

# 全国计算机技术与软件专业技术资格（水平）考试

## 2005 年下半年 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 年下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“11 月 5 日”，故在答题纸的对应栏内写上“11”和“5”（参看下表）。

例题	解答栏
(1)	11
(2)	5

试题一（共 15 分）

阅读下列说明和流程图，将应填入\_\_ (n) \_\_处的字句写在答题纸的对应栏内。

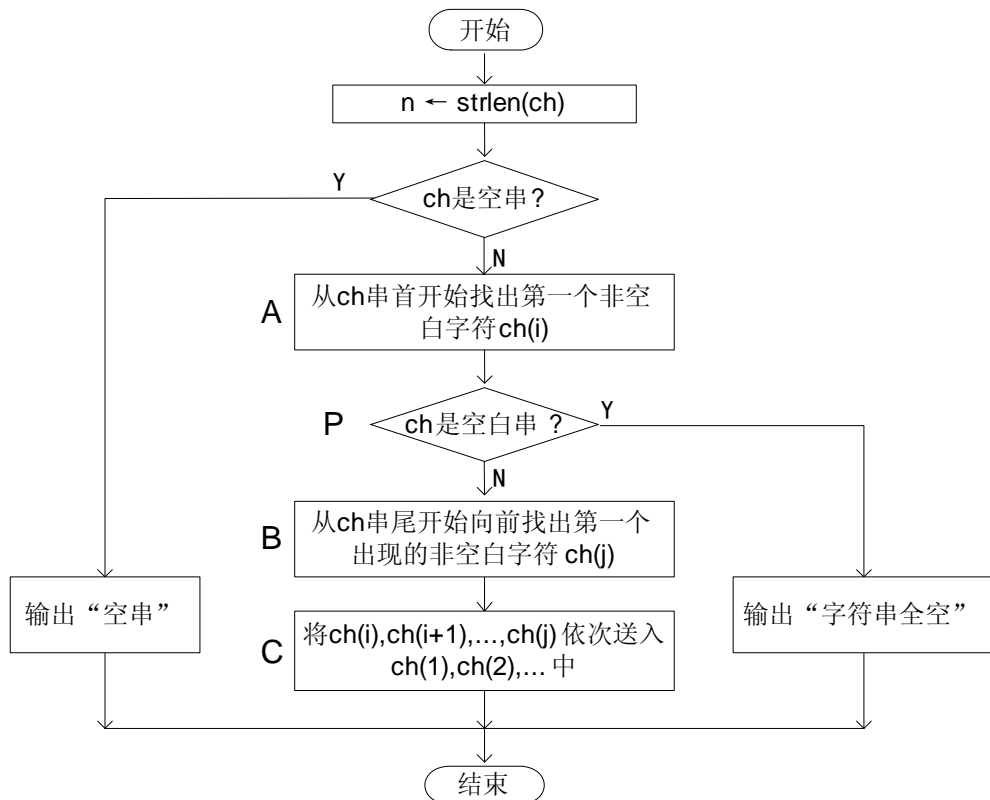
[流程图说明]

流程图 1-1 描述了一个算法，该算法将给定的原字符串中的所有前导空白和尾部空白都删除，但保留非空字符之间的空白。例如，原字符串“ File Name ”，处理后变成“File Name”。流程图 1-2、流程图 1-3、流程图 1-4 分别详细描述了流程图 1-1 中的框 A、B、C。

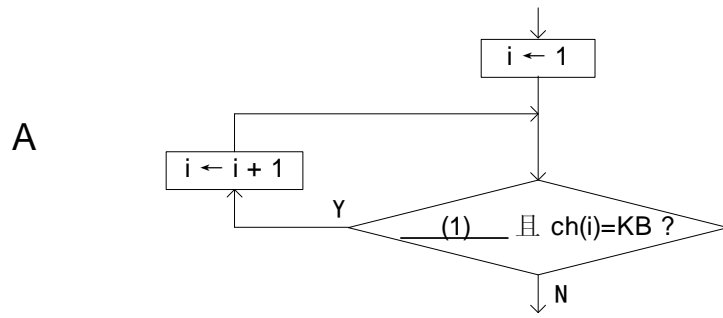
假设原字符串中的各个字符依次存放在字符数组 ch 的各元素 ch(1)、ch(2)、...、ch(n) 中，字符常量 KB 表示空白字符。

流程图 1-1 的处理过程是：先从头开始找出该字符串中的第一个非空白字符 ch(i)，再从串尾开始向前找出位于最末位的非空白字符 ch(j)，然后将 ch(i)、...、ch(j) 依次送入 ch(1)、ch(2)、... 中。如果原字符串中没有字符或全是空白字符，则输出相应的说明。在流程图中，strlen 是取字符串长度函数。

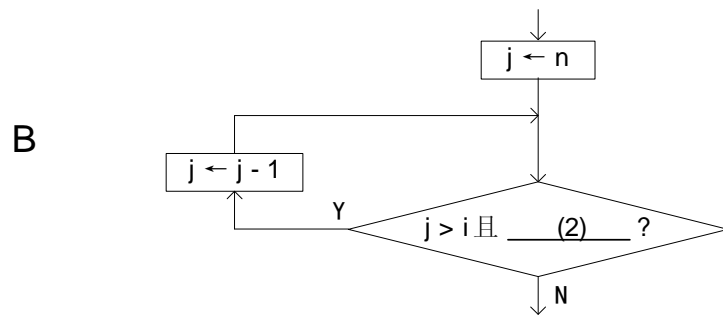
[流程图 1-1]



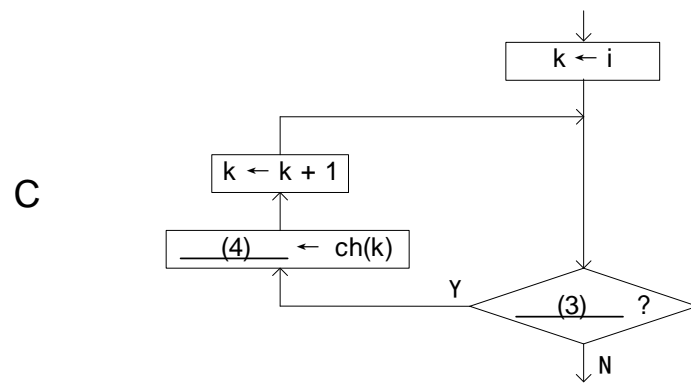
[流程图 1-2]



[流程图 1-3]



[流程图 1-4]



[问题]在流程图 1-1 中，判断框 P 中的条件可表示为：

$$i > \underline{\hspace{2cm}} (5)$$

## 试题二（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 `int find_Max_Min(int a[], int n)` 的功能是：找出  $n$  个元素的数组  $a$  中的最大元素和最小元素并输出，返回查找过程中元素的比较次数。查找方法如下：比较  $a[0]$  和  $a[n-1]$ ，若  $a[0]$  大，则交换  $a[0]$  和  $a[n-1]$  的值；再比较  $a[1]$  和  $a[n-2]$ ，若  $a[1]$  大，则交换  $a[1]$  和  $a[n-2]$  的值；依此类推，直到所有的元素都比较完。然后在数组的前半区从前往后找出小元素，在后半区从后往前找出大元素。

### [函数]

```
int find_Max_Min(int a[], int n)
{ /*找出 n 个元素的数组 a 的最大、最小元素并输出，返回查找过程元素中的比较次数*/
  int i, Count = 0;
  int temp, Maxnum, Minnum;
  for(i = 0; i < n/2; i++) {
    Count = Count + 1;          /*元素比较次数计数*/
    if (a[i] > a[      (1)      ])
      { /*数组元素交换代码略*/ }
  }
  Maxnum = a[n-1];  Minnum = a[0];
  for(i = 1; i < n/2 + n%2; i++) {
    Count =       (2)      ;          /*元素比较次数计数*/
    Minnum =       (3)       ? a[i] : Minnum;          /*找最小元素*/
    Maxnum =       (4)       ?       (5)       : Maxnum;          /*找最大元素*/
  }
  printf("Max=%d\n", Maxnum);
  printf("Min=%d\n", Minnum);
  return Count;
}
```

试题三（共 15 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]

某种传感器的输出值 Ratio 依赖于环境温度 temp ( $-40^{\circ}\text{C} \leq \text{temp} \leq 50^{\circ}\text{C}$ )。对一组环境温度值 (ITEMS 个)，人们已经测量得到了相应的 Ratio 值 (见表 1)。该表粗略地描述了曲线 Ratio(temp)。

表 1 曲线 Ratio(temp) 的列表值

环境温度 temp	传感器的输出值 Ratio
$-40^{\circ}\text{C}$	0.20
$-20^{\circ}\text{C}$	0.60
$-10^{\circ}\text{C}$	0.80
$0^{\circ}\text{C}$	1.00
$10^{\circ}\text{C}$	1.17
$30^{\circ}\text{C}$	1.50
$50^{\circ}\text{C}$	1.80

表 2 曲线 K(temp) 的列表值

环境温度 temp	校正系数 K
$-40^{\circ}\text{C}$	5.00
$-39^{\circ}\text{C}$	4.55
$-38^{\circ}\text{C}$	4.17
$-37^{\circ}\text{C}$	3.85
...	...
$-20^{\circ}\text{C}$	1.67
$-19^{\circ}\text{C}$	1.61
...	...
$-10^{\circ}\text{C}$	1.25
$-9^{\circ}\text{C}$	1.22
...	...
$50^{\circ}\text{C}$	0.56

校正系数 K 是 Ratio 的倒数，因此也依赖于环境温度 temp。在数据处理中，人们需要用更多的列表值细致地描述曲线 K(temp)，如表 2 所示。在表 2 中，各温度值所对应的 K 值是对表 1 进行线性插值再求倒数得到的，具体的计算方法如下：

1. 根据 temp 值，在表 1 中用二分法查找；
2. 若找到相应的温度值，则按相应的 Ratio 值求倒数得到 K 值；
3. 若没找到相应的温度值，则可确定 temp 所在的温度区间 [Tp1, Tp2]，同时获得了相应的 Ratio1 和 Ratio2，再按如下公式计算 K 值：

$$\text{Step} = (\text{Ratio1} - \text{Ratio2}) / (\text{Tp1} - \text{Tp2})$$
$$K = 1.0 / (\text{Ratio1} + \text{Step} * (\text{temp} - \text{Tp1}))$$

在程序中，当 temp 高于  $50^{\circ}\text{C}$  或低于  $-40^{\circ}\text{C}$  时，设定  $K=0$ 。

[程序]

```
#include <stdio.h>
typedef struct {
    int Temp;          /*环境温度*/
    double Ratio;     /*传感器的输出值*/
}CURVE;

#define ITEMS 7
double GetK(int, CURVE *, int);

void main()
{
    int Degree;
    double k;
    CURVE Curve[ITEMS] = { {-40, 0.2}, {-20, 0.60}, {-10, 0.8}, {0, 1.0},
                           {10, 1.17}, {30, 1.50}, {50, 1.8} };
    printf("环境温度      校正系数\n");
    for( Degree = -40; Degree <= 50; Degree++ ) {
        k = GetK(Degree, Curve, ITEMS);
        printf(" %3d          %4.2f \n", Degree, k);
    }
}

double GetK(int Temp, CURVE *p, int n)
{ /*用二分法在 n 个元素的有序表 p 中查找与 Temp 对应的传感器输出值*/
    int low, high, m; double Step;
    low = 0; high = n-1;
    if ( (Temp < p->Temp) || (Temp > (p+high)->Temp) )
        return 0.0; /*超出温度范围时返回 0.0*/
    while (low <= high) {
        m = _____(1)_____;
        if ( Temp == (p+m)->Temp )
            return _____(2)_____;
        if ( Temp < (p+m)->Temp ) high = m-1;
        else low = _____(3)_____;
    }
    p += high;
    Step = (_____(4)_____) / ((p+1)->Temp - p->Temp);
    return 1.0/(p->Ratio +Step * (_____(5)_____));
}
```

从下列的 2 道试题（试题四至试题五）中任选 1 道解答。  
如果解答的试题数超过 1 道，则题号小的 1 道解答有效。

#### 试题四（共 15 分）

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

##### [应用说明]

启动某应用程序运行的登录窗口如下：



其中的标签 (Label1) “用户名 (U)” 对应文本框 Username，标签 (Label2) “密 码 (P)” 对应文本框“Password”。当用户在 Username 文本框中输入用户名“ali88”，在 Password 文本框中输入“zkm”（显示的字符均为“\*”），并单击“确定”按钮时，就会弹出应用程序的主窗口 frmAPP，而该登录窗口就会卸载。

如果用户名或密码输入错误，则会弹出报警信息框。当用户单击其中的“确定”按钮后，登录窗口中 Password 文本框的内容就会消失，光标停留在该框内，以使用户重新输入密码，必要时用户还可以再修改用户名，再次做登录尝试。本应用程序允许发生 3 次输入错误。在第 3 次输入错误后，就会立即退出该应用程序。

在弹出登录窗口后，当按键“Alt+U”时光标就会停留在 Username 文本框中；当按键“Alt+P”时光标就会停留在 Password 文本框中。当用户按“Enter”键时，就相当于单击“确定”按钮；当用户按“Esc”键时，就相当于单击“取消”按钮，立即退出该应用程序。

##### [属性设置]

在开发过程中，部分控件及属性设置如下：

对象	对象名	属性	属性值
标签	Label2	Caption	(1)
文本框	Username	Text	(空白)
文本框	Password	Text	(空白)
		Password	*
命令按钮	CmdOK	Caption	确定
		Default	(2)
命令按钮	CmdCancel	Caption	取消
		Cancel	True

##### [程序代码]

```

Private Sub cmdOK_Click()
    _____ (3) _____ As Integer          ' 静态变量 times 的说明
    If _____ (4) _____ Then
        Unload Me                                ' 卸载本登录窗口
        _____ (5) _____                ' 弹出应用程序主窗口 frmAPP
    Else
        MsgBox ("用户名或密码错！")
        times = times + 1
        Password.Text = ""                       ' 清除密码框中的内容
        Password.SetFocus                       ' 将光标定位于密码框
        If times = 3 Then End
    End If
End Sub

Private Sub cmdCancel_Click()
    End
End Sub

```

### 试题五（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]

二叉排序树或者是一棵空树，或者是具有如下性质的二叉树：若它的左子树非空，则左子树上所有结点的值均小于根结点的值；若它的右子树非空，则右子树上所有结点的值均大于根结点的值；左、右子树本身就是两棵二叉排序树。

函数 insert\_BST(char \*str)的功能是：对给定的字符序列按照 ASCII 码值大小关系创建二叉排序树，并返回指向树根结点的指针。序列中重复出现的字符只建一个结点，并由结点中的 Count 域对字符的重复次数进行计数。

二叉排序树的链表结点类型定义如下：

```
typedef struct BSTNode{
    char Elem;           /*结点的字符数据*/
    int Count;          /*记录当前字符在序列中重复出现的次数*/
    struct BSTNode *Lch,*Rch; /*结点的左、右子树指针*/
}*BiTree;
```

#### [函数]

```
BiTree insert_BST(char *str)
{ BiTree root,parent,p;
  char     (1)    ; /* 变量定义及初始化 */
  root = (BiTree)malloc(sizeof(struct BSTNode));
  if (!root || *s=='\0') return NULL;
  root->Lch = root->Rch = NULL; root->Count = 1; root->Elem = *s++;
  for(; *s != '\0'; s++) {
        (2)    ; parent = NULL;
    while (p) { /* p 从树根结点出发查找当前字符*s 所在结点 */
      parent = p;
      if (*s == p->Elem) /*若树中已存在当前字符结点，则当前字符的计数值加 1*/
        { p->Count++; break; }
      else /*否则根据字符*s 与结点*p 中字符的关系，进入*p 的左子树或右子树*/
        if (*s > p->Elem) p = p->Rch;
        else p = p->Lch;
    }/*while*/
    if (    (3)    ) { /* 若树中不存在字符值为*s 的结点，则申请结点并插入树中 */
      p = (BiTree)malloc(sizeof(struct BSTNode));
      if (!p) return NULL;
      p->Lch = p->Rch = NULL; p->Count = 1; p->Elem = *s;
      /*根据当前字符与其父结点字符值的大小关系，将新结点作为左子树或右子树插入*/
      if (p->Elem > parent->Elem)     (4)     = p;
      else     (5)     = p;
    }
  }/*for*/
  return root;
}
```

从下列的 3 道试题（试题六至试题八）中任选 1 道解答。  
如果解答的试题数超过 1 道，则题号小的 1 道解答有效。

### 试题六（共 15 分）

阅读以下说明和 C++ 代码，将解答写入答题纸的对应栏内。

#### [说明]

类 Stock 的定义中有三处错误，分别在代码的第 04、06、10 行。请补齐下述代码中的空缺 1，修改错误并给出修改后该行的完整代码，最后完善程序运行后的输出结果。

#### [C++ 代码]

```
01 #include <iostream>
02 using namespace std;
03 class Stock{
04 protected:
05     Stock() {shares=0; share_val=0.0; Output();}
06     Stock(int n=0, double pr=3.5):____(1)____ { //初始化 shares 值为 n
07         share_val=pr;
08         Output();
09     };
10     void ~Stock() {};
11     void Output() {cout << shares << ':' << share_val << endl;}
12 public:
13     //成员函数
14 private:
15     //成员变量
16     int shares;
17     double share_val;
18 };
19
20 void main() { //构造三个 Stock 对象 a, b, c
21     Stock a(1);
22     Stock b;
23     Stock c = Stock();
24     //其它代码省略，且代码无输出
25 }
```

程序运行后的输出结果为：

1:3.5

\_\_\_\_\_  
(2)

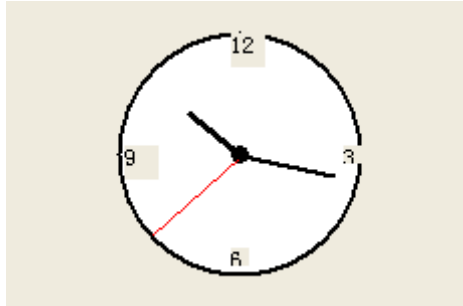
\_\_\_\_\_  
(3)

**试题七（15分）**

阅读以下应用说明以及用 Visual Basic 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。

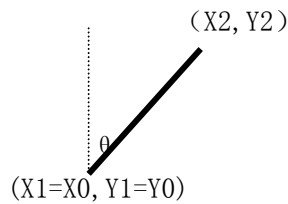
**[应用说明]**

本应用程序的运行窗口中将显示一个简单的模拟时钟如下：



该圆形钟面上有时针、分针和秒针在运动，不断显示系统的当前时间。

在开发该应用时，首先建立一个圆形 Shape 控件作为钟面（背景为白色）。设圆心为像素坐标  $(X0, Y0)$ ，半径为 940。再在圆心处建立一个小的实心圆 Shape 控件作为轴心。钟面上的刻度 3, 6, 9, 12 为四个标签。时针、分针、秒针（红色）为线型控件，名称分别为 LineH、LineM、LineS，线宽属性值分别为：3, 2, 1，线长属性值分别为 500, 700, 900。这三个线型控件的起点坐标都固定在  $(X0, Y0)$ ，其终点坐标随转角  $\theta$  而动态变化：



设置定时器 Timer1 使该时钟能反映系统的当前时间，其定时间隔为 0.5 秒，该定时器的 Interval 属性值应设置为     (1)    。每隔 0.5 秒，秒针需要调整位置，但分针和时针只是在初始显示时以及在每次秒计数为 0 时才需要调整位置（可节省计算量）。

**[程序代码]**

```
Const Pi = 3.1416
Const X0 = 1750
Const Y0 = 1150
Const LH = 500, LM = 700, LS = 900
Public first As Boolean
Private Sub Form_Load()
    LineS.X1 = X0 ' 秒针的起点位置
    LineS.Y1 = Y0
    LineM.X1 = X0 ' 分针的起点位置
    LineM.Y1 = Y0
    LineH.X1 = X0 ' 时针的起点位置
    LineH.Y1 = Y0
    first = True
End Sub

Private Sub Timer1_Timer()
    S = Second(Time)
    S_seta = Pi *           (2)           ' 秒针的转角
    LineS.X2 = X0 + LS * Sin(S_seta) ' 秒针的终点位置
    LineS.Y2 = Y0 - LS * Cos(S_seta)
    If (           (3)           ) Then
        M = Minute(Time)
        LineM.X2 = X0 + LM * Sin(Pi*M/30) ' 分针的终点位置
        LineM.Y2 = Y0 - LM * Cos(Pi*M/30)
        H = Hour(Time)
        If (H >= 12) Then H = H - 12
        H_seta = Pi * (           (4)           ) ' 时针的转角
        LineH.X2 = X0 + LH * Sin(H_seta) ' 时针的终点位置
        LineH.Y2 = Y0 - LH * Cos(H_seta)
        first =           (5)          
    End If
End Sub
```

### 试题八（共 15 分）

阅读以下说明和 Java 代码，将解答写入答题纸的对应栏内。

#### [说明]

已知类 Stock 和类 JavaMain 都定义在 JavaMain.java 文件中，类 Stock 的定义中有四处错误，分别在代码的第 01、02、06、07 行。请修改错误并给出修改后该行的完整代码，并写出改正错误后程序运行的输出结果。

#### [Java 代码]

```
01 public class Stock{
02     static {
03         shares = 0;
04         share_val=0.0;
05     }
06     private Stock() {getData();}
07     private Stock(int n, double pr=0) {
08         shares=n;
09         share_val=pr;
10         getData();
11     }
12     public void getData() {
13         System.out.print(shares + ":" + share_val + " ");
14     }
15     private int shares;          //非静态变量
16     private double share_val;    //非静态变量
17 };
18
19 public class JavaMain{
20     public static void main(String args[]) {
21         Stock a = new Stock();
22         Stock b = new Stock(1,67.5);
23         //其它代码省略，且代码无输出
24     }
25 }
```