

全国计算机技术与软件专业技术资格（水平）考试

2006 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

2006 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 20 日”，故在答题纸的对应栏内写上“5”和“20”（参看下表）。

例题	解答栏
(1)	5
(2)	20

试题一（共 15 分）

阅读以下说明和流程图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

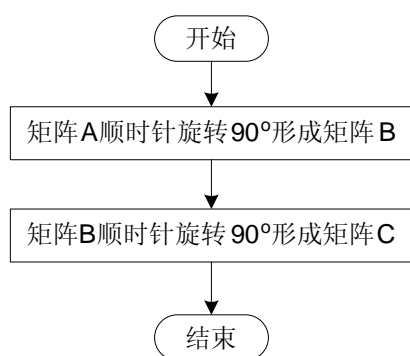
【说明】

信息处理过程中经常需要将图片或汉字点阵做旋转处理。一个矩阵以顺时针方向旋转 90° 后可以形成另一个矩阵，如下图所示：

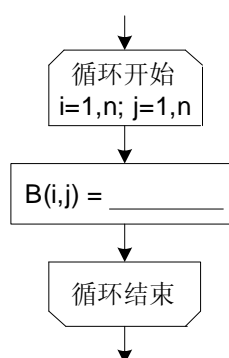
A	B	C	D		M	I	E	A
E	F	G	H		N	J	F	B
I	J	K	L		O	K	G	C
M	N	O	P		P	L	H	D

流程图 1-1 描述了对 $n \times n$ 矩阵的某种处理。流程图 1-2 是将矩阵 A 顺时针旋转 90° 形成矩阵 B 的具体算法。

【流程图 1-1】



【流程图 1-2】



【问题 1】（3 分） 请写出以下 3×3 单位矩阵沿顺时针方向旋转 90° 后所形成的矩阵。

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

【问题 2】（3 分） 如果以下 3×3 矩阵沿顺时针方向旋转 90° 后所形成的矩阵就是原来的矩阵：

$$\begin{pmatrix} a & b & * \\ * & c & * \\ * & * & * \end{pmatrix} \quad \text{其中，位于*处的元素需要考生填写}$$

请完整地写出该矩阵。

【问题 3】（9 分） 在上述流程图 1-1 和 1-2 的算法中，

- (1) 矩阵 A 第 i 行第 j 列的元素 $A(i, j)$ 被复制到矩阵 B 中的哪个位置？
- (2) $A(i, j)$ 后来又被复制到矩阵 C 中的哪个位置？
- (3) 填补流程图 1-2 中的空缺。

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 `change(int num)` 的功能是对四位以内（含四位）的十进制正整数 `num` 进行如下的变换：将 `num` 的每一位数字重复一次，并返回变换结果。例如，若 `num=5234`，则函数的返回值为 55223344，其变换过程可描述为：

$$(4*10+4)*1 + (3*10+3)*100 + (2*10+2)*10000 + (5*10+5)*1000000 = 55223344$$

【C 语言函数】

```
long change(int num)
{
    int d, m = num;
    long result, mul;

    if (num <= 0 ||           (1)          ) /* 若 num 不大于 0 或 num 的位数大于 4，则返回-1*/
        return -1;

    mul = 1;
              (2)          ;
    while (m > 0) {
        d = m % 10;
        m =           (3)          ;
        result = result + (          (4)          ) * mul;
        mul =           (5)          ;
    }
    return result;
}
```

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 `bool Del_elem(STACK *s, char para_ch)` 的功能是：删除栈 *s 中与 `para_ch` 之值相等且最接近栈顶的元素（字符），若栈中不存在该元素，则函数返回 `FALSE`，否则返回 `TRUE`。其中，`STACK` 是栈的类型名。

函数 `Del_elem` 实现上述功能的方法是：利用栈的基本操作，先将栈 *s 中所有比 `para_ch` 之值更接近栈顶的元素暂时存放在临时工作栈 `s_bak` 中，使得与 `para_ch` 之值相等的元素成为栈顶元素，此时执行出栈操作，即从栈中删除与 `para_ch` 之值相等的元素，最后再将 `s_bak` 中的元素依次存回栈 *s。

在函数 `Del_elem` 中必须使用栈的基本操作进行栈上的运算，实现栈的基本操作的函数原型说明如下：

`void InitStack(STACK *S)`：初始化栈。

`void Push(STACK *S, char e)`：将一个字符压栈，栈中元素数目增 1。

`void Pop(STACK *S)`：栈顶元素出栈，栈中元素数目减 1。

`char Top(STACK S)`：返回非空栈的栈顶元素值，栈中元素数目不变。

`bool IsEmpty(STACK S)`：若 S 是空栈，则返回 `TRUE`；否则返回 `FALSE`。

`bool` 类型定义如下：

```
typedef enum {FALSE = 0, TRUE = 1} bool;
```

【C 语言函数】

```
bool Del_elem(STACK *s, char para_ch)
{
    STACK s_bak;          /* 定义临时工作栈 s_bak */
    char ch;
    bool tag = FALSE;

    ____ (1) ____;        /* 初始化临时工作栈 s_bak */

    /* 将栈*s 中所有比 para_ch 更接近栈顶的元素暂时存放在临时工作栈 s_bak 中 */
    while( !IsEmpty(*s) ) {
        ch = ____ (2) ____; /* 取栈顶元素 */
        Pop(s);
        if (ch == para_ch) {
            tag = TRUE;
            break;
        }
        ____ (3) ____;
    }

    /* 将暂存于临时工作栈 s_bak 中的元素存回栈*s */
    while ( ____ (4) ____ ) {
        ch = Top(s_bak);
        ____ (5) ____;
        Push(s, ch);
    }

    return tag;
}
```

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】

某工厂 A 负责为某大型企业 B 加工零件，A 每天必须为 B 提供一定数量的零件。由于某种客观原因，A 每天生产的零件的单价都不相同。若 A 某天生产的零件数多于 B 需要的数目，则多余的零件可以放到第二天及以后再使用，但需要收取每个零件的保管费（产品单价之外附加的费用），每个零件在不同日期收取的保管费也不相同。

例如，在 5 天的生产中，B 要求的零件需求量及 A 核算出的零件单价和保管费用如表 1 所示：

表 1

	第一天	第二天	第三天	第四天	第五天
需求量（个）	25	15	30	35	30
零件的单价（元）	20	30	32	25	35
单个零件的保管费用（元）	4.5	8.0	6.8	5.5	6.1

A 可以制订多种生产计划，但费用可能不同。例如，表 2 所示为生产计划及其费用。

表 2

	计划产量（个）					总费用 (元)
	第一天	第二天	第三天	第四天	第五天	
计划 1	25	15	30	35	30	3835
计划 2	40	0	30	50	15	3685
计划 3	70	0	0	65	0	3632.5
计划 4	35	10	20	40	35	/

注：（1）计划 1 的总费用： $25*20+15*30+30*32+35*25+30*35=3835$ （元）

（2）计划 2 的总费用： $40*20+15*4.5+30*32+50*25+15*5.5+15*35=3685$ （元）

（3）计划 3 的总费用： $70*20+45*4.5+30*8+65*25+30*5.5=3632.5$ （元）

（4）计划 4 不可行，虽然第一天和第二天生产的零件总数比需求量多 5 个，但加上第三天生产的 20 个零件（共 25 个），仍不能满足 B 第三天的需求量(30 个)。

函数 find_a_plan(FILE *in)的功能是：从文件中读入若干个生产计划，从可行的计划中选出费用最小者，记录该生产计划并返回该最小费用。

全局结构体数组 data[]用于保存表 1 所示的数据(data[0]不用)，说明如下：

data[i].Qty_req: int 型，表示第 i 天的零件需求量。

data[i].Price: double 型，表示第 i 天生产的零件单价(元)。

data[i].Keeping_fee: double 型，表示第 i 天保管单个零件的费用(元)。

【C 语言函数】

```
int B_s[DAYS+1];/* 记录成本最小的生产计划, B_s[0]不用, DAYS 定义为天数*/
double find_a_plan(FILE *inf)
{   int P_num[DAYS+1], acc_req[DAYS+1];
    int i, tag = 0, acc_qty = 0;
    double mincost = 1.0e20, cost_Produce, cost_Keep;

    for(i = 1; i <= DAYS; i++) { /* 到第 i 天时的累计零件需求量存入 acc_req[i] */
        acc_qty += data[i].Qty_req;
        acc_req[i] = acc_qty;
    }

    while (!feof(inf)) {
        for(i = 1; i <= DAYS; i++) /*读入一个生产计划, 第 i 天的产量存入 P_num[i]*/
            if (!feof(inf))
                fscanf(inf, "%d", &P_num[i]);

        tag = 0; cost_Produce = 0; cost_Keep = 0;
        for(i = 1, (1); i <= DAYS; i++){ /* 考察当前的生产计划*/
            acc_qty += P_num[i]; /* acc_qty 记录到第 i 天时的累计零件生产量 */
            if (acc_qty < acc_req[i]) { /* 当前生产计划不能满足需求 */
                tag = 1; break;
            } /*if*/
            cost_Produce += (2); /*计算当前生成计划的总零件价格*/
            /*计算当前生成计划下的零件保管费*/
            cost_Keep += ((3)) * data[i].Keeping_fee;
        }/*for*/
        if ((4)) /* 若当前生产计划不可行, 则继续读取下一计划 */
            continue;
        if ((5)) { /*记录成本更小的生产计划*/
            mincost = cost_Produce + cost_Keep;
            for(i = 1; i <= DAYS; i++)
                B_s[i] = P_num[i];
        }/*if*/
    }/*while*/
    return mincost;
}
```

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】

某应用程序在运行过程中的一个界面如下图所示：



在“供选课程”列表框（名为 List1）中选择某个课程后，再单击“>”按钮（名称为 cmdSelect），就能将该课程复制到“所选课程”列表框（名称为 List2）中（但要注意不要出现重复项）；如果单击“>>”按钮（名称为 cmdSelAll），就能将全部供选课程名复制到列表框 List2 中（注意应先删除 List2 中的已选项）；如果在“所选课程”列表框中选中某个课程，再单击“<”按钮（名称为 cmdDelete），就会从 List2 中删除该课程（未选课程时应不做处理）；如果单击“<<”按钮（名称为 cmdDelAll），就会从 List2 中删除所有已选课程。

【Visual Basic 程序代码】

```
Private Sub CmdSelect_Click()  
    For i = 0 To List2.ListCount - 1  
        If List1.Text = List2.List(i) Then Exit Sub    ‘不要重复选择  
    Next i  
    List2.AddItem ____ (1) ____    ‘在 List2 中增添 List1 中所选的项  
End Sub  
Private Sub CmdSelAll_Click()  
    List2.Clear    ‘先删除 List2 中的已有项  
    For i = 0 To ____ (2) ____    ‘对 List1 中所有各项做循环处理  
        List2.AddItem ____ (3) ____    ‘将该项增添到 List2 中  
    Next i  
End Sub  
Private Sub CmdDelete_Click()  
    If List2.ListIndex >= 0 Then    ‘如果 List2 中有选中的项  
        List2.RemoveItem ____ (4) ____    ‘则删除所选的项  
    End If  
End Sub  
Private Sub CmdDelAll_Click()  
    ____ (5) ____  
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（共 15 分）

阅读以下说明和 C++ 代码，将解答写在答题纸的对应栏内。

【说明】

已知类 SubClass 的 getSum 方法返回其父类成员与类 SubClass 成员 j 的和，类 SuperClass 中的 getSum 为纯虚拟函数。程序中的第 23 行有错误，请修改该错误并给出修改后的完整结果，然后完善程序中的空缺，分析程序运行到第 15 行且尚未执行第 15 行的语句时成员变量 j 的值，最后给出程序运行后的输出结果。

【C++ 代码】

行号	代码
01	#include <iostream>
02	using namespace std;
03	class SuperClass{
04	private:
05	int i;
06	public:
07	SuperClass() {i = 5;}
08	virtual int getValue() {return i;}
09	virtual int getSum()=0;
10	};
11	class SubClass:public SuperClass{
12	int j;
13	public:
14	SubClass(int j):j(0){
15	____(1)____ = j; //用参数 j 的值更新数据成员
16	}
17	int getValue() {return j;}
18	int getSum() {
19	return ____ (2) ____ getValue() + j;
20	}
21	};
22	void main(void) {
23	SuperClass s = new SubClass(-3);
24	cout << s->getValue() << " ";
25	cout << s->getSum() << endl;
26	delete s;
27	}

试题七（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】

启动某应用程序后，其运行窗口左边会出现两只蝴蝶，下边有“启动”和“停止”两个按钮。单击“启动”按钮后，伴随“两只蝴蝶”音乐，两只蝴蝶会分别沿两条互绞的曲线，自左至右翩翩飞行（见下图），而且飞出窗口右边后又会再从窗口左边飞入。单击“停止”按钮时，两只蝴蝶停止飞行，回到初始状态，音乐也同时停止。再单击“启动”按钮时，蝴蝶重新开始伴音飞行。

在开发该应用时，两只蝴蝶用两个图像框来表示，分别命名为 Image1 和 Image2，其 Picture 属性分别设置为各个蝴蝶的图像文件名，其 stretch 属性都应设置为 (1)，使图像能自动改变大小以适应图像框。再在窗体中建立“启动”按钮（名为 CmdStart）和“停止”按钮（名为 CmdStop）。

蝴蝶所飞行的路线分别用正弦和余弦曲线描述。为此，首先需要用 Scal 方法对窗体定义坐标系统。设左上角坐标为 (0, 0)，右下角坐标为 (20, 6)。蝴蝶 1 的飞行曲线设为： $y_1 = 2 + \sin x$ ，蝴蝶 2 的飞行曲线设为： $y_2 = 2 + \cos x$ 。蝴蝶飞行的位置坐标 (x, y) 确定了蝴蝶图像框的位置 (Left 属性值和 Top 属性值)。

设置一个定时器（名为 timer1），其定时时间间隔为 0.1 秒，所以其 Interval 属性值应设置为 100。每隔 0.1 秒需要对两个图像框的位置进行一次调整。初始时，该定时器的 Enabled 属性应设为 False（关闭状态）。

为使该应用程序运行时能播放声音文件，在开发时应利用菜单“工程-部件”，选择加载组件“Microsoft Multimedia Control 6.0”，并在窗体中设置多媒体控件（设命名为 MMC）。由于不需要用户控制，其 Visible 属性可设置为 False（不可见）。右击该控件，可选择该控件中各按钮的属性，应注意选择“播放”有效、“停止”有效。“两只蝴蝶”的音乐文件应提前做好，并通过某种音频处理程序将其转换成 WAV 格式，存放在开发该应用的当前目录中。



【Visual Basic 程序代码】	
Dim <u> (2) </u>	‘声明全局变量
Private Sub Form_Load()	
Scale (0, 0)-(20, 6)	‘定义窗体的坐标系统
End Sub	
Private Sub CmdStart_Click()	
x = 0	
MMC.DeviceType = "WaveAudio"	‘设置多媒体设备类型
MMC.FileName = "两只蝴蝶.wav"	‘确定声音文件名
MMC.Command = "Open"	‘打开多媒体设备
MMC.Command = "Play"	‘启动声音的播放
Timer1.Enabled = True	‘打开定时器
End Sub	
Private Sub Timer1_Timer()	
If x > 20 Then x = x - 20	‘蝴蝶飞到右边界后再
Image1.Left = x	‘动态调整蝴蝶框的位
Image1.Top = <u> (3) </u>	
Image2.Left = x	
Image2.Top = <u> (4) </u>	
x = x + 0.1	‘确定蝴蝶下一步飞行
End Sub	
Private Sub CmdStop_Click()	
<u> (5) </u>	
Image1.Left = 0	‘蝴蝶位置初始化
Image1.Top = 2	
Image2.Left = 0	
Image2.Top = 3	
MMC.Command = "Stop"	‘停止播放
MMC.Command = "Close"	‘关闭多媒体设备
End Sub	

试题八（共 15 分）

阅读以下说明和 Java 代码，将解答写在答题纸的对应栏内。

【说明】

已知类 SubClass 的 getSum 方法返回其父类成员 i 与类 SubClass 成员 j 的和，类 SuperClass 中的 getSum 为抽象函数，程序中的第 14 行有错误，请修改该错误并给出修改后的完整结果，然后完善程序中的空缺，当程序运行到第 22 行且尚未执行第 22 行语句时成员变量 i 的值，最后给出程序运行后的输出结果。

【Java 代码】

行号	代码
01	public class MainJava {
02	public static void main(String[] args) {
03	SuperClass s = new SubClass();
04	System.out.println(s.getValue());
05	System.out.println(s.getSum());
06	}
07	}
08	abstract class SuperClass{
09	private int i;
10	public SuperClass() {i = 5;}
11	public int getValue() {
12	return i;
13	}
14	public final abstract int getSum();
15	}
16	class SubClass extends SuperClass{
17	int j;
18	public SubClass() {
19	this(-3);
20	}
21	public SubClass(int j){
22	_____(1)_____.j = j;
23	}
24	public int getValue() {return j;}
25	public int getSum() {
26	return _____(2)_____.getValue() + j;
27	}
28	}