

全国计算机技术与软件专业技术资格（水平）考试

2004 年下半年 程序员级 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卡

- 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

2004 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 6 日”，故在答题纸的对应栏内写上“11”和“6”（参看下表）。

例题	解答栏
(1)	11
(2)	6

试题一（15 分，每空 3 分）

阅读下列说明和流程图，将应填入 (n) 的字句写在答题纸的对应栏内。

[流程图说明]

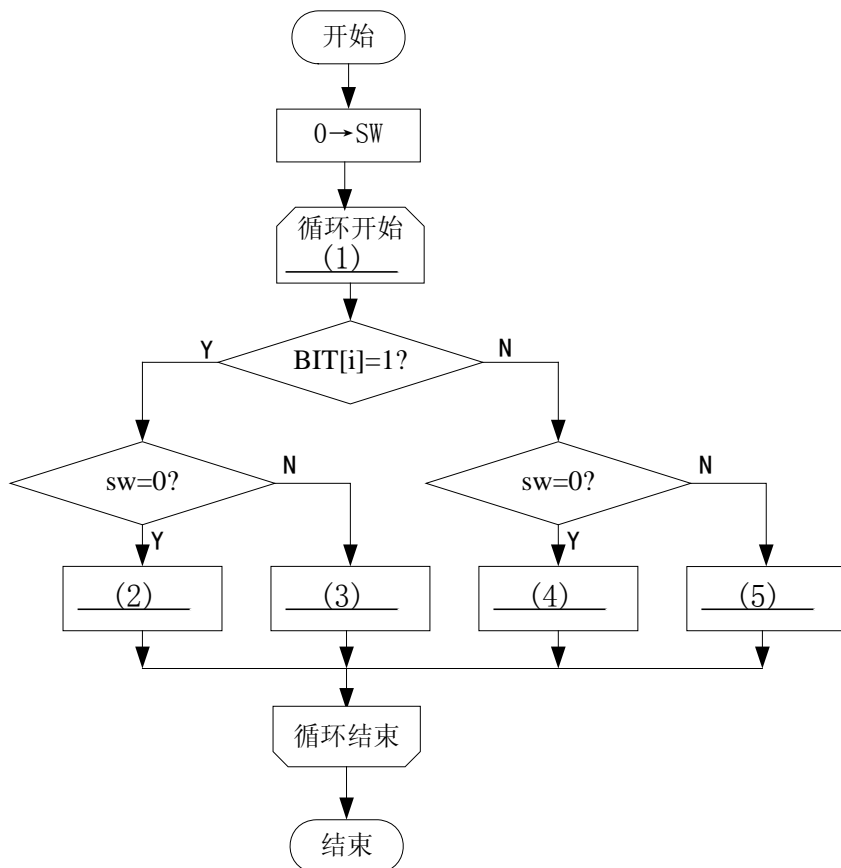
下面的流程图描述了对 8 位二进制整数求补的算法。

该算法的计算过程如下：从二进制数的低位（最右位）开始，依次向高位逐位查看，直到首次遇到“1”时，停止查看。然后，对该“1”位左面的更高位（如果有的话），逐位求反，所得的结果就是对原二进制数求补的结果。

例如：对二进制整数 10101000 求补的结果是 01011000。

设 8 位二进制整数中的各位，从低位到高位，依次存放在整型数组 BIT 的 BIT[1]～BIT[8] 中。例如，二进制整数 10101000 存放在数组 BIT 后，就有 BIT[1]=0, BIT[2]=0, ..., BIT[7]=0, BIT[8]=1。若流程图中存在空操作，则用 NOP 表示。

[流程图]



流程图中 (1) 处按“循环变量名：循环初值，增量，循环终值”格式描述。

试题二（15 分，每空 3 分）

阅读下列函数说明和C代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

[函数说明]

函数 sort(int a[], int n)的功能是对数组 a 中的 a[0]~a[n-1]这 n 个元素进行排序。排序过程如下：第一趟对所有的偶数下标 p，比较 a[p]和 a[p+1]，若 a[p]>a[p+1]，则将两者交换；第二趟对所有的奇数下标 q，比较 a[q]和 a[q+1]，若 a[q]>a[q+1]，则将两者交换；第三趟对偶数下标，第四趟对奇数下标，...，依次类推，直至整个数组元素有序排列为止。

[函数]

```
void sort(int a[],int n)
{
    int i,j,t,tag=1;
    for(j=0; j<n; j++) {
        for(i=  (1)  ; i<n-1 ;   (2)  ) {
            if (a[i]>a[i+1]) {t=a[i]; a[i]=a[i+1];a[i+1]=t; tag=0;}
        }
        tag++;
        if (  (3)  ) break;
    }
}
```

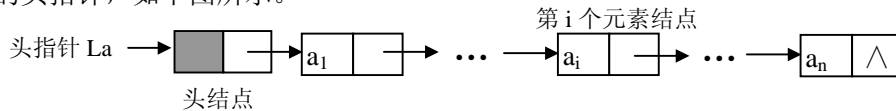
对包含十个元素的数组a采用上述方法排序时，元素间的比较次数最少为 (4) 次，最多为 (5) 次。

试题三（15 分，每空 3 分）

阅读下列函数说明和C代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

[函数说明]

函数 movetoend(LinkedList La, int i)的功能是：将线性表 A 的第 i 个元素移到表尾。若移动成功，则返回 0，否则返回-1。线性表 A 采用带头结点的单链表表示，La 为表 A 的头指针，如下图所示。



链表结点的类型定义为：

```
typedef struct node {  
    int key;  
    struct node *next;  
}*LinkedList;
```

[函数]

```
int movetoend(LinkedList La, int i)  
{  
    LinkedList p, q, prep;  
    int k=1;  
    p = La->next; prep = La;  
    while (   (1)   ) { /*查找第i个元素并使指针p指向该结点*/  
        prep = p; p = p->next; k++;  
    }  
    if (!p || k>i) return -1;  
    if (   (2)   ) /*第i个元素结点已经是表尾结点，则无需移动*/  
        return 0;  
    q=p;  
    while (   (3)   ) q = q->next; /*查找表尾并使q指向表尾结点*/  
      (4)   =p->next;  
    p->next = NULL;  
      (5)   ;  
    return 0;  
}
```

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（15 分，每空 3 分）

阅读以下说明和C代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

【说明】

某单位举办了一场知识竞赛，参加竞赛的选手为 300 名，依次从 1~300 进行编号。竞赛时间为 9:00~11:00。8 道竞赛题目依次从 ‘A’ ~ ‘H’ 编号，选手可按任意次序答题，每完成一道题目，可立即提交答案。若答案正确(Y)，则选择其他题目进行解答，否则，可继续做该题目或选择其他题目进行解答，直至竞赛结束。

选手提交答案的情况及判定结果由专人即时录入，录入的数据如下表 1 所示，对竞赛情况进行统计和排名的结果如表 2 所示。

表 1：输入数据			
提交时间	选手编号	题目编号	是否正确
09:37	14	A	N
09:50	8	A	Y
09:51	9	B	N
09:52	14	A	N
10:14	5	A	N
10:16	9	B	N
10:23	5	A	Y
10:23	8	B	Y
10:27	14	A	Y

表 2：输出结果			
名次	选手编号	完成的 题目数量	总用时 (分钟)
1	8	2	133
2	5	1	103
3	14	1	127

统计和排名的规则如下：

1. 若选手 X 在竞赛时提交的题目 P 解答正确，则解答该题目所用时间如下计算：
解答题目 P 的用时 = 提交题目 P 正确的时间 — 竞赛的开始时间 + 罚时
罚时 = 提交题目 P 错误解答的次数 × 20

例如：表 1 中 14 号选手在 10:27 提交了题目 A 的正确解答，因此该选手正确解答该题目所用时间为 87 分钟，由于之前的两次提交错误解答，罚时为 $2 \times 20 = 40$ 分钟，所以 14 号选手解答题目 A 的用时 = $87 + 40 = 127$ （分钟）。

2. 已经提交正确答案的题目再次提交时不再计算。

3. 竞赛结束时，选手的总用时为所有解答正确的题目用时累加所得，解答不正确的题目不计时。

4. 排名时，完成题目数量多者排名靠前；若完成的题目数相同，则用时少者排名靠前；若完成的题目数和所用时间均相等，则名次相同；完成题目数为 0 的选手不参加排名。

函数 void Statistic() 的功能是：读取输入数据，进行统计、排名并输出结果。

```
#define MAXN 300
typedef struct {
    int no; /*选手编号*/
    int num; /*完成的题目数量*/
    int time; /*完成题目的总用时*/
    int d[8]; /*d[i]用于记录提交第 i 个题目错误答案的次数*/
    int a[8]; /*a[i]用于记录第 i 个题目是否已经提交正确答案*/
} Info;
【函数】
void Statistic() {
    char ch, pass;
    int i, j, k, h, m, t, time, MaxIndex;
    Info R[MAXN+1];
    for(i=1; i<=MAXN; i++) { /*数组 R 的元素置初值 0*/
        R[i].no = 0; R[i].num = 0; R[i].time = 0;
        for(j=0; j<8; j++) {R[i].d[j] = 0; R[i].a[j] = 0;}
    } /*for*/
    MaxIndex = 0;
    while (1) {
        /*录入一名选手提交答案的信息（小时:分钟, 选手编号, 题目号, 是否正确）*/
        scanf("%d:%d,%d,%c,%c", &h, &m, &k, &ch, &pass);
        if (h == 0) break;
        R[k].no = k; /*k 为选手编号*/
        time = (1); /*计算答题时间, 以分钟为单位*/
        if (isupper(ch)) ch = 'a' + ch - 'A';
        if (pass != 'Y' && pass != 'y') {R[k].d[ch-'a']++; continue;}
        if (R[k].a[ch-'a'] == 1) continue;
        R[k].a[ch-'a'] = 1;
        R[k].num++;
        R[k].time += (2);
        if (k > MaxIndex) MaxIndex = k;
    } /*while*/
    for(i=1; i<MaxIndex; i++) { /*选择排序*/
        for(t=i, j=i+1; j<=MaxIndex; j++)
            if (R[t].num<R[j].num || (3) t = j;
        if ((4) {R[0] = R[t]; R[t] = R[i]; R[i] = R[0];}
    } /*for*/
    k=1; R[0] = R[1];
    for(i=1; i<=MaxIndex; i++) /*输出排名情况*/
        if (R[i].num > 0) {
            if (R[i].num!=R[0].num || R[i].time!=R[0].time) k++;
            R[0] = (5);
            printf("%d: %3d %4d %5d\n", k, R[i].no, R[i].num, R[i].time);
        } /*if*/
    } /*Statistic*/
```

试题五（15 分，每空 3 分）

阅读以下应用说明及 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。

[应用说明]

某单位举办了一场知识竞赛，有 300 名选手参加竞赛，依次从 1~300 为选手编号。竞赛时间为 9:00~11:00。8 道竞赛题目依次从“A”~“H”编号，选手可按任意次序答题，每完成一道题目，可立即提交答案。若答案正确(Y)，则选择其他题目进行解答，否则，可继续做该题目或选择其他题目进行解答，直至竞赛结束。

选手提交答案的情况及判定结果由专人即时录入，录入数据包括提交答案的时间、选手编号、题目编号(A~H)、是否正确(Y/N)。

对竞赛情况进行统计和排名的规则如下：

1. 若选手 X 在竞赛时提交的题目 P 解答正确，则解答该题目所用时间如下计算：

解答题目 P 的用时 = 提交题目 P 正确的时间 - 竞赛的开始时间 + 罚时

罚时 = 提交题目 P 错误解答的次数 × 20

例如：设 14 号选手在 10:27 提交了题目 A 的正确解答，则该选手正确解答该题目所用时间为 87 分钟，由于他在 09:37 和 09:52 两次提交了题目 A 的错误解答，因此罚时为 $2 \times 20 = 40$ 分钟，所以 14 号选手解答题目 A 的用时 = $87 + 40 = 127$ （分钟）。

2. 已经提交正确答案的题目再次提交时不作计算。

3. 竞赛结束时，选手的总用时为所有解答正确的题目用时累加所得，解答不正确的题目不计。

4. 排名时，完成题目数量多者排名靠前；若完成的题目数相同，则用时少者排名靠前；若完成的题目数和所用时间均相等，则名次相同；完成题目数为 0 的选手不参加排名。

本应用程序的运行窗口如下图所示：

提交时间	选手编号	题目编号	是否正确
09:52	14	A	N
10:14	5	A	N
10:16	9	B	N
10:23	5	A	Y
10:23	8	B	Y
10:27	14	A	Y

名次	选手编号	题目数量	总用时
1	8	2	133
2	5	1	103
3	14	1	127

窗口中的两个文本框为 Txt_time 和 Txt_player，分别用于录入提交答案的时间和选手编号。组合列表框 Combo1 提供题目编号(A~H)，录入时从中选择。检查框 Chk_yn 用于输入解答是否正确信息。当单击“确定”按钮(Cmd_confirm)时，录入的提交信息加入列表框 List1 中，排名情况在列表框 List2 输出。单击“关闭”按钮时退出应用程序。

在开发过程中，需要编写的部分程序代码如下：

[程序代码]

```
Private Type Info
    no As Integer          ' 选手编号
    num As Integer         ' 完成的题目数量
    time As Integer        ' 完成题目的总用时
    d(8) As Integer        ' d[i]用于记录提交第 i 个题目错误答案的次数
    a(8) As Boolean        ' a[i]用于记录第 i 个题目是否已经提交正确答案
End Type
Dim R(301) As Info        ' R[j]用于统计编号为 j 的选手提交答案的情况
Dim MaxIndex As Integer   ' MaxIndex 记录提交答案的选手中编号最大者

Private Sub Form_Load()
    For i = 1 To 8
        Combol.AddItem Chr( (1) )
    Next
    Combol.Text = Combol.List(0): Txt_time.Text = "": Txt_player.Text = ""
    For i = 1 To 300
        R(i).num = 0: R(i).time = 0: R(i).no = i
        For j = 1 To 8
            R(i).d(j) = 0: R(i).a(j) = False
        Next j, i
    End Sub

Private Sub Cmd_confirm_Click()
    Dim h, m, k, time As Integer, ch, pass, s1 As String
    k = InStr(Txt_time.Text, ":"): If k < 2 Then GoTo error1
    h = Val(Left(Txt_time.Text, k - 1)): m = Val(Mid(Txt_time.Text, k + 1))
    If h > 11 Or h = 11 And m > 0 Or m >= 60 Then GoTo error1
    time = (2) ' 计算答题时间,以分钟为单位
    If Txt_player.Text < 1 Or Txt_player.Text > 300 Then GoTo error1
    ch = (3) ' 取题目编号
    pass = IIf(Chk_yn.(4) = 0, "N", "Y")
    s1 = Txt_time.Text + Space(4) + Txt_player.Text
    s1 = s1 + Space(10 - Len(Txt_player.Text)) + ch + Space(8) + pass
    List1.AddItem s1
    k = Val(Txt_player.Text) ' k 为选手编号
    R(k).no = k ' 编号为 k 的选手的提交信息记录在下标为 k 的数组元素中
    If k > MaxIndex Then MaxIndex = k
    m = Asc(ch) - Asc("A")
    If pass <> "Y" Then ' 编号为 k 的选手提交第 m 个题目的解答不正确
        R(k).d(m) = R(k).d(m) + 1
    ElseIf R(k).a(m) <> True Then ' 已经提交正确的题目的解答不再计算
        R(k).a(m) = True: R(k).num = R(k).num + 1
        R(k).time = R(k).time + (5)
        Call Statistic ' 调用过程 Statistic 进行实时排名和输出
    End If
    Exit Sub
error1: MsgBox "录入信息有误!", vbOKOnly
End Sub
```


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（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】

以下程序的功能是计算三角形、矩形和正方形的面积并输出。

程序由 4 个类组成：类 Triangle、Rectangle 和 Square 分别表示三角形、矩形和正方形；抽象类 Figure 提供了一个纯虚拟函数 getArea()，作为计算上述三种图形面积的通用接口。

【C++程序】

```
#include <iostream.h>
#include <math.h>

class Figure {
public:
    virtual double getArea() = 0;           // 纯虚拟函数
};

class Rectangle :_(1)_ {
protected:
    double height;
    double width;
public:
    Rectangle() {} ;
    Rectangle(double height, double width) {
        this->height = height;
        this->width = width;
    }
    double getArea() {
        return  _(2)_ ;
    }
};

class Square :_(3)_ {
public:
    Square(double width) {
        _(4)_ ;
    }
};
```

```

    }
};

class Triangle :____(5)____ {
    double la;
    double lb;
    double lc;
public:
    Triangle(double la, double lb, double lc) {
        this->la = la;    this->lb = lb;    this->lc = lc;
    }
    double getArea() {
        double s = (la+lb+lc)/2.0;
        return sqrt(s*(s-la)*(s-lb)*(s-lc));
    }
};

void main() {
    Figure* figures[3] = {
        new Triangle(2,3,3), new Rectangle(5,8), new Square(5) };
    for (int i = 0; i < 3; i++) {
        cout << "figures[" << i << "]" area = " << (figures[i])->getArea() << endl;
    }
}

```

试题七（15 分，每空 3 分）

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

[应用说明]

设一个简单的“通讯录”存储在一个 ACCESS 类型的数据库表中，包括姓名、电话和 email 三个字段。下面的应用程序实现对“通讯录”数据库表中的记录进行增加、删除及修改处理，其运行界面如下：

(1) 数据控件(data1)与“通讯录”数据库表相连接，用户可通过“▶”和“◀”按钮指定表中的当前记录。

(2) 文本框 Txt_name、Txt_phone 和 Txt_email 分别与数据库表中的“姓名”、“电话”和“email”字段绑定，用于显示当前记录的内容。

(3) 应用程序启动时，“确定”按钮(Cmd_ok)和“取消”按钮(Cmd_cancel)不可操作，文本框中显示表中的第一条记录，这时文本框处于不可操作状态。

(4) 单击“增加”按钮(Cmd_add)或“修改”按钮(Cmd_modify)后，方可以编辑记录内容，同时“增加”、“删除”、“修改”和“退出”按钮变为不可操作状态。“增加”和“修改”操作需通过“确定”和“取消”按钮确认。

(5) 单击“删除”按钮(Cmd_del)后，弹出对话框，再单击“确定”按钮，当前记录被删除。

[程序代码]

```
Private Sub enableop(isEnabled As Boolean)
    Txt_name.Enabled = isEnabled: Txt_phone.Enabled = isEnabled
    Txt_email.Enabled = isEnabled
    Cmd_ok.Enabled = isEnabled: Cmd_cancel.Enabled = isEnabled
    Cmd_add.Enabled = Not isEnabled: Cmd_del.Enabled = Not isEnabled
    Cmd_end.Enabled = Not isEnabled: Cmd_modify.Enabled = Not isEnabled
End Sub
```

```

Private Sub Form_Load()
    Call enableop(False)
    Data1.Refresh
    If Data1.Recordset.RecordCount = 0 Then
        Cmd_del.Enabled = False:    Cmd_modify.Enabled = ____ (1) ____
    End If
End Sub

Private Sub Cmd_add_Click()        ' 单击“增加”按钮的代码
    Call enableop(____ (2) ____ )
    Data1.Recordset.AddNew        ' 在数据库表中添加一个新记录
    Txt_name. ____ (3) ____
End Sub

Private Sub Cmd_del_Click()        ' 单击“删除”按钮的代码
    On Error GoTo error3
    ans = MsgBox("确定删除吗?", vbYesNo + vbInformation, "操作提示!")
    If (ans = vbYes) Then
        Data1.Recordset. ____ (4) ____
        Data1.Recordset.MoveNext
        If Data1.Recordset.EOF Then Data1.Refresh
    End If
    Exit Sub
error3:
    MsgBox Err.Description, vbOKOnly, "错误提示!"
End Sub

Private Sub Cmd_ok_Click()        ' 单击“确定”按钮的代码
    On Error GoTo error1
    Data1.Recordset. ____ (5) ____
    Call enableop(False)
    Exit Sub
error1:
    MsgBox Err.Description, vbOKOnly, "错误提示!"
    Data1.UpdateControls
    Call enableop(False)
End Sub
'    “修改”和“取消”按钮的程序代码(略)

```

试题八（15分，每空3分）

阅读以下说明和Java源程序，将应填入(n)处的字句写在答题纸的对应栏内。

【说明】

以下程序的功能是计算三角形、矩形和正方形的面积并输出。

程序由5个类组成：AreaTest 是主类，类 Triangle、Rectangle 和 Square 分别表示三角形、矩形和正方形，抽象类 Figure 提供了一个计算面积的抽象方法。

【程序】

```
public class AreaTest{
    public static void main(String args[]) {
        Figure[] figures={
            new Triangle(2,3,3), new Rectangle(5,8), new Square(5)
        };
        for (int i=0; i<figures.length;i++) {
            System.out.println(figures[i]+"area="+figures[i].getArea());
        }
    }
}

public abstract class Figure {
    public abstract double getArea();
}

public class Rectangle extends (1) {
    double height;
    double width;
    public Rectangle(double height,double width){
        this.height=height;
        this.width=width;
    }
    public String toString(){
        return "Rectangle:height="+height+",width="+width+": ";
    }
    public double getArea(){
        return (2);
    }
}

public class Square extends (3) {
```

```

    public Square(double width) {
        (4);
    }
    public String toString() {
        return "Square:width="+width+":";
    }
}

public class Triangle extends (5) {
    double la;
    double lb;
    double lc;
    public Triangle(double la,double lb,double lc) {
        this.la=la;    this.lb=lb;    this.lc=lc;
    }
    public String toString(){
        return "Triangle:sides="+la+", "+lb+", "+lc+":";
    }
    public double getArea() {
        double s=(la+lb+lc)/2.0;
        return Math.sqrt(s*(s-la)*(s-lb)*(s-lc));
    }
}

```