

上海大学 2021~2022 学年度秋季学期试卷(模拟机考)

成绩

课程名: 高级计算机语言(1)(强) 课程号: 00814388 学分: 4

应试人声明:


我保证遵守《上海大学学生手册》中的《上海大学考场规则》，如有考试违纪、作弊行为，愿意接受《上海大学学生考试违纪、作弊行为界定及处分规定》的纪律处分。

应试人 应试人学号 应试人所在院系

注意事项

1.

为了使同学们了解本课程期末机考的形式和具体的操作方法，特组织本次模拟机考。模拟机考的时长（随堂两节实验课，100 分钟）、题量（4 题）均小于期末机考的时长（120 分钟）和题量（5 题）。机考采取闭卷方式，不允许带任何纸质或电子资料。
2.

必须使用机房的计算机设备。应试人需提前做好准备，若发现所用计算机无法正常工作，请及时报告并由监考教师安排其他机位。开机启动 Win10 操作系统，打开“极域电子教室”学生端, 自行选用 C++ 程序集成开发环境（如：MinGW Developer Studio C/C++, Dev-C++ 或 Code::Blocks 等）。
3.

请在 E 盘以应试人学号-姓名（如 21120000-张三）为名创建新文件夹，分别将应试人的**解题源程序文件**复制到其中，以便在交卷时随该文件夹一起上传到“极域电子教室系统”。重复提交将以最后一次提交为准。应试人解题源程序文件名的主名必须与题号相同（如 A, B, C, D），扩展名为 cpp（如 A.cpp, B.cpp, C.cpp, D.cpp）。不要提交其他文件。
4.

编写、调试及自测解题程序，请在 E 盘上另外新建的文件夹中进行。每题一个程序，采用单文件结构（不要使用多文件结构）。请在每个源程序文件的第一行用注释语句写下应试人的学号和姓名。
5.

【友情提醒】重启计算机会丢失存放在 C 盘、D 盘上的新建文件夹和文件，因此若需安装 C++ 集成开发环境，建议安装到 E 盘。在编写程序过程中，请随时保存解题程序源程序文件以免丢失。
6.

【友情提醒】交卷（即上传解题代码所在的文件夹）后，请至教师机确认上传是否成功，若不成功，需重新提交。

A

问题描述

判断给定的一元二次方程的类型。一元二次方程按其不同实根的个数分为 3 种类型：0 型（无实数根）、1 型（有重根）、2 型（两个相异的实根）。

输入

输入数据有若干行。每一行上有 3 个浮点型数据，分别代表一元二次方程的二次项系数（非零）、一次项系数和常数项，对应一种情形。

输出

一元二次方程的类型。

输入样例

1 2 3

1 -2 1

1 2 -3

输出样例

0

1

2

B

问题描述

统计给定字符串（字符串中可能含有空格字符，字符串的长度不超过 1024 个字符）中元音字母（a, e, i, o, u 及 A, E, I, O, U）出现的次数。

输入

输入数据有若干行。每一行上有一个字符串，字符串的长度不超过 1024 字符，对应一种情形。

输出

字符串中元音字母出现的次数。

输入样例

Shanghai University

Algorithm + Data Structure = Program

输出样例

7

10

C				D																																																			
<p>问题描述</p> <p>给定某同学若干门课程的信息（课程学分、课程百分制成绩），计算该生的平均绩点。百分制成绩与五级十一等成绩及其绩点之间的对应如表 1 所示。</p> <p>表 1 百分制与五级十一等成绩及绩点对照表</p> <table><tr><th></th><th>百分制成绩</th><th>五级十一等成绩</th><th>绩点</th></tr><tr><td>1</td><td>90~100</td><td>A</td><td>4.0</td></tr><tr><td>2</td><td>85~89.9</td><td>A-</td><td>3.7</td></tr><tr><td>3</td><td>82~84.9</td><td>B+</td><td>3.3</td></tr><tr><td>4</td><td>78~81.9</td><td>B</td><td>3.0</td></tr><tr><td>5</td><td>75~77.9</td><td>B-</td><td>2.7</td></tr><tr><td>6</td><td>72~74.9</td><td>C+</td><td>2.3</td></tr><tr><td>7</td><td>68~71.9</td><td>C</td><td>2.0</td></tr><tr><td>8</td><td>66~67.9</td><td>C-</td><td>1.7</td></tr><tr><td>9</td><td>64~65.9</td><td>D</td><td>1.5</td></tr><tr><td>10</td><td>60~63.9</td><td>D-</td><td>1.0</td></tr><tr><td>11</td><td><60</td><td>F</td><td>0.0</td></tr></table> <p>平均绩点的计算公式为</p> $\overline{A_{\text{gpa}}} = \frac{\sum_i c_i g_i}{\sum_i c_i}$ <p>其中，c_i和g_i分别为该生第 i 门课程的学分和绩点。</p> <p>输入</p> <p>输入数据有若干行，每两行对应一种情形。每一种情形中，一行上有一个非负整数 n（n 不超过 100），紧接着的下一行上有 n 门课程的学分和百分制成绩（学分 百分制成绩 学分 百分制成绩……）。其中课程的学分、百分制成绩皆为整数。</p> <p>输出</p> <p>该生 n 门课程的平均绩点，保留 2 位小数。</p> <p>输入样例</p> <pre>1 4 80 2 4 80 3 90</pre>					百分制成绩	五级十一等成绩	绩点	1	90~100	A	4.0	2	85~89.9	A-	3.7	3	82~84.9	B+	3.3	4	78~81.9	B	3.0	5	75~77.9	B-	2.7	6	72~74.9	C+	2.3	7	68~71.9	C	2.0	8	66~67.9	C-	1.7	9	64~65.9	D	1.5	10	60~63.9	D-	1.0	11	<60	F	0.0	<p>输出样例</p> <pre>3.00 3.43</pre> <p>【提示】输出格式控制需要包含头文件<iomanip>。浮点型数据输出时保留 2 位小数的方法为 cout << fixed << setprecision(2);。</p>			
	百分制成绩	五级十一等成绩	绩点																																																				
1	90~100	A	4.0																																																				
2	85~89.9	A-	3.7																																																				
3	82~84.9	B+	3.3																																																				
4	78~81.9	B	3.0																																																				
5	75~77.9	B-	2.7																																																				
6	72~74.9	C+	2.3																																																				
7	68~71.9	C	2.0																																																				
8	66~67.9	C-	1.7																																																				
9	64~65.9	D	1.5																																																				
10	60~63.9	D-	1.0																																																				
11	<60	F	0.0																																																				
				<p>问题描述</p> <p>在给定的有序（升序）整型数组（元素个数为 10，数组元素的初值为 0~9）中插入新元素使之保持有序，同时挤掉一个最大元素。如果新元素的值不小于数组中的最大元素值，则不插入该新元素。</p> <p>输入</p> <p>输入数据有若干个新元素。将新元素依次插入到数组中。</p> <p>输出</p> <p>每次插入一个新元素到数组中的结果。数组元素间用一个空格分隔，行尾无空格。</p> <p>输入样例</p> <pre>1 3 -10 -7 -100 5</pre> <p>输出样例</p> <pre>0 1 1 2 3 4 5 6 7 8 0 1 1 2 3 3 4 5 6 7 -10 0 1 1 2 3 3 4 5 6 -10 -7 0 1 1 2 3 3 4 5 -100 -10 -7 0 1 1 2 3 3 4 -100 -10 -7 0 1 1 2 3 3 4</pre>																																																			