

CS 实验题目

7-1 C# 1.6 求孪生素数

编写控制台应用程序，查找一个用户输入的正整数区间中的孪生素数（孪生素数就是指相差2的素数对）。

输入格式:

用户在一行中输入两个正整数，中间用一个空格间隔。

输出格式:

如果第二个数小于或等于第一个数，则输出“Inputting illegal characters”

如果第二个数大于第一个数，则每行输出一组两个数的闭区间中的孪生素数对，两个数之间用一个空格间隔。

7-2 C# 1.7 查找最高分及学生

编写控制台应用程序，实现如下功能：

- 1) 输入学生姓名（英文字母组成）和考试成绩（推荐保存到结构体数组中）。
- 2) 求最高分并输出对应的姓名（推荐使用foreach语句）。

输入格式:

第一行输入学生人数n

然后，每一行输入一位同学的姓名和其考试成绩，姓名和考试成绩之间用一个空格间隔。

一共输入n位同学的姓名和成绩。

输出格式:

第一行输出最高分。

第二行按用户录入信息的先后次序输出得最高分的同学姓名，不同同学姓名之间用一个空格间隔。

7-3 C# 1.8 体型判断

编写控制台应用程序，根据用户输入的身高 h （以米为单位）和体重 w （以公斤为单位），计算体指数 $t=w/h/h$ ，并输出用户体型。当 $t<18$ 时，体型结果为thin；当 $18\leq t<25$ 时，体型结果为standard；当 $25\leq t<27$ 时，体型结果为little fat；当 $27\leq t$ 时，体型结果为 fat。

输入格式:

第一行输入身高

第二行输入体重

输出格式:

thin, standard, little fat, fat中的其中一个

7-4 C# 1.9 判定闰年

编写控制台应用程序，根据用户输入的一个年份，判断该年份是否为闰年并输入结果（Yes或No）。

输入格式:

第一行输入一个四位的年份

输出格式:

输出Yes表示该年份是闰年，输出No表示该年份不是闰年。

7-5 C# 1.10 计算分段函数

编写控制台应用程序，根据用户输入的x值，参照如下的公式，计算并输出y。

$$y = \begin{cases} 1 - 2x & 0.4 \leq x < 1.4 \\ x & 2.4 \leq x < 4.4 \\ 1 + 2x & 5.4 \leq x < 6.4 \end{cases}$$

输入格式:

输入x值

输出格式:

根据公式计算出的y值

7-6 C# 1.11 打印杨辉三角

编写控制台应用程序，根据用户输入的n值(n>0)，输出前n行的杨辉三角。

输入格式:

输入n值

输出格式:

输出前n行的杨辉三角（每行中的两个相邻数字之间用一个空格间隔）

输入样例:

在这里给出一组输入。例如：

输出样例:

在这里给出相应的输出。例如：

```
1
1 1
1 2 1
1 3 3 1
```

7-1 2.1 求点的欧式距离

编写控制台应用程序，求四维空间中的两点间的欧氏距离。声明一个点类型Point，每个点在x,y,z,t四个维度上有坐标值（可为小数），点的坐标值类外不能访问；定义构造函数用于设定点的4维坐标；定义方法成员Print()，调用该函数时，以格式(x,y,z,t)输出当前点的坐标；定义方法成员Distance(Point p)，调用该函数时，显示当前点与p点之间的欧式距离；定义方法成员Move(char c, double n)，c表示x,y,z,t中的一个维度，调用该函数时，当前点可沿c维度移动位置n。

主程序实现：

- (1) 读入一行输入，该行由四个实数构成（四个数之间用一个空格间隔），用于初始化点对象p1。
- (2) 读入一行输入，该行由四个实数构成（四个数之间用一个空格间隔），用于初始化点对象p2。
- (3) 读入一行输入，该行有一个坐标维度字符和一个实数构成（两者之间用一个空格间隔），其中坐标维度表示点对象将沿着该维度移动，实数表示移动的距离。计算p1点移动后的位置并输出该位置坐标。
- (4) 计算并输出p1与p2点之间的距离。

输入格式:

第一行输入四个实数（四个数之间用一个空格间隔）

第二行输入四个实数（四个数之间用一个空格间隔）

第三行输入一个坐标维度字符和一个实数（两者之间用一个空格间隔）

输出格式:

第一行输出p1沿着第三行输入的维度移动相应距离后的坐标位置，输出四个实数坐标位置（四个数之间用一个逗号间隔）

第二行输出移动后的p1和p2之间的欧式距离

输入样例:

在这里给出一组输入。例如：

```
1 0 0 0
1 1 1 1
x -1
```

输出样例:

在这里给出相应的输出。例如：

```
(0,0,0,0)
Distance=2
```

7-1 2.1 求点的欧式距离

编写控制台应用程序，求四维空间中的两点间的欧氏距离。声明一个点类型Point，每个点在x,y,z,t四个维度上有坐标值（可为小数），点的坐标值类外不能访问；定义构造函数用于设定点的4维坐标；定义方法成员Print()，调用该函数时，以格式(x,y,z,t)输出当前点的坐标；定义方法成员Distance(Point p)，调用该函数时，显示当前点与p点之间的

欧式距离；定义方法成员Move(char c, double n), c表示x,y,z,t中的一个维度，调用该函数时，当前点可沿c维度移动位置n。

主程序实现：

- (1) 读入一行输入，该行由四个实数构成（四个数之间用一个空格间隔），用于初始化点对象p1。
- (2) 读入一行输入，该行由四个实数构成（四个数之间用一个空格间隔），用于初始化点对象p2。
- (3) 读入一行输入，该行有一个坐标维度字符和一个实数构成（两者之间用一个空格间隔），其中坐标维度表示点对象将沿着该维度移动，实数表示移动的距离。计算p1点移动后的位置并输出该位置坐标。
- (4) 计算并输出p1与p2点之间的距离。

输入格式:

第一行输入四个实数（四个数之间用一个空格间隔）

第二行输入四个实数（四个数之间用一个空格间隔）

第三行输入一个坐标维度字符和一个实数（两者之间用一个空格间隔）

输出格式:

第一行输出p1沿着第三行输入的维度移动相应距离后的坐标位置，输出四个实数坐标位置（四个数之间用一个逗号间隔）

第二行输出移动后的p1和p2之间的欧式距离

输入样例:

在这里给出一组输入。例如：

```
1 0 0 0
1 1 1 1
x -1
```

输出样例:

在这里给出相应的输出。例如：

```
(0,0,0,0)
Distance=2
```

7-3 2.4 时间类的定义与使用

编写控制台应用程序，定义一个时间类Time。该类有3个私有整型字段（分别记录24小时制下的时分秒）：hour, minute, second；一个私有的字符串字段timeSystem，可取空、“am”或“pm”；一个公有布尔型字段flag（false表示24小时制，true表示12小时制）一个带三个整型参数（分别对应时分秒）的构造函数为字段进行初始化；一个只读的公有属性Hour用于读取小时数，当flag为true时，将24小时制时间变为12小时制并返回该值，同时修改timeSystem为“am”或“pm”，当flag为false时，直接返回小时数，将timeSystem改为空；成员方法Print()用于输出当前时间，12小时制时格式为：timeSystem Hour:minute:second，24小时制时格式为：hour:minute:second；成员方法AddSecond()用于对秒增加1秒（注意六十进位的问题）。

说明：24小时制时，取值范围[0,23]；12小时制时，取值范围[0,11]；分和秒的取值范围[0,59]。

主程序实现：

- (1) 读入一行输入，该行由三个整数构成（三个数之间用一个空格间隔），分别代表时、分、秒信息。
- (2) 验证用户输入的时分秒信息（按24小时制）是否合法，当用户输入非法日期时，输出“Inputting illegal characters”，程序结束。
- (3) 当用户输入的时分秒信息合法时，用三个参数的构造函数定义一个Time对象t。
- (4) 输入true或false用于设定flag值。

(5) 用户输入n个1（两个1之间用一个空格间隔），每一个1都表示要调用1次AddSecond（）方法，Print输出最新时间。

输入格式:

读入一行输入，该行由三个整数构成（三个数之间用一个空格间隔），分别代表时、分、秒信息。

读入一行输入，zhi为true或false。

读入一行输入，该行由n个1构成（两个1之间用一个空格间隔）。

输出格式:

如果用户输入非法日期时，输出“Inputting illegal characters”，程序结束。

Print输出AddSecond后的最新时间。

输入样例:

在这里给出一组输入。例如：

```
1 1 1
true
1 1 1
```

输出样例:

在这里给出相应的输出。例如：

```
am 1:1:4
```

7-4 2.3 日期类的定义与使用

编写控制台应用程序，定义一个Birthday类。该类有3个私有整型字段

(分别记录某一出生的年月日) : year, month, day; 一个不带参数的构造函数, 取系统时间的年月日为字段初始化; 一个只带年份参数的构造函数, 月和日取1月1日为字段初始化; 一个带年月日三个参数的构造函数为字段初始化; 方法成员GetAge()的功能是计算到系统时间为止该Birthday对象的年龄(按周岁算)并输出。

主程序实现:

- (1) 读入一行输入, 该行由三个整数构成(三个数之间用一个空格间隔), 分别代表年月日信息。
- (2) 验证用户输入的年月日信息是否合法, 年份应不早于1990年, 月和日应为该年份的有效日期(例如 2018 2 29为非法日期), 此外, 用户输入的日期不能大于当前系统日期, 当用户输入非法日期时, 输出“Inputting illegal characters”, 程序结束。
- (3) 当用户输入的年月日信息合法时, 用三个参数的构造函数定义一个Birthday对象b, 并计算对象的年龄并输出。
- (4) 用一个参数的构造函数定义一个Birthday对象b, 并计算对象的年龄并输出。
- (5) 用不带参数的构造函数定义一个Birthday对象b, 并计算对象的年龄并输出。

输入格式:

读入一行输入, 该行由三个整数构成(三个数之间用一个空格间隔), 分别代表年月日信息。

输出格式:

如果用户输入非法年月日取值, 输出“Inputting illegal characters”, 程序结束。

用三个参数的构造函数定义一个Birthday对象b, 并计算对象的年龄并输出。

用一个参数的构造函数定义一个Birthday对象b, 并计算对象的年龄并

输出。

用不带参数的构造函数定义一个Birthday对象b，并计算对象的年龄并输出。

输入样例:

在这里给出一组输入。例如：

```
2020 6 11
```

输出样例:

在这里给出相应的输出。例如：

```
Age is 3
Age is 4
Age is 0
```

7-5 2.5 班级成绩管理系统

编写控制台应用程序，完成班级成绩管理功能。定义学生类Student，记录学号（sno字符串）、姓名（name字符串）、语文

（chinese、）、数学（math）和英语（english）三门课程的期末考试成绩（均为整数）、总成绩（sumScore）的信息，单科及总成绩只在初始化的时候设置，初始化后不能修改。定义班级类SchoolClass，记录该班级全体学生的信息；方法成员SumScore(sno) 能根据参数查询并输出该学号学生的总成绩；方法成员AvgScore() 能统计并输出全班学生各门课程的平均成绩；方法成员MaxScore() 能统计并输出该班级各门课程的最高分；方法成员Fail(cname)能根据参数查询并输出该课程名的课程不及格同学名单，姓名显示顺序与学生信息录入顺序一致；方法成员Paragraph(cname) 能根据参数统计并输出该课程名的课程在不同分数段（ ≥ 90 ， $[80,90)$ ， $[70,80)$ ， $[60,70)$ ， <60 ）的学生人数百

分比；方法成员TopThree()能统计并输出全班前三名的学生名单（如果出现更多同学与前三名同学总成绩相同，需要把总成绩相同的学生姓名一并输出），学生姓名显示顺序先按总成绩高低排，总成绩一样时，按照学生录入顺序排。

主程序实现：

- (1) 读入一行输入，该行只有一个整数n，表示班级学生人数。
- (2) 当用户输入的 $n \leq 0$ 时，输出“Inputting illegal characters”，程序结束。
- (3) 读入n行输入，每行由5个数据项构成（每个数据项之间用一个空格间隔），分别代表学号、姓名、语文成绩、数学成绩、英语成绩信息。根据读入的信息构建相应对象。
- (4) 读入一行输入：
 - a) 当输入为1 sno时(两个输入项直接由一个空格间隔，sno指代某一具体学号)，输出该学号学生的总成绩，格式如：SumScore=267”，程序结束。如果没有该学号的学生，则输出“No student.”，程序结束。
 - b) 当输入为2时，输出全班学生各门课程的平均成绩，格式如：AvgChinese=80;AvgMath=76;AvgEnglish=68，程序结束。
 - c) 当输入为3时，输出各门课程的最高分，格式如：MaxChinese=80;MaxMath=76;MaxEnglish=68，程序结束。
 - d) 当输入为4 cname时(两个输入项直接由一个空格间隔，cname指代某一具体课程名，取值为chinese，english或math)，输出该课程不及格同学名单，姓名显示顺序与学生信息录入顺序一致，格式如：Lucy;Mick;Tom，程序结束。如参数值不是chinese，english或math，则输出“Inputting illegal characters.”，程序结束。
 - e) 当输入为5 cname时（两个输入项直接由一个空格间隔，cname指代某一具体课程名，取值为chinese，english或math），输出各分数段（分数段由高到低）的学生人数百分比，格式如：“20% 20% 20% 20% 20%”，程序结束。如参数值不是chinese，english或math，则输出“Inputting illegal characters.”，程序结束。

f) 当输入为6时，输出全班总成绩前三名的学生名单（如果出现更多同学与前三名同学总成绩相同，需要把总成绩相同的学生姓名一并输出），学生姓名显示顺序先按总成绩高低排，当总成绩一样时，按照学生录入顺序排，格式如：“Lucy;Mick;Tom;Lily”，程序结束。

g) 当输入不是上面任何一组时，则输出”Inputting illegal characters.”，程序结束。

输入格式:

读入一行输入，该行只有一个整数n，表示班级学生人数。

读入n行输入，每行由5个数据项构成（每个数据项之间用一个空格间隔），分别代表学号、姓名、语文成绩、数学成绩、英语成绩信息。

读入一行输入，表示用户要进行的查询选项及参数。

输出格式:

当用户输入的 $n \leq 0$ 时，输出“Inputting illegal characters”，程序结束。
根据用户的查询选项，输出查询或统计结果。

输入样例:

在这里给出一组输入。例如：

```
4
1 a 90 80 70
2 b 90 80 90
3 c 90 70 80
4 d 70 80 90
6
```

输出样例:

在这里给出相应的输出。例如：

b;a;c;d

5月17日01

给出下面的一个基类框架：`class Person { protected int no;//编号
public virtual void display()//输出相关信息 {} }`以Person为基类，构建出Student、Teacher两个类。生成上述类并编写主函数，要求主函数中有一个基类Person的一维数组，用于存放学生或教师对象，数组长度为10。主函数根据输入的信息，相应建立Student, Teacher类对象，对于Student给出期末5门课的成绩（为整数，缺考的科目填-1），对于Teacher则给出近3年，每年发表的论文数量。输入格式：每个测试用例占一行，第一项为人员类型，1为Student，2为Teacher。接下来为编号（1-9999），接下来Student是5门课程成绩，Teacher是3年的论文数。最后一行为0，表示输入的结束。输出格式：要求输出编号，以及Student缺考的科目数和已考科目的平均分（四舍五入取整，已考科目数为0时，不输出平均分），Teacher的3年论文总数。

5月17日02

一个茶吧提供三类饮料：茶、咖啡和牛奶。其中本地茶要另加50%的服务费，其它茶要加20%的服务费；现磨咖啡要加100%的服务费，其它咖啡加20%的服务费；牛奶不加服务费。服务费精确到小数点后一位。定义饮料类Drink，它有三个受保护的字段分别记录饮料编号（整型），购买数量（整型），单价（最多1位小数）；一个带三个参数的构造函数用来初始化字段；一个虚拟的成员方法用来计算并输出该编号的饮料总价格是多少（如果总价格为小数，则保留到小数点后一位，推荐使用Math.Round()方法）。以Drink为基类，构建出Tea、Coffee和Milk三个派生类。派生类Tea有一个私有的字段记录地区码（整型）；有两个私有常量serviceCharge1和serviceCharge2，分别记录本地茶加收的服务费50%和其它茶加收的服务费20%。派生类

Coffee有一个私有的字段记录加工类型（整型）；有一个私有常量serviceCharge1和serviceCharge2，分别记录现磨咖啡加收的服务费100%和其它咖啡加收的服务费20%。（提示：应用重写虚拟实现多态）主函数功能：定义一个Drink数组（长度10）；根据用户输入的信息，相应建立Tea, Coffee或Milk类对象，计算并显示收费信息。用户输入的信息的每一行信息格式如下：第一项为饮料的类型，茶为1，咖啡为2，牛奶为3。第二项是申请的编号[100-999]。第三项是数量。第四项是单价。第五项：对于茶叶来说，接下来输入一个地区代码，其中1代表本地；对于咖啡来说，接下来要输入一个加工代码，其中1代表现磨。对于牛奶来说没有第五项。当用户输入的一行为0时，表示输入结束。输入格式：每个测试用例占一行，用户可输入多行测试用例。当用户输入的一行为0时，表示输入结束。输出格式：每一行输出对应用户的一行有效输入，输出饮料编号和收费（如果总价格为小数，则保留到小数点后一位，推荐使用Math.Round()方法）。如果某行测试记录饮料类型不为1, 2, 3, 则该条输入对应的输出为"Drink type error." 如果某行测试记录饮料类型对，但是饮料编号不在[100,999]之间，则输出"Drink ID error." 如果某行测试记录饮料类型、编号对，购买数量小于0，则输出"Drink number error." 如果某行测试记录饮料类型、编号、购买数量对，单价小于0，则输出"Drink price error." 如果用户一行为0，则无输出结果。

7-1 C# 3.1 Person派生类

分数 10

全屏浏览

切换布局

作者 陈卓

单位 青岛科技大学

给出下面的一个基类框架：

```
class Person
{ protected int no;//编号
public virtual void display()//输出相关信息
{}}
```

以Person为基类，构建出Student、Teacher两个类。生成上述类并编写主函数，要求主函数中有一个基类Person的一维数组，用于存放学生或教师对象，数组长度为10。

主函数根据输入的信息，相应建立Student, Teacher类对象，对于Student给出期末5门课的成绩（为整数，缺考的科目填-1），对于Teacher则给出近3年，每年发表的论文数量。

输入格式:

每个测试用例占一行，第一项为人员类型，1为Student，2为Teacher。接下来为编号（1-9999），接下来Student是5门课程成绩，Teacher是3年的论文数。最后一行为0，表示输入的结束。

输出格式:

要求输出编号，以及Student缺考的科目数和已考科目的平均分（四舍五入取整，已考科目数为0时，不输出平均分），Teacher的3年论文总数。

输入样例:

在这里给出一组输入。例如：

```
1 19 -1 -1 -1 -1 -1
1 125 78 66 -1 95 88
2 68 3 0 7
2 52 0 0 0
```

```
1 6999 32 95 100 88 74
0
```

输出样例:

在这里给出相应的输出。例如：

```
19 5
125 1 82
68 10
52 0
6999 0 78
```

7-2 C# 3.2 Drink及其派送类

一个茶吧提供三类饮料：茶、咖啡和牛奶。其中本地茶要另加50%的服务费，其它茶要加20%的服务费；现磨咖啡要加100%的服务费，其它咖啡加20%的服务费；牛奶不加服务费。服务费精确到小数点后一位。定义饮料类Drink，它有三个受保护的字段分别记录饮料编号（整型），购买数量（整型），单价（最多1位小数）；一个带三个参数的构造函数用来初始化字段；一个虚拟的成员方法用来计算并输出该编号的饮料总价格是多少（如果总价格为小数，则保留到小数点后一位，推荐使用Math.Round()方法）。

以Drink为基类，构建出Tea、Coffee和Milk三个派生类。派生类Tea有一个私有的字段记录地区码（整型）；有两个私有常量serviceCharge1和serviceCharge2，分别记录本地茶加收的服务费50%和其它茶加收的服务费20%。派生类Coffee有一个私有的字段记录加工类型（整型）；有一个私有常量serviceCharge1和serviceCharge2，分别记录现磨咖啡加收的服务费100%和其它咖啡加收的服务费20%。（提示：应用重写虚拟实现多态）

主函数功能：定义一个Drink数组（长度10）；根据用户输入的信息，

相应建立Tea, Coffee或Milk类对象，计算并显示收费信息。用户输入的信息的每一行信息格式如下：

第一项为饮料的类型，茶为1，咖啡为2，牛奶为3。

第二项是申请的编号[100-999]。

第三项是数量。

第四项是单价。

第五项：对于茶叶来说，接下来输入一个地区代码，其中1代表本地；对于咖啡来说，接下来要输入一个加工代码，其中1代表现磨。对于牛奶来说没有第五项。

当用户输入的一行为0时，表示输入结束。

输入格式:

每个测试用例占一行，用户可输入多行测试用例。当用户输入的一行为0时，表示输入结束。

输出格式:

每一行输出对应用户的一行有效输入，输出饮料编号和收费（如果总价格为小数，则保留到小数点后一位，推荐使用Math.Round()方法）。

如果某行测试记录饮料类型不为1，2，3，则该条输入对应的输出为”Drink type error.”

如果某行测试记录饮料类型对，但是饮料编号不在[100,999]之间，则输出”Drink ID error.”

如果某行测试记录饮料类型、编号对，购买数量小于0，则输出”Drink number error.”

如果某行测试记录饮料类型、编号、购买数量对，单价小于0，则输出”Drink price error.”

如果用户一行为0，则无输出结果。

输入样例:

在这里给出一组输入。例如:

```
1 106 3 33 1
1 103 2 20 2
3 109 1 15
2 107 2 15.8 1
2 232 3 21 29
0
```

输出样例:

在这里给出相应的输出。例如:

```
106 148.5
103 48
109 15
107 63.2
232 75.6
```

7-3 C# 3.3 学生信息管理

设计一个学生信息管理控制台应用程序，实现对小学生、中学生、大学生个人姓名、年龄及考试课程成绩的输入，以及平均成绩的统计和显示。功能要求如下：

- 1) 每个学生都有姓名和年龄。
- 2) 小学生有语文、数学成绩。
- 3) 中学生有语文、数学和英语成绩。
- 4) 大学生有必修课学分总数和选修课学分总数，不包含单科成绩。
- 5) 学生类提供向外输出信息的方法。
- 6) 学生类提供统计个人总成绩或总学分的方法。

- 7) 通过静态成员自动记录学生总人数。
- 8) 能通过构造函数完成各字段成员初始化。

提示：

(1) 定义一个抽象学生类：定义受保护字段分别记录学生姓名和年龄，定义公有静态字段记录班级人数。定义一个带两个参数的构造函数，初始化学生姓名和年龄，同时更新班级人数。定义公有只读属性获取学生姓名，定义公有只读虚属性获取学生类型——student。定义公有抽象方法计算学生总分。定义公有方法成员返回学生信息字符串。

(2) 定义学生类的派生类小学生类Pupil：定义受保护字段分别记录语文和数学成绩（可为小数），定义带4个参数的构造函数初始化相应字段。重写虚属性获取学生类型——pupil。重写抽象方法计算两门课的平均成绩（保留两位小数）。

(3) 定义学生类的派生类中学生类Middle：定义受保护字段分别记录语文、数学和英语成绩（可为小数），定义带5个参数的构造函数初始化相应字段。重写虚属性获取学生类型——middle school student。重写抽象方法计算三门课的平均成绩（保留两位小数）。

(4) 定义学生类的派生类大学生类College：定义受保护字段分别记录必修课学分和选修课学分（可为小数），定义带4个参数的构造函数初始化相应字段。重写虚属性获取学生类型——college student。重写抽象方法计算总学分（保留两位小数）。

主函数功能：

定义一个小学生一维数组（长度10）；根据用户输入的信息，相应建立小学生、中学生或大学生类对象，用户输入结束后显示学生信息。用户输入的信息的每一行信息格式如下：

第一项为学生类型，小学生为1，中学生为2，大学生为3。

第二项是学生姓名。

第三项是学生年龄。

第四项对小学生和中学生来说是语文成绩；对大学生来说是必修课学分。

第五项对小学生和中学生来说是数学成绩；对大学生来说是选修课学分。

第六项对中学生来说是英语成绩；对小学生和大学生来说没有该项。当用户输入的一行为0时，表示输入结束。

输入格式:

每个测试用例占一行，用户可输入多行测试用例。当用户输入的一行为0时，表示输入结束。

输出格式:

每一行输出对应用户的一行有效输入，输出学生总人数和该位同学的个人信息。

如果一行中用户输入多个值或一个非零值，且学生类型不属于[1,3]，或者学生信息没有按要求给齐，则没有任何输出，统计学生人数时也不统计该学生。

如果用户输入一行为0，则无输出结果。

输入样例:

在这里给出一组输入。例如：

```
1 a 10 90
2 b 15 90 90 78
3 c 20 19 89
0
```

输出样例:

在这里给出相应的输出。例如：

```
Total number of student:2, Name:b, middle school student,  
Age is 15, AvgScore:86.00;  
Total number of student:2, Name:c, college student, Age  
is 20, TotalCredits :108.00;
```

7-4 C# 3.4 图形类

设计一个求图形面积的控制台应用程序。定义一个抽象类Figure，有一个受保护的常量pi值为3.1415926；有一个抽象成员Area()可以计算该图形的面积并返回面积值（小数）。定义一个接口IFigure，有一个方法成员Perimeter()可以计算该图形的周长并返回周长值（小数）。

定义Figure和IFigure的派生类Circle、Rectangle、Triangle。

Circle类有一个私有字段记录半径（小数），有一个带一个参数的构造函数。

Rectangle类有两个私有字段记录长（小数）、宽（小数），有一个带两个参数的构造函数。

Triangle类有三个私有字段记录三条边长（小数），有一个带三个参数的构造函数。

主函数功能：

- (1) 读入用户的一行输入。
- (2) 根据其中值的个数创建对象或报错：
 - a) 当输入值个数为1时，如果该值是合理的圆半径（大于0）则创建圆对象；否则报错“Cannot build a circle.”
 - b) 当输入值个数为2时，如果该值是合理的矩形长、宽值（值都大于0）则创建矩形对象；否则报错“Cannot build a rectangle.”
 - c) 当输入值个数为3时，如果该值是合理的三角形边的值（值都大于0，任意两边和大于第三边）则创建三角形对象；否则报错“Cannot build a triangle.”
 - d) 当输入值个数为其他值时，报错“Inputting illegal characters.”，程序

结束。

(3) 如果能创建对象，则调用对象的方法求面积和周长，并输出。
输出格式如下：

- a) Rectangle area is 1.2
- b) Rectangle circumference is 2.4
- c) Circle area is 3.1415926
- d) Circle circumference is 6.2831852
- e) Triangle area is 0.433012701892219
- f) Triangle circumference is 3

输入格式:

输入一行数据测试数据。

输出格式:

如果输入有效，则第一行输入图形的面积；第二行输出图形的周长。
如果输入无效，则输出一行错误提示。

输入样例:

在这里给出一组输入。例如：

```
1
```

输出样例:

在这里给出相应的输出。例如：

```
Circle area is 3.1415926
Circle circumference is 6.2831852
```

7-5 C# 3.5 接口类

设计一个模拟播放器控制台应用程序。定义接口IPlayer，包含5个返回值为字符串的方法：播放Play()、停止Stop()、暂停Pause()、上一首Pre()、下一首Next()。定义接口IPlayer的派生类MP3。

主函数功能：

- (1) 声明一个MP3对象。
- (2) 获取用户的一行输入。
 - a) 当输入0时，程序结束。
 - b) 当输入1时，模拟播放功能，输出“Playing music.”。
 - c) 当输入为2时，模拟停止功能，输出“Stop play.”。
 - d) 当输入为3时，模拟暂停功能，输出“Suspend play.”。
 - e) 当输入为4时，模拟上一首功能，输出“Previous song.”。
 - f) 当输入为5时，模拟下一首功能，输出“Next song.”。
 - g) 当输入[0,5]之外的值时，输出“Inputting illegal characters.”。

输入格式:

输入一个整数。

输出格式:

当输入[1-5]之间的数字时模拟音乐播放、停止、暂停、上一首、下一首功能。当输入小于0或大于5的整数时，输出“Inputting illegal characters.”。

输入样例:

在这里给出一组输入。例如：

输出样例:

在这里给出相应的输出。例如:

```
Playing music.
```

7-6 C# 3.7 该日是该年的第几天

定义一个日期类Date，内有数据成员年、月、日，构造函数用于初始化数据成员，方法Day()能计算并返回该日是该年的第几天。主函数：根据用户的输入，创建一个Date对象。用户可以多次输入年月日信息，直到用户输入0 0 0，则表示输入结束。根据用户的输入日期顺序，调用Day()方法，依次计算并输出每个日期是该年的第几天。

输入格式:

每一行由3个整数构成，分别表示年、月、日信息，两个整数之间用一个空格间隔。当用户输入0 0 0则表示输入结束。

输出格式:

按用户的输入顺序逐行输出每个日期是该年的第几天。

输入样例:

在这里给出一组输入。例如:

```
2006 1 2
2000 3 5
0 0 0
```

输出样例:

在这里给出相应的输出。例如:

2

65