

全国计算机技术与软件专业技术资格（水平）考试

2005 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

2005 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 29 日”，故在答题纸的对应栏内写上“5”和“29”（参看下表）。

例题	解答栏
(1)	5
(2)	29

试题一至试题三是必答题

试题一（15 分，每空 3 分）

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

[说明]

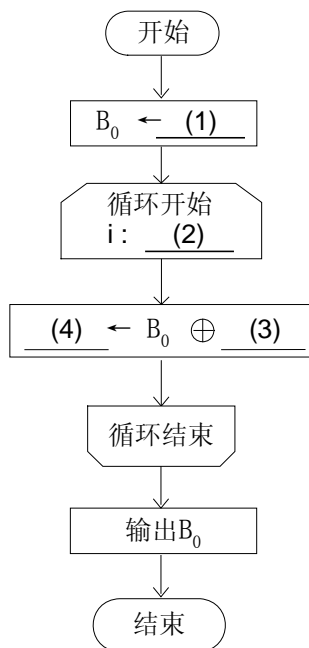
设 8 位二进制代码 $B_0B_1...B_7$ 中的最高位 B_0 为奇偶校验位。对于任何给定的代码 $B_1B_2...B_7$ ，可按下式计算偶校验位：

$$B_0 = B_1 \oplus B_2 \oplus ... \oplus B_7$$

其中，“ \oplus ”表示“异或”运算。

下面的流程图描述了计算偶校验位的过程。

[流程图]



注：流程图中，循环开始的说明按照“循环变量名：循环初值，循环终值，增量”格式描述。

[问题 1] 将流程图中的（1）～（4）处补充完整。

[问题 2] 若按下式计算奇校验位，则上述流程图中的（1）处应填_____（5）_____。

$$B_0 = B_1 \oplus B_2 \oplus ... \oplus B_7 \oplus 1$$

试题二 (15 分, 每空 3 分)

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

[函数 2.1 说明]

函数 int fun1(int m, int n)的功能是: 计算并返回正整数 m 和 n 的最大公约数。

[函数 2.1]

```
int fun1(int m, int n)
{
    while(   (1)   ) {
        if ( m > n ) m = m - n;
        else n = n - m;
    }
      (2)  ;
}
```

[函数 2.2 说明]

函数 long fun2(char *str)的功能是: 自左至右顺序取出非空字符串 str 中的数字字符形成一个十进制整数(最多 8 位)。

例如, 若字符串 str 的值为 “f3g8d5.ji2e3p12fkp”, 则函数返回值为 3852312。

[函数 2.2]

```
long fun2(char *str)
{
    int i = 0;
    long k = 0;
    char *p = str;
    while ( *p != '\0' &&   (3)   ) {
        if ( *p >= '0' && *p <= '9' ) {
            k =   (4)   + *p - '0';
            ++i;
        }
          (5)  ;
    }
    return k;
}
```

试题三 (15 分, 每空 3 分)

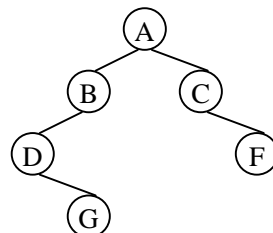
阅读以下说明和C语言函数, 将应填入 (n) 处的字句写在答题纸的对应栏内。

[说明]

一棵非空二叉树中“最左下”结点定义为: 若树根的左子树为空, 则树根为“最左下”结点; 否则, 从树根的左子树根出发, 沿结点的左孩子分支向下查找, 直到某个结点不存在左孩子时为止, 该结点即为此二叉树的“最左下”结点。例如, 下图所示的以 A 为根的二叉树的“最左下”结点为 D, 以 C 为根的子二叉树中的“最左下”结点为 C。

二叉树的结点类型定义如下:

```
typedef struct BSTNode{
    int data;
    struct BSTNode *lch,*rch; //结点的左、右孩子指针
}*BSTree;
```



函数 BSTree Find_Del (BSTree root) 的功能是: 若 root 指向一棵二叉树的根结点, 则找出该结点的右子树上的“最左下”结点 *p, 并从树中删除以 *p 为根的子树, 函数返回被删除子树的根结点指针; 若该树根的右子树上不存在“最左下”结点, 则返回空指针。

[函数]

```
BSTree Find_Del (BSTree root)
{
    BSTree p, pre;
    if ( !root ) return NULL;          /* root 指向的二叉树为空树 */
    (1) _____;                    /* 令p指向根结点的右子树 */
    if ( !p ) return NULL;
    (2) _____;                    /* 设置pre的初值 */
    while( p->lch ) {                    /* 查找“最左下”结点 */
        pre = p; p = (3) _____;
    }
    if ( (4) _____ == root )      /* root的右子树根为“最左下”结点 */
        pre->rch = NULL;
    else
        (5) _____ = NULL;        /* 删除以“最左下”结点为根的子树 */
    return p;
}
```

试题四 (15 分, 每空 3 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]

假设一个剧场有 $N \times N$ 个座位, 顾客买票时可以提出任意有效的座号请求。下面用二维数组 $a[N][N]$ 模拟剧场中的座位, $a[i][j]$ 等于 0 表示第 i 排第 j 列 ($0 \leq i, j \leq N-1$) 的票尚未售出。

函数 $\text{int Find}(\text{int } a[][N], \text{int } R, \text{int } *row, \text{int } *col)$ 的功能是: 在部分票已售出的情况下, 找出剧场中的 $R \times R$ 个空座位, 要求这些座位的排列形成一个正方形。若找到满足要求的一个座位排列, 则函数返回 1, 并算出该正方形左上角的行、列号; 若未找到, 返回 0。

例如, 一个 7×7 个座位的剧场如下图(a)所示, 已售出部分座位的剧场如下图(b)所示, 图中阴影部分表示已售出的座位, 从图(b)中找出的 3×3 正方形空座位如图(c)中斜线区所示。

	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

(a) 剧场

	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

(b) 已售出部分座位的剧场

	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

(c) 找出 3×3 个空座位

[函数]

```
int Find(int a[][N], int R, int *row, int *col)
{ int i, j, k, c, t; int FOUND = 0;
  for(i=0; !FOUND && i < N - R + 1; i++) { /*从第 0 排开始查找*/
    (1) ;
    while (j < N-R+1 && !FOUND) {
      for(k = 0; (2) && a[i][j+k]==0; k++); /*查找第i排连续的R个空座位*/
      if (k >= R) { /*找到第 i 排连续的 R 个空座位*/
        for(c = 0; c < R; c++) { /*查找其余的 R*(R-1) 个座位*/
          for(t = 1; t < R; t++)
            if (a[(3)] [j+c] != 0) break;
          if (t < R) break;
        } /* for */
        if ((4) FOUND = 1;
      } /* if */
      (5) ;
    } /* while */
  } /* for i */
  if (FOUND) {
    *row = i - 1; *col = j - 1; /*计算正方形区域的左上角坐标*/
    return 1;
  }
  return 0;
}
```

试题五（15 分，每空 3 分）

阅读以下说明及Visual Basic程序代码，将应填入____(n)____处的字句写在答题纸的对应栏内。

[说明]

甲乙两地之间的铁路线上有二十多个火车站，为方便乘客查询票价，特开发了火车票价查询程序，其运行窗口如下图所示。乘客只要通过两个下拉列表框分别选择起点与终点站名，再单击“查询”按钮，则票价文本框中就会显示相应的票价。

在开发过程中，设选择起点和终点站名的组合框 (ComboBox) 名分别为 ST 和 SP，显示票价的文本框 (TextBox) 名为 Price，“查询”按钮 (CommandButton) 名为 Query；设 ST 与 SP 中的诸列表项完全相同，且都是从甲地开始依次排列，从第 i 站到第 j 站的票价存放在二维对称矩阵 P 的元素 P(i, j) 中。假设从甲地到第 i 站行驶的距离存放在 d(i) 中，火车票价等于其行驶距离乘以 C（常数）。

[Visual Basic 代码]

```
Dim p(30, 30) As Single          ' 通用声明
Const C=0.2
Private Sub Form_Load()
    Dim d(30) As Single
    ____ (1) ____ = "甲地"        ' 在ST组合框中设置首个列表项
    .....                        ' 在 ST 和 SP 中设置其他列表项（若干行语句）
    .....                        ' 对数组 d(i)赋值（若干行语句）
    For i=0 to ST. ____ (2) ____ -1 ' 对ST中各列表项循环
        For j=0 to SP. ____ (2) ____ -1 ' 对SP中各列表项循环（这两处填写内容相同）
            P(i, j)= ____ (3) ____
        Next j
    Next i
End Sub
Private Sub query_Click()
    i = ____ (4) ____              ' 组合框ST中选择的当前项编号
    j = .....                    ' 组合框 SP 中选择的当前项编号（语句略）
    ____ (5) ____ =str$(p(i, j))
End Sub
```

试题六（15 分，每空 3 分）

阅读以下说明、Java代码和HTML文档，将应填入 (n) 处的字句写在答题纸的对应栏内。

[说明]

当用户启动 html 浏览器并首次打开下面的 html 文档时，Java Applet 小程序在显示面板上显示字符串“Welcome !”；当 html 页面被其它窗口遮挡后再次显示时，小程序在显示面板上显示“Welcome back !”。

[Java 代码]

```
import java.awt.*;
import java.applet.*;
public class HelloApplet extends     (1)     {
    public void paint(    (2)    ) {
        g.drawString(message, 10, 20);
        message = "welcome back ! ";
    }
    public void     (3)    () {
        message = "Welcome ! ";
    }
    private     (4)     message;
}
```

[HTML 文档]

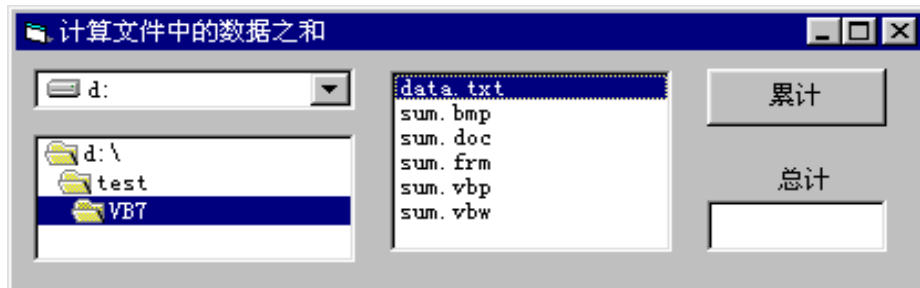
```
<html>
<head>
    <title> HTML Test HelloApplet Page</title>
</head>
<body>
    Test HelloApplet 小程序<br>
    <applet
        codebase = "."
        code = "    (5)    "
        name = "TestApplet"
        width = "400"
        height = "300"
    >
</applet>
</body>
</html>
```

试题七（15 分，每空 3 分）

阅读以下说明及 Visual Basic 程序代码，将应填入____(n)____处的字句写在答题纸的对应栏内。

[说明]

某个文本文件中存放了若干个非零数值有效数据，每个数据占一行，最后一行以数字“0”作为结束标志。下面的程序用于计算该文件中这些数据之和，其运行窗口如下：



用户只要通过驱动器列表框、目录列表框和文件列表框，选择文本文件所在的驱动器、文件夹及相应的文件名，再单击“累计”按钮，就能在“总计”文本框中看到该文本文件中的数据之和。

在开发过程中，设驱动器列表框名为 Drive1，目录列表框名为 Dir1，文件列表框名为 File1，存放数据的文本文件名为 D:\test\vb7\data.txt，“累计”按钮名为 sum，“总计”文本框(TextBox)名为 text1。

[Visual Basic 代码]

```
Private Sub Drive1_Change()  
    Dir1.Path = ____ (1) ____ ' 用户选择的驱动器反映到目录列表框中  
End Sub  
Private Sub Dir1_Change()  
    File1.Path = ____ (2) ____ ' 用户选择的文件夹改变文件列表框内容  
End Sub  
Private Sub sum_Click()  
    Dim ____ (3) ____ As New FileSystemObject, ____ (4) ____ As TextStream  
    ' 创建自己的文件系统对象，说明自己的文本流对象  
    If Right(Dir1.Path, 1) = "\" Then  
        Set myTextFile = fso.OpenTextFile(Dir1.Path + File1.FileName)  
    Else  
        Set myTextFile = fso.OpenTextFile(Dir1.Path + "\" + File1.FileName)  
    End If  
    S = 0  
    Do  
        d = Val(myTextFile.ReadRow)  
        S = S + d  
    Loop While ____ (5) ____  
    Text1.Text = Str$(S)  
End Sub
```


试题八（15 分，每空 3 分）

阅读以下说明和C++程序，将应填入 (n) 处的字句写在答题纸的对应栏内。

[说明]

下面程序的功能是计算并输出某年某月的天数。

[C++程序]

```
#include <iostream>
using namespace std;
____(1)____ Month{Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec};
class Date{
public:
    Date(int year, Month m_month){
        ____ (2) ____ = year;
        if(m_month < Jan || m_month > Dec) month = Jan;
        else month = m_month;
    };
    ~Date() {};
    bool IsLeapYear() {
        return ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0);
    };
    int CaculateDays() {
        switch(____ (3) ____){
            case Feb:{
                if(____ (4) ____) return 29;
                else return 28;
            }
            case Jan: case Mar: case May: case Jul: case Aug: case Oct:
            case Dec: return 31;
            case Apr: case Jun: case Sep: case Nov: return 30;
        }
    };
private:
    int year;
    Month month;
};

void main() {
    Date day(2000, Feb);
    cout << day.____ (5) ____();
}
```