

全国计算机技术与软件专业技术资格（水平）考试

2007 年上半年 程序员 下午试卷

（考试时间 14:00～16:30 共 150 分钟）

请按下述要求正确填写答题纸

1. 在答题纸的指定位置填写你所在的省、自治区、直辖市、计划单列市的名称。
2. 在答题纸的指定位置填写准考证号、出生年月日和姓名。
3. 答题纸上除填写上述内容外只能写解答。
4. 本试卷共 8 道题，试题一至试题三是必答题，试题四至试题五选答 1 道，试题六至试题八选答 1 道。每题 15 分，满分 75 分。

试题号	一～三	四～五	六～八
选择方法	必答题	选答 1 题	选答 1 题

5. 解答时字迹务必清楚，字迹不清时，将不评分。
6. 仿照下面例题，将解答写在答题纸的对应栏内。

例题

2007 年上半年全国计算机技术与软件专业技术资格（水平）考试日期是(1)月(2)日。

因为正确的解答是“5 月 26 日”，故在答题纸的对应栏内写上“5”和“26”（参看下表）。

例题	解答栏
(1)	5
(2)	26

试题一（共 15 分）

阅读以下说明和流程图，回答问题 1 和问题 2，将解答填入答题纸的对应栏内。

[说明]

为便于管理，每本正式出版的图书都印有国际标准书号。标准书号由“ISBN”和 10 个数字组成，其格式为：ISBN 组号-出版者号-书名号-校验码。其中，校验码是根据前面 9 个数字计算得到的，用于计算机自动校验。假设标准书号的 10 个数字依次是 $a(1), a(2), \dots, a(10)$ ，则校验码 $a(10)$ 的设置应使 $S=10*a(1)+9*a(2)+8*a(3)+\dots+1*a(10)$ 能被 11 整除。如果校验码 $a(10)$ 应设置成 10，则规定以“X”表示之。例如，信息处理技术人员考试大纲的标准书号为：ISBN 7-302-11191-X。第 1 段上的数字“7”是国际 ISBN 中心分配给中国 ISBN 中心管理的组号。第 2 段上的“302”表示清华大学出版社。标准书号的校验过程如图 1-1 所示，计算校验码的过程如图 1-2 所示，其中， $\text{Mod}(S, 11)$ 表示 S 除以 11 得到的余数。

[流程图]

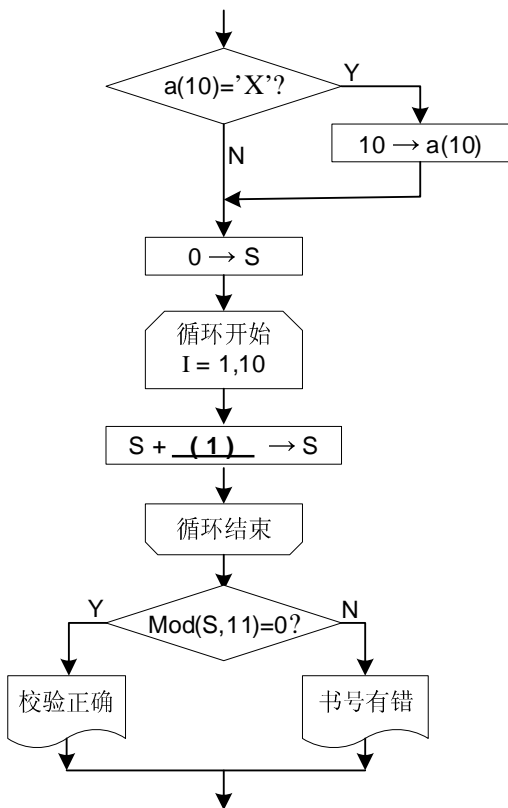


图 1-1 校验过程

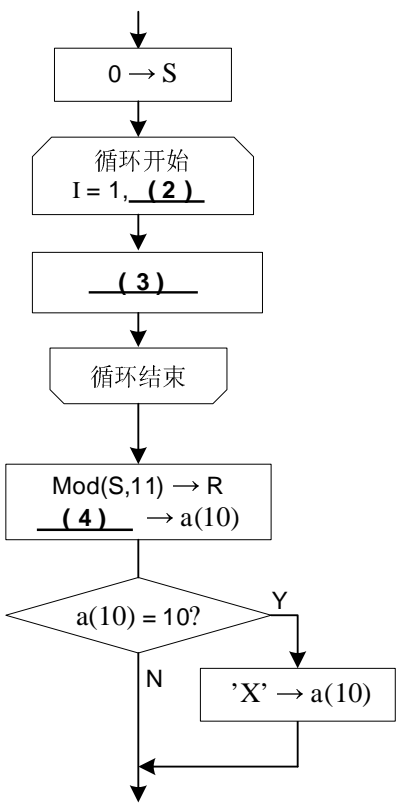


图 1-2 计算校验码的过程

[问题 1] 请填补流程图中的空缺 (1) ~ (4)。

[问题 2] 设“程序员考试大纲”标准书号前 9 个数字为 7-302-08493，请写出其校验码。

试题二（共 15 分）

阅读以下说明和 C 语言函数，将解答填入答题纸的对应栏内。

[说明]

下面待修改的 C 程序完成的功能是：对于给定的一个长正整数，从其个位数开始，每隔一位取一个数字（即取其个位、百位、万位等数字），形成一个新的整数并输出。例如，将该程序修改正确后，运行时若输入“14251382”，则输出的整数为“4532”。

下面给出的 C 程序代码中有五个错误，请指出所有的错误。

[C 程序代码]

行号	代码
01	#include <stdio.h>
02	
03	int main()
04	{
05	long n, num;
06	int i;
07	
08	do {
09	printf("请输入一个正整数: ");
10	scanf("%ld", n);
11	}while(n <= 0);
12	k = 1;
13	for (i = 1; n >= 0; i++) {
14	if (i % 2 = 1) {
15	num = num + (n % 10) * k;
16	k = k * 10;
17	}
18	n = n / 10;
19	}
20	printf("新数据为: %d \n", num);
21	return 0;
22	}

试题三（共 15 分）

阅读以下说明和 C 语言函数，将应填入 (n) 处的字句写在答题纸的对应栏内。

[说明]

函数 count_months(DATE start, DATE end)的功能是：计算两个给定日期之间所包含的完整月份数。

该函数先算出起止日期中所含的完整年数，再计算余下的完整月份数。
 规定两个相邻年份的同月同日之间的间隔为 1 年。例如，2007. 5. 30~2008. 5. 30 的间隔为 1 年。若相邻两年中前一年是闰年，并且日期是 2 月 29 日，则到下一年的 2 月 28 日为 1 年, 即 2008. 2. 29~2009. 2. 28 的间隔为 1 年。

规定两个相邻月份的相同日之间的间隔为 1 个月, 但需要特别考虑月末的特殊情况。例如，2007. 1. 29~2007. 2. 28 的间隔为 1 个月，同理， 2007. 1. **30**~2007. 2. 28、2007. 1. **31**~2007. 2. 28 的间隔都是 1 个月。

- 计算起止日期间隔不足一年的完整月份数时，分两种情况：
- 1) 起止日期不跨年度。先用终止日期的月号减去起始日期的月号得到月份数，然后再根据情况进行修正。例如，起止日期为 2008. 3. 31~2008. 9. 20，通过月号算出月份数为 6。修正时，通过调用函数 makevalid 将 2008. 9. 31 改为 2008. 9. 30，与终止日期 2008. 9. 20 比较后，将月份数修正为 5。
 - 2) 起止日期跨年度。计算方法如下例所示：对于起止日期 2008. 7. 25~2009. 3. 31，先计算 2008. 7. 25~2008. 12. 25 的月份数为 5，再算出 2008. 12. 25~2009. 3. 25 的月份数为 3，因此 2008. 7. 25~2009. 3. 31 之间的完整月份数为 8。

日期数据类型定义如下：

```

typedef struct {
    int year;    int month;    int day; /*日期的年号（4 位）、月和日号*/
}DATE;

```

程序中使用的函数 cmp_date()、isLeapYear() 和 makevalid() 说明如下：

函数名	参数	返回值	说明
cmp_date	DATE start, DATE end	-1: start<end 0: start=end 1: start>end	比较两个日期的大小，例如： 2007. 1. 31 小于 2007. 5. 15， 2008. 11. 23 等于 2008. 11. 23， 2008. 1. 31 大于 2007. 5. 15
isLeapYear	int year	1:year 表示的年号是闰年 0:year 表示的年号不是闰年	判断给定年号是否为闰年
makevalid	DATE *r	无	若日期*r 是非法的，即*r 不是闰年时其日期为 2 月 29 日，或者其 4、6、8、10 等月份出现了 31 日，则将其日期改为当月最后一天的日期。

[C 语言函数]

```
int count_months (DATE start, DATE end)
{
    int years = 0, months = 0;
    DATE r;

    if (cmp_date(start, end) > 0) {
        r = start; start = end; end = r;
    }

    years = end.year - start.year; /*计算年数*/
    r = start;
    r.year = end.year;

    if (cmp_date(r, end) > 0) { /*修正年数*/
        (1);
        r.year--;
    }

    if (r.year < end.year) { /*跨年度时, 先计算到 12 月的月份数*/
        months = (2);
        r.month = 12;
    }

    months += (end.month + 12 - r.month) % 12;
    r.year = end.year;    r.month = end.month;
    makevalid((3)); /*将日期 r 修正为有效日期*/
    if (cmp_date(r, end) > 0) /*修正月份数*/
        (4);

    months += (5); /*计算总月份数*/
    return months;
}
```

试题四（共 15 分）

阅读以下说明和 C 语言函数，将应填入 (n) 处的字句写在答题纸的对应栏内。

[说明]

函数 `sort(NODE *head)` 的功能是：用冒泡排序法对单链表中的元素进行非递减排序。对于两个相邻结点中的元素，若较小的元素在前面，则交换这两个结点中的元素值。其中，`head` 指向链表的头结点。排序时，为了避免每趟都扫描到链表的尾结点，设置一个指针 `endptr`，使其指向下趟扫描需要到达的最后一个结点。例如，对于图 4-1 (a) 的链表进行一趟冒泡排序后，得到图 4-1 (b) 所示的链表。

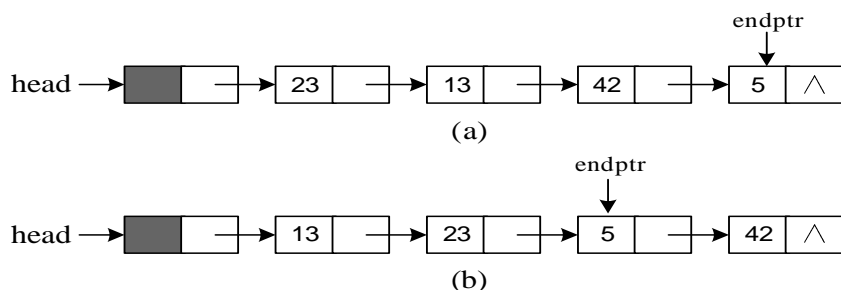


图 4-1

链表的结点类型定义如下：

```
typedef struct Node {  
    int data;  
    struct Node *next;  
} NODE;
```

[C 语言函数]

```
void sort(NODE *head)  
{  
    NODE *ptr,*preptr,*endptr;  
    int tempdata;  
    ptr = head->next;  
    while (     (1)     ) /*查找表尾结点*/  
        ptr = ptr->next;  
    endptr = ptr; /*令 endptr 指向表尾结点*/  
  
    ptr =     (2)    ;  
    while(ptr != endptr) {  
        while(     (3)     ) {  
            if (ptr->data > ptr->next->data) {  
                tempdata = ptr->data; /*交换相邻结点的数据*/  
                ptr->data = ptr->next->data;  
                ptr->next->data = tempdata;  
            }  
            preptr =     (4)    ;          ptr = ptr->next;  
        }  
        endptr =     (5)    ;    ptr = head->next;  
    }  
}
```

试题五（共 15 分）

阅读以下应用说明以及用 Visual Basic 开发过程中所编写的程序代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

[应用说明]

某应用的登录界面如下：



登录界面中的文本框 txtUserID 和 txtPassword 分别用于接受用户输入的用户名和口令，命令按钮为确定(cmdOk)和取消(cmdCancel)。ADO 数据控件 Adodc1 与数据库 EmployeeData 连接。在 EmployeeData 中，表“Users”的字段“UserID”和“Pswd”分别记录了所有的合法用户名和口令。

在登录界面中，输入正确的用户名和口令并单击“确定”按钮后，就会打开应用程序的主窗口 frmApp。用户名或密码输入错误时，给出提示信息并返回登录界面。

[Visual Basic 程序代码]

```
Private Sub cmdOK_Click()  
    Adodc1.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=D:\data\EmployeeData.mdb;Persist Security Info=False"  
    Adodc1.RecordSource = "select * ____ (1) ____"  
    Adodc1.Refresh  
    Do  
        username = ____ (2) ____  
        pwd = Adodc1.Recordset("Pswd")  
        Adodc1.Recordset.MoveNext  
    Loop Until (username = ____ (3) ____) Or (Adodc1.Recordset.EOF)  
    If username <> txtUserID.Text Then  
        MsgBox "无效的用户 ID, 请重新输入!", , "提示信息"  
    Else  
        If ____ (4) ____ Then  
            MsgBox "欢迎进入系统主界面!", , "提示信息"  
            Unload me  
            ____ (5) ____ ' 打开应用程序的主窗口  
        Else  
            MsgBox "无效的用户密码, 请重新输入!", , "提示信息"  
        End If  
    End If  
End Sub
```

试题六（共 15 分）

阅读下列说明和 C++ 代码，请回答问题 1 至问题 3，将解答写在答题纸的对应栏内。

[说明]

已知下列程序运行时的输出应为：

1:1

1:1

1:1

[问题 1]

请补齐下述代码中的空缺 1 和 2。

[问题 2]

类 Stock 的定义中有两处错误，请指出错误所在行号并给出该行修改后的完整结果。

[问题 3]

指出该代码是否存在内存泄漏。

[C++ 程序]

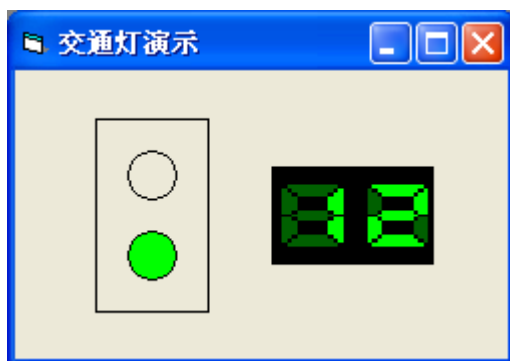
```
01 #include <iostream>
02 using namespace std;
03 class Stock{
04 protect:
05     (1) {};
06     Stock(int n, int pr=1) {
07         shares = n; share_val=pr;
08     };
09     void ~Stock() {};
10 public:
11     //成员函数
12     void output() {
13         (2) << shares << ":" << share_val << endl;
14     }
15 private:
16     //成员变量
17     int shares;
18     int share_val;
19 };
20
21 void main() {
22     Stock a(1);        a.output();
23     Stock b;           b.output();
24     Stock c = Stock(); c.output();
25 }
```


试题七（共 15 分）

阅读以下应用说明、Visual Basic 开发说明以及程序代码，将应填入 (n) 的字句写在答题纸的对应栏内。

[应用说明]

某交通灯演示程序启动后，其运行窗口中包括红绿灯标志以及当前红绿灯状态剩余秒数的动态显示（如下图）。红灯与绿灯持续的时间分别为 30 秒与 40 秒。持续时间结束时，立即变更红绿灯，并重新显示剩余秒数。



[Visual Basic 开发说明]

在开发过程中，先在窗体内制作形状“红绿灯框架”（长方形，透明），在该框架中，上部制作形状“红灯”（ShpRed，圆形），默认属性为透明；下部制作形状“绿灯”（ShpGreen，圆形），初始设置其填充方式属性 FillStyle 为实心，填充颜色属性 FillColor 为绿色。

再设置两个图像框（Image1 在左，Image2 在右），使其能分别装入数字图像文件，以显示当前红绿灯状态剩余的秒数。10 个数字图像文件 N0. bmp~N9. bmp 分别用于显示数字 0、1、…、9，这些文件存放在该应用程序所在目录中，以便在程序运行时按需要选择装入这两个图像框。为使图像文件装入图像框时能自动改变大小以适应图像框，这些图像框的 Stretch 属性都应设置成 (1)。为使这些数字能每秒变化一次，开发窗体中设置了计时器 Timer1，其 Enable 属性设置成 True，其 Interval 属性设置成 1000。

程序中，全局变量 Light 用以标志当前红灯（0）或绿灯状态（1），RedT 和 GreenT 分别表示红灯或绿灯状态剩余的秒数。

计时器 Timer1 的定时过程是在该程序启动 1 秒后首次执行的，以后每隔 1 秒执行 1 次。开发者应考虑对以下四种情况的处理：绿灯持续、绿灯转红灯、红灯持续、红灯转绿灯。

过程 LoadNumber 的功能是：将红绿灯剩余秒数 N（2 位整数）分离出两个一位数 i 与 j，再将其转换成字符，删除前面可能产生的空格，再将数字图像文件 N*i*. bmp 与 N*j*. bmp 装入两个图像框（其中 *i* 与 *j* 应以相应的数字字符代替）。

[Visual Basic 程序代码]

```
Public Light As Integer, RedT As Integer, GreenT As Integer
Private Sub Form_Load()
    Light = 1 : GreenT = 40 : RedT = 0 ' 初始化设置
End Sub
Private Sub Timer1_Timer() ' 计时器定时过程
    If Light = 1 Then ' 绿灯状态时
        (2)
        If GreenT > 0 Then ' 绿灯持续时
            Call LoadNumber(GreenT) ' 调用过程，显示数字 GreenT
        Else ' 绿灯转红灯时
            ShpRed.FillStyle = 0 ' 置红灯形状实心
            ShpRed.FillColor = vbRed ' 为红灯形状填充红色
            ShpGreen.FillStyle = 1 ' 置绿灯形状透明
            Call LoadNumber(30)
            Light = 0 : RedT = 30
        End If
    Else ' 红灯状态时
        RedT = RedT - 1
        If RedT > 0 Then ' 红灯持续时
            Call LoadNumber(RedT)
        Else ' 红灯转绿灯时
            ShpRed.FillStyle = 1
            ShpGreen.FillStyle = 0
            ShpGreen.FillColor = vbGreen
            (3)
            Light = 1 : GreenT = 40
        End If
    End If
End Sub
Sub LoadNumber(N As Integer) ' 根据剩余秒数 N 将数字图像装入图像框
    Dim i As Integer, j As Integer
    i = Int( (4) )
    j = (5)
    Image1.Picture = LoadPicture("N"&Trim(Str(i))&". bmp") ' 显示十位上的数字
    Image2.Picture = LoadPicture("N"&Trim(Str(j))&". bmp") ' 显示个位上的数字
End Sub
```

试题八（共 15 分）

阅读以下说明和 Java 代码，请回答问题 1 和问题 2，将解答写在答题纸的对应栏内。

[说明]

已知类 Stock 和类 cxyjava 都定义在 cxyjava.java 文件中，类 Stock 的定义中第 14 行前共有四行出现了错误，将下面代码修改正确并完善后的输出结果为：

0:0

1:23

[问题 1]

请指出错误所在行号并给出该行修改后的完整结果。

[问题 2]

请补齐第 21 行代码中的空缺。

[Java 代码]

```
01 public class Stock{
02     static {
03         shares = 0;
04         share_val = 0.0;
05     }
06     public Stock() {getData();}
07     public Stock(int n, int pr=0) {
08         shares = n;
09         share_val = pr;
10         getData();
11     }
12     public void getData() {
13         System.out.println(shares + ":" + share_val);
14     }
15     private int shares;
16     private int share_val;
17 };
18
19 public class cxyjava{
20     public static void main(String args[]) {
21         Stock a = ____ (1) ____;
22         Stock b = new Stock(1,23);
23         //其它无输出代码省略
24     }
25 }
```