /* 更新 S 的长度 */
        }
        else break;                      /* 串 S 中不存在子串 T */
    }
}
```

跨步软考[www.kuabu.xyz]答案解析:

- (1)i-j+1,或其等价形式
- (2) $j=T.length$ ,或 $j>=T.length$ , 或其等价形式
- (3)i-j,或 $i-T.length$
- (4) $k-T.length$ ,或 $k-j$ ,或其等价形式
- (5) $S->length -T.length$ , 或 $S->length-j$

本题考查c程序设计基本能力, 包括指针、结构体类型的应用。

根据题目部分的描述, 首先要在字符串S中查找与T相同的子串, 基本方法是逐个字符进行比对, 其代码如下所示:

```
j = 0; /* j 为串 T 中字符的下标 */
while ( i < S->length && j < T.length ) { /* 在串 S 中查找与 T 相同的子串 */
    if ( S->ch[i]==T.ch[j] ) {
        i++; j++;
    }
    else {
        i = (1); j = 0; /* i 值回退, 为继续查找 T 做准备 */
    }
}
```

显然, 当S->ch[i]与T.ch[j]不相等时, i值需回退至本趟开始位置后再后移一个位置, 从而为开始下一趟查找做准备。由于相同的字符个数为j个, 因此i需回退至位置i-j+1, 即空(1)处填入i-j+1。

由于在串S中查找与T相同的子串是以“i<S->length&&j<T.length”作为条件的, 因此查找结束时, 该条件必然为假。若在S中找到与T相同的子串, 则“j<T.length”必然不成立, 即将等于T.length。因此, 空(2)处应填入j==T.length。根据注释, 空(3)处需计算S中子串T的起始下标, 显然应填入i-j或i-T.length。通过覆盖子串T进行删除, 也就是将后面的字符前移来实现删除。假设需要删除的子串在S->ch[p]?S->ch[p+T.length-1]中, 则需要将S->ch[p+T.length]移至S->ch[p], S->ch[p+T.length+1]移至S->ch[p+1], 依此类推, 直到将S->ch[S->length-1]移至S->ch[S->length-T.length]为止, 因此以下代码中的空(4)处应填入k-T.length。

for(k=i+T.length;k<s->length;k++)/\*通过搜盖子串T进行删除\*/S->ch[(4)]=S->ch[k];

最后, 空(5)处很明显是要将S的长度减去子串T的长度值, 因为删除了一个相应长度值的子串, 因此空(5)处填入S->length-T.length。

#### 第5题: 【说明】

已知类LinkedList表示列表类, 该类具有四个方法:addElement()、lastElement()、number-OfElement()以及removeLastElement()。四个方法的含义分别为:

void addElement(Object): 在列表尾部添加一个对象;

Object lastElement(): 返回列表尾部对象;

int number-OfElement(): 返回列表中对象个数;

void removeLastElement(): 删除列表尾部的对象。

现需要借助LinkedList来实现一个Stack栈类, C++代码1和C++代码2分别采用继承和组合的方式实现。

#### 【C++代码1】

```
class Stack :public LinkedList{
public:
    void push(Object o){ addElement(o); } //压栈
    Object peek(){ return (1); } //获取栈顶元素
    bool isEmpty(){ //判断栈是否为空
        return number-OfElement() == 0;
    };
    Object pop(){ //弹栈
        Object o = lastElement();
        (2);
        return o;
    };
};
```

问题: 5.1



## 【C++代码2】

```
class Stack {  
private:  
    ____ (3) ____;  
public:  
    void push(Object o){    //压栈  
        list.addElement(o);  
    };  
    Object peek(){          //获取栈顶元素  
        return list.____ (4) ____;  
    };  
    bool isEmpty(){         //判断栈是否为空  
        return list.numberOfElement() == 0;  
    };  
    Object pop(){           //弹栈  
        Object o = list.lastElement();  
        list.removeLastElement();  
        return o;  
    };  
};
```

若类LinkedList新增加了一个公有的方法removeElement(int index),用于删除列表中第index个元素,则在用继承和组合两种实现栈类Stack的方式中,哪种方式下Stack对象可访问方法removeElement(int index)? (5) (A.继承B.组合)

跨步软考[www.kuabu.xyz]答案解析:

- (1) lastElement()
- (2) removeLastElement()
- (3) LinkedList list
- (4) lastElement()
- (5) A

本题考查面向对象基本概念和C++语言程序设计能力。

继承和组合是面向对象程序设计中复用的基本手段。继承的方式是在实现类的过程中指定其父类,则该类将自动拥有父类的属性和方法;组合方式则是将某类的对象作为该类的成员,通过成员使用该类的属性和方法。

本题中C++代码1采用的是继承方式,因此在Stack类中可以直接使用LinkedList类中定义的方法。空缺(1)要求获得栈顶元素,LinkedList类提供了lastElement()方法获得列表尾部的元素,因此可使用该方法获得栈顶元素;空缺(2)位于Stack类的pop()方法中,该方法不仅要获得栈顶元素,还要将栈顶元素删除,因此可以直接复用LinkedList类中的removeLastElement()方法。

本题中C++代码2采用的是组合方式,因此在Stack类中需要通过LinkedList对象来使用其属性和方法。Stack类中的peek()方法要求获得栈顶元素,list一定是LinkedList类型的对象,因此空缺(3)应该定义link成员变量,空缺(4)应该调用link对象获得队列的尾部元素方法。

若LinkedList方法新增加了一个公有的removeElement(int index)方法,采用继承方式的子类对象可直接使用该方法,而采用组合方式则必须通过对象形式进行访问,因此必须在Stack类中增加新的方法,并在该方法中通过list对象调用removeElement(int index)。

第6题: 【说明】

已知类LinkedList表示列表类, 该类具有四个方法:addElement()、lastElement()、numberOfElement()以及removeLastElement()。四个方法的含义分别为:

void addElement(Object): 在列表尾部添加一个对象;

Object lastElement(): 返回列表尾部对象;

int numberOfElement(): 返回列表中对象个数;

void removeLastElement(): 删除列表尾部的对象。

现需要借助LinkedList来实现一个Stack栈类, Java代码1和Java代码2分别采用继承和组合的方式实现。

【Java 代码 1】

```
public class Stack extends LinkedList{
    public void push(Object o){ //压栈
        addElement(o);
    }
    public Object peek(){ //获取栈顶元素
        return ____ (1) ____;
    }
    public boolean isEmpty(){ //判断栈是否为空
        return numberOfElement() == 0;
    }
    public Object pop(){ //弹栈
        Object o = lastElement();
        ____ (2) ____;
    }
}
```

问题: 6.1

```
return o;
```

```
}
```

```
}
```

## 【Java 代码 2】

```
public class Stack {  
    private ____ (3) ____;  
    public Stack() {  
        list = new LinkedList();  
    }  
    public void push(Object o) {  
        list.addElement(o);  
    }  
    public Object peek() {          //获取栈顶元素  
        return list.____ (4) ____;  
    }  
    public boolean isEmpty() {      //判断栈是否为空  
        return list.numberOfElement() == 0;  
    }  
    public Object pop() {           //弹栈  
        Object o = list.lastElement();  
        list.removeLastElement();  
        return o;  
    }  
}
```

若类LinkedList新增加了一个公有的方法removeElement(int index),用于删除列表中第index个元素,则在用继承和组合两种实现栈类Stack的方式中,哪种方式下Stack对象可访问方法removeElement(int index)? (5) (A.继承B.组合)

跨步软考[www.kuabu.xyz]答案解析:

- (1)lastElement()
- (2)removeLastElement()
- (3)LinkedList list
- (4)lastElement()
- (5)A

本题考查面向对象基本概念和Java语言程序设计能力。

继承和组合是面向对象程序设计中复用的基本手段。继承的方式是在实现类的过程中指定其父类,则该类将自动拥有父类的属性和方法;组合方式则是将某类的对象作为该类的成员,通过成员使用该类的属性和方法。

本题中Java代码1采用的是继承方式,因此在Stack类中可以直接使用LinkedList类中定义的方法。空缺(1)要求获得栈顶元素,LinkedList类提供了lastElement()方法获得列表尾部的元素,因此可使用该方法获得栈顶元素;空缺(2)位于Stack类的pop()方法中,该方法不仅要获得栈顶元素,还要将栈顶元素删除,因此可以直接复用LinkedList类中的removeLastElement()方法。

本题中Java代码2采用的是组合方式,因此在Stack类中需要通过LinkedList对象来使用其属性和方法。Stack类中的peek()方法要求获得栈顶元素,list一定是LinkedList类型的对象,因此空缺(3)应该定义link成员变量,空缺(4)应该调用link对象获得队列的尾部元素方法。

若LinkedList方法新增加了一个公有的removeElement(int index)方法,采用继承方式的子类对象可直接使用该方法,而采用组合方式则必须通过对象形式进行访问,因此必须在Stack类中增加新的方法,并在该方法中通过list对象调用removeElement(int index)。