**全国计算机技术与软件专业技术资格（水平）考试**

**2011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6道题，试题一至试题四是必答题，试题五至试题六选答 1 道。每

题 15 分，满分 75 分。

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**

2011 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）。

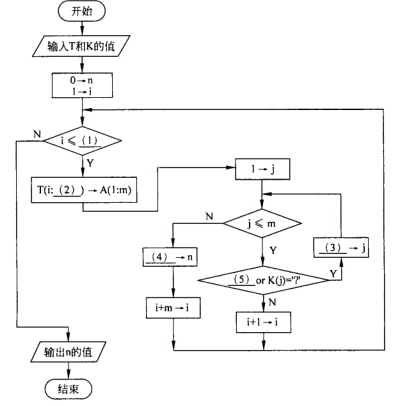
|  |  |
| --- | --- |
| 例题 | 解答栏 |
| （1） | 5 |
| （2） | 20 |

**试题一**

下面的流程图可在正文字符串T(1:L)中计算关键词字符串K(l:m)出现的次数（用n表示)。其中，L为字符串T的长度，m为字符串K的长度(m<L)„为便于模糊查找，关键词中的字符“？”可以匹配任意一个字符。

在该流程图中，先从T中取出长度为m的子串存入A中，再将A与K进行逐个字符的比较（其中，K可以包含字符“？ ”)。注意：从正文字符串中取出的关键词字符串不允许交叉。例如，“aaaaaa”中有3个关键词字符串“aa”。

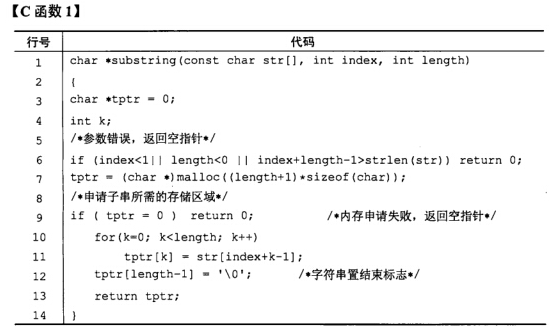
**【流程图】**



阅读以下说明和流程图，填补流程图中的空缺（1)〜（5)，将解答填入答题纸的对应栏内。

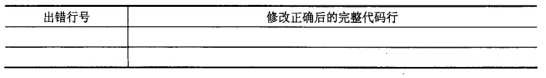
**试题二**

函数 substring(const char str[], int index, int length)的功能是求出字符串 str 中指定序号index开始且长度为length的子串，并返回所取出的子串。以字符串“China today” 为例，其第一个字符“C”的序号为1 (而其在字符数组str中的下标为0)，从序号5开始且长度为3的子串为“at”。



**【问题1】**

函数substring中有两处错误，请指出这些错误所在代码的行号，并在不增加和删除代码行的情况下进行修改，写出修改正确后的完整代码行（有注释时，注释可省略)。



**【问题2】**

请根据说明2,填充C函数2中的空缺（1)和（2)。

**【问题3】**

请说明以62354879643作为实参调用函数reverse时返回结果出错的原因。

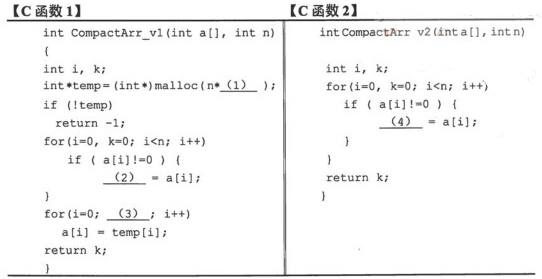
**试题三**

对于具有n个元素的整型数组a，需要进行的处理是删除a中所有值为0的数组元素，并将a中所有非0元素按照原顺序连续地存储在数组空间的前端。

下面分别用函数CompactArr\_vl和CompactArr\_v2来实现上述处理要求，函数的返回值为非零元素的个数。

函数CompactArr\_vl(int a[], intn)的处理思路是：首先申请一个与数组a的大小相同的动态数组空间，然后顺序扫描数组.a的每一个元素，将遇到的非0元素依次复制到动态数组空间中，最后再将动态数组中的元素传回数组a中。

函数CompaetArr\_v2(int a[], intn)的处理思路是：利用下标i (初值为0)顺序扫描数组a的每一个元素，下标k (初值为0)表示数组a中连续存储的非0元素的下标。扫描时，每遇到一个数组元素，i就增1，而遇到非0元素并将其前移后k才增1。



**【问题1】**

请根据说明中函数CompactArr\_vl的处理思路填补空缺(1)〜(3)，根据CompactArr\_ v2的处理思路填补空缺（4)。

**【问题2】**

请说明函数CompactArr\_vl存在的缺点。

**试题四**

假设一个算术表达式中可以包含以下三种括号：“(”和“)”、“[”和“]”及和 “}”，并且这三种括号可以按照任意的次序嵌套使用。

下面仅考虑表达式中括号的匹配关系，其他问题暂时忽略。例如，表达式 [a-(b-5)]\*c[{}]中的括号是完全匹配的，而表达式[a-(b-5]))\*c中的括号不是完全匹配的， 因为“(”与“]”不能匹配，而且多了一个“)”，即缺少一个与“)”相匹配的“(”。

函数ifMatched (char expr[])的功能是用栈来判断表达式中的括号是否匹配，表达式 以字符串的形式存储在字符数组expr中。若表达式中的括号完全匹配，则该函数的返回 值为Matched,否则返回值为Mismatched。

该函数的处理思路如下：

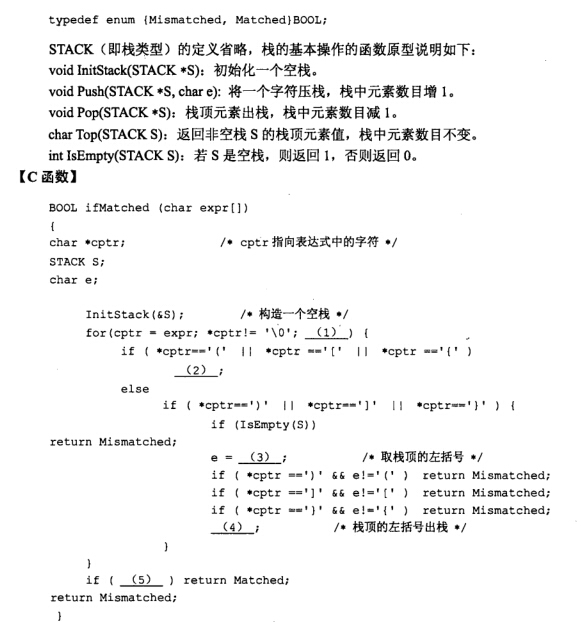
(1) 设置一个初始为空的栈，从左至右扫描表达式。

(2) 若遇上左括号，则令其入栈；若遇上右括号，则需要与栈顶的左括号进行匹配。

(3) 若所遇到的右括号能与栈顶的左括号配对，则令栈顶的左括号出栈' 然后继续匹配过程；否则返回Mismatched，结束判断过程。

(4) 若表达式扫描结束，同时栈变为空，则说明表达式中的括号能完全匹配，返回 Matched o

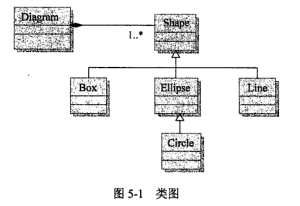
函数ifMatched中用到了两种用户自定义数据类型BOOL和STACK,其中，BOOL 类型的定义如下：



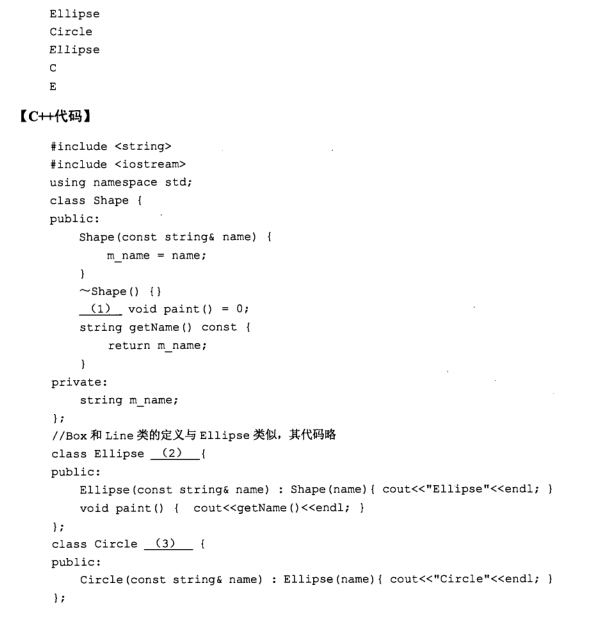
填补C函数中的空缺（1)〜（5)

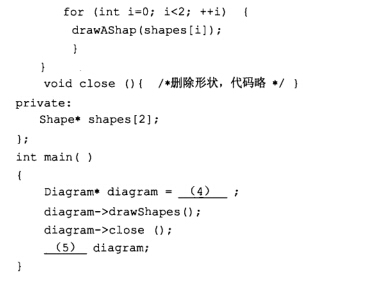
**试题五**

已知对某几何图形绘制工具进行类建模的结果如图5-1所示，其中Shape为抽象类 (应至少包含一个纯虚拟（virtual)函数)，表示通用图形，Box表示矩形，Ellipse表示椭圆，Circle表示圆（即特殊的椭圆），Line表示线条。



下面的C++代码用于实现图5-1所给出的设计思路，将其空缺处填充完整并编译运行，输出结果为：

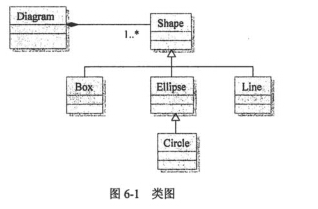




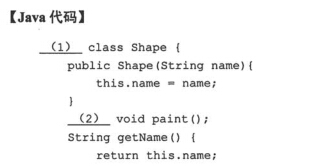
填补C++代码中的空缺（1)〜（5)

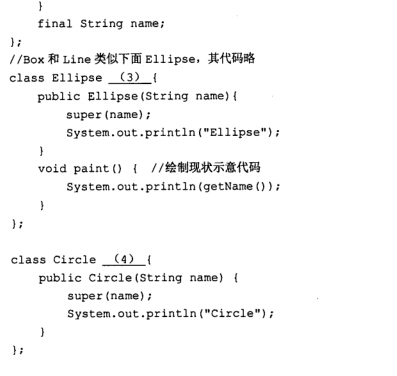
**试题六**

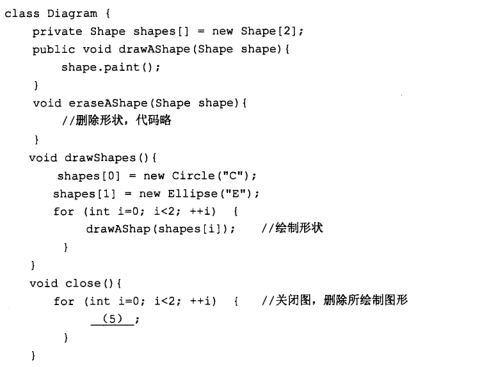
已知对某几何图形绘制工具进行类建模的结果如图6-1所示，其中Shape为抽象 (abstract)类，表示通用图形，Box (矩形)、Ellipse (椭圆）和Line (线条）继承（extends) 了 Shape类，其中，Circle表示圆（即特殊的椭圆）。

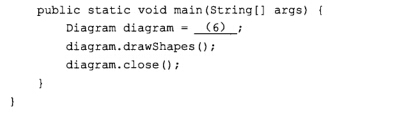


下面的Java代码用于实现图6-1所给出的设计思路，将其空缺处填充完整并编译运行，输出结果为：









填补Java代码中的空缺（1)〜（6)