编程题库

* 基础概念

1. 编写程序，显示Welcome to XXX（XXX代表具体的编程语言）、Welcome to Computer Science和Programming is fun。注，显示三条消息。
2. 编写程序，显示Welcome to XXX（XXX代表具体的编程语言）五次。注，显示五条消息。
3. 编写程序，显示以下表格：

a a^2 a^3

1 1 1

2 4 8

3 9 27

4 16 64

注，打印表格。

1. 编写程序，显示以下公式的结果：。注，计算表达式。
2. 编写程序，显示 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9的结果。注，数列求和。
3. 我们可以使用以下公式计算PI：。编写程序，显示 和 的结果。在程序中用1.0代替1。注，近似求PI。
4. 编写程序，计算并显示半径为5.5的圆的面积和周长。注，圆的面积和周长。
5. 编程程序，计算并显示宽度为4.5、长度为7.9的矩形的面积和周长。注，矩形的面积和周长。
6. 假设一个跑步者在45分钟30秒内跑了14公里。编写程序，计算并显示以“英里/每小时”为单位的平均速度。提示：1英里等于1.6公里。注，以英里计的平均速度。
7. 假设一个跑步者在1小时40分钟35秒内跑了24英里。编写程序，显示以“公里/每小时”为单位的平均速度。提示：1英里等于1.6公里。注，以公里计的平均速度。
8. （人口估算，\*）统计局基于以下假设进行人口估算：
   1. 每7秒有一个人出生
   2. 每13秒有一个人死亡
   3. 每45秒有一个移民迁入

编写程序，显示未来5年内每年的人口数。假设，当前的人口是312032486，每年有365天。注，人口估算。

提示：在C++/Java中，两个整数相除，结果还是整数，小数部分被（自动）截断。例如，5/4 等于1（而不是1.25）；10/4等于2（而不是2.5）。如果想得到有小数部分的精确结果，那么进行除法运算的两个值中的任意一个必须是一个具有小数点的数值。例如，5.0/4等于1.25；10/4.0等于2.5。

1. （求解二元线性方程组，\*）我们可以使用Cramer规则求解下面的二元线性方程组：

编写程序，求解以下方程组并显示x和y的值。注，代数：求解二元线性方程组。

* 基本程序设计（进阶：变量的输入）

1. 编写程序，从控制台读入double型的摄氏温度，然后将其转换为华氏温度，并显示结果。转换公式如下所示：华氏温度 = (9/5) \* 摄氏温度 + 32。

下面是一个运行示例：

Enter a degree in Celsius: 43

43 Celsius is 109.4 Fahrenheit

注，将摄氏温度转换为华氏温度。

1. 编写程序，读入圆柱体的半径和高，并使用下列公式计算圆柱的体积：

，

下面是一个运行示例：

Enter the radius and length of a cylinder: 5.5 12

The area is 95.0331

The volume is 1140.4

注，计算圆柱体的体积。

1. 编写程序，读入“英尺”数，将其转换为“米”数并显示结果。一英尺等于0.305米。下面是一个运行示例：

Enter a value for feet: 16.5

16.5 feet is 5.0325 meters

注，将英尺转换为米。

1. 编写程序，将“磅”数转换为“千克”数。程序提示用户输入磅数，然后转换成千克并显示结果。一磅等于0.454千克。下面是一个运行示例：

Enter a number in pounds: 55.5

55.5 pounds is 25.197 kilograms

1. （计算小费，\*）编写一个程序，读入一笔费用和酬金率，计算酬金和总钱数。例如，如果用户输入10作为费用，15%作为酬金率，那么计算结果显示酬金为$1.5，总费用为$11.5。下面是一个运行示例：

Enter the subtotal and a gratuity rate: 10 15

The gratuity is $1.5 and total is $11.5

1. (求一个整数各位数的和，\*\*)编写程序，读取一个在0到1000之间的整数，并将该整数的各位数字相加。例如，整数是932，各位数字之和为14。提示：利用操作符 % 分解数字，然后使用操作符 / 去掉分解出来的数字。例如，932 % 10 = 2 ，932 / 10 = 93。下面是一个运行示例：

Enter a number between 0 and 1000: 999

The sum of the digits is 27

1. （求出年数，\*）编写程序，提示用户输入分钟数（例如，十亿），然后显示这些分钟数代表多少年和多少天。为了简化问题，假设一年有365天。下面是一个运行示例：

Enter the number of minutes: 1000000000

1000000000 minutes is approximately 1902 years and 214 days

1. （加速度）平均加速度定义为速度的变化量除以这个变化所用的时间，如下式所示：。编写程序，提示用户输入以“米/秒”为单位的起始速度、以“米/秒”为单位的终止速度，以及以“秒”为单位的时间段t，最后显示平均加速度。下面是一个运行示例：

Enter v0, v1, and t: 5.5 50.9 4.5

The average acceleration is 10.0889

1. （计算能量）编写程序，计算将水从初始温度加热到最终温度所需的能量。程序应该提示用户输入水的重量（以千克为单位），以及水的初始温度和最终温度。计算能量的公式是：。其中，M是以千克为单位的水的重量，温度以摄氏度为单位，能量Q以焦耳为单位。下面是一个运行示例：

Enter the amount of water in kilograms：55.5

Enter the initial temperature: 3.5

Enter the final temperature: 10.5

The energy needed is 1625484.0

1. （人口统计）重新编写练习11，提示用户输入年数，然后显示这个年数之后的人口值。将编程练习题11中的提示用于这个程序。人口数应该类型转换为一个整数。下面是一个运行示例：

Enter the number of years: 5

The population in 5 years is 325932970

1. （求出跑道长度）假设一架飞机的加速度是a而起飞速度是v，那么可以使用下面的公式计算出飞机起飞时所需要的跑道最短长度： 。编写程序，提示用户输入以“米/秒”（m/s）为单位的速度v和以“米/秒的平方”（m/s2）为单位的加速度a，然后显示跑道最短长度。下面是一个运行示例：

Enter speed and acceleration: 60 3.5

The minimum runway length for this airplane is 514.286

1. （复利值，\*\*）假设你每月向银行账户存100美元，年利率为5%，那么每月利率是0.05 / 12 = 0.00417。第一个月之后，账户上的值就变成：

100 \* (1 + 0.00417) = 100.417

第二个月之后，账户上的值就变成：

(100 + 100.417) \* (1 + 0.00417) = 201.252

第三个月之后，账户上的值就变成：

(100 + 201.252) \* (1 + 0.00417) = 302.507

依次类推。编写程序，显示六个月后账户上的钱数。下面是一个运行示例：

Enter the monthly saving amount: 100

After the sixth month, the account value is $608.81

1. （计算BMI，\*）身体质量指数（BMI）是对体重的健康测量。它的值可以通过将体重（以公斤为单位）除以身高（以米为单位）的平方得到。编写程序，提示用户输入体重（以磅为单位）和身高（以英寸为单位），然后显示BMI。提示：一磅是0.45359237公斤；一英寸是0.0254米。下面是一个运行示例：

Enter weight in pounds: 95.5

Enter height in inches: 50

BMI is 26.8573

1. （两点间距离）编写程序，提示用户输入两个点(x1,y1)和(x2,y2)，然后显示两点间的距离。注意，可以使用sqrt(a)来计算。下面是一个运行示例：

Enter x1 and y1: 1.5 -3.4

Enter x2 and y2: 4 5

The distance between the two points is 8.7641

1. （六边形面积）编写程序，提示用户输入六边形的边长，然后显示它的面积。计算六边形面积的公式是： 。其中，s就是边长。下面是一个运行示例：

Enter the side: 5.5

The area of the hexagon is 78.5918

1. （风寒温度，\*）除了温度以外，包括风速、相对湿度以及阳光等其他的因素在确定室外是否寒冷方面都起了很重要的作用。2001年，美国国家气象服务局（NWS）利用温度和风速计算新的风寒温度，来衡量寒冷程度。计算公式如下所示：。其中， 是室外的温度，以华氏摄氏度为单位；v 是风速，以每小时英里数为单位。是风寒温度。该公式不适用于风速低于2 mph，或温度在-58以下或41以上的情况。

编写程序，提示用户输入在-58和41之间的温度度数，同时大于或等于2的风速，然后显示风寒温度。使用pow(a, b)来计算。下面是一个运行示例：

Enter the temperature in Fahrenheit between -58 and 41:5.3

Enter the wind speed (>=2) in miles per hour: 6

The wind chill index is -5.56707

1. 编写程序，显示下面的表格。将浮点数值类型转换为整数。

a b pow(a,b)

1 2 1

2 3 8

3 4 81

4 5 1024

5 6 15625

1. （三角形的面积，\*）编写程序，提示用户输入三角形的三个顶点(x1,y1)、(x2,y2)和(x3,y3)，然后显示它的面积。计算三角形面积的公式是：

。

下面是一个运行示例：

Enter three points for a triangle: 1.5 -3.4 4.6 5 9.5 -3.4

The area of the triangle is 33.6

1. （计算利息，\*）如果知道收支余额和年利率（百分比），那么就可以使用下面的公式计算下个月要支付的利息额：。编写程序，读取收支余额和年利率，显示下个月的利息。下面是一个运行示例：

Enter balance and interest rate (e.g., 3 for 3%):1000 3.5

The interest is 2.91667

1. （计算未来投资值，\*）编写程序，读取投资总额、年利率和年数，然后使用下面的公式显示投资金额的未来值：

。

例如，如果输入的投资金额为1000，年利率为3.25%，年数为1，那么未来投资额为1032.98。下面是一个运行示例：

Enter investment amount: 1000.56

Enter annual interest rate in percentage: 4.25

Enter number of years: 1

Accumulated value is $1043.92

1. （驾驶费用，\*）编写一个程序，提示用户输入驾驶的距离、每加仑多少英里的汽车燃油性能，以及每加仑的价格，然后显示旅程的费用。下面是一个运行示例：

Enter the driving distance: 900.5

Enter miles per gallon: 25.5

Enter price per gallon: 3.55

The cost of driving is $125.36

* 选择

1. （解一元二次方程，\*）考虑一元二次方程。称作一元二次方程的判别式。如果它是正值，那么方程就有两个实数根。如果它为0，那么方程就只有一个根。如果它是负值，那么方程无实根。编写程序，提示用户输入a、b和c的值，并且显示基于判别式的结果。如果这个判别式为正，那么显示两个根。如果判别式为0，那么显示一个根。否则，显示“The equation has no real roots”。注意：可以使用pow(x,0.5)或sqrt(x)来计算。下面是一些运行示例：

Enter a,b,c: 1.0 3 1

The equation has two roots -0.381966 and -2.61803

Enter a,b,c: 1 2.0 1

The equation has one roots -1

Enter a,b,c: 1 2 3

The equation has no real roots

1. （三个数的加法）编写程序，随机产生三个一位整数，然后提示用户输入这三个整数的和，并给出“正确或错误”的提示。
2. （求解二元一次线性方程组，\*）编写程序，使用Cramer规则解线性方程组： 。编写程序，提示用户输入a、b、c、d、e和f，然后显示结果。如果ad-bc为0，那么显示“The equation has no solution”。下面是一些运行示例：

Enter a,b,c,d,e,f:9.0 4.0 3.0 -5.0 -6.0 -21.0

x is -2.0 and y is 3.0

Enter a,b,c,d,e,f:1.0 2.0 2.0 4.0 4.0 5.0

The equation has no solution

1. （随机月份，\*\*）编写一个随机产生1到12之间整数的程序，并且根据数字1,2,…,12显示相应的英文月份：January,February,…,December。
2. （找到将来的日期，\*）编写一个程序，提示用户输入代表今天日期的数字（周日为0，周一为1，……，周六为6）。同时，提示用户输入一个今天之后的天数，作为代表将来某天的数字，然后显示这天是星期几。下面是一些运行示例：

Enter today’s day: 1

Enter the number of days elapsed since today: 3

Today is Monday and the future day is Thursday

Enter today’s day: 0

Enter the number of days elapsed since today: 31

Today is Sunday and the future day is Wednesday

1. （BMI，\*）编写程序，提示用户输入重量（磅）、身高（英尺和英寸），计算BMI及其相应的信息(表1)。例如，一个人身高是5英尺10英寸，输入的英尺值就是5、英寸值为10。下面是一个运行示例：

Enter weight in pounds: 140

Enter feet: 5

Enter inches: 10

BMI is 20.0877

Normal

表1. BMI说明

|  |  |
| --- | --- |
| BMI | 说明 |
|  | 偏瘦 |
|  | 正常 |
|  | 超重 |
|  | 过胖 |

1. （对三个整数排序，\*）编写程序，提示用户输入三个整数。按照从小到大的形式（升序）显示这三个整数。
2. （检查ISBN-10，\*\*）ISBN-10（国际标准书号）是一个10位整数，最后的一位是校验和。它是使用下面的公式，用另外9个数计算出来的：。如果校验和为10，那么按照ISBN-10的规定，最后一位应该表示为X。编写程序，提示用户输入前9个数，然后显示10位ISBN（包括前面起始位置的0）。程序应该读取一个整数输入。下面是一些运行示例：

Enter the first 9 digits of an ISBN as integer: 013601267

The ISBN-10 number is 0136012671

Enter the first 9 digits of an ISBN as integer: 013031997

The ISBN-10 number is 013031997X

1. 编写程序，随机产生一个计算两个小于100的整数的加法问题。
2. （给出一个月的总天数，\*）编写程序，提示用户输入月份和年份，然后显示这个月的天数。例如，如果用户输入的月份是2而年份是2012，那么程序应该显示“February 2012 has 29 days”（2012年2月有29天）。如果用户输入的月份是3而年份是2015，那么程序就应该显示“March 2015 has 31 days”。
3. （回文数字）编写一个程序，提示用户输入一个三位的整数，然后确定它是否是回文数字。如果一个数字从左到右、以及从右到左都是一样的，那么这个数字称为回文数。下面是一些运行示例：

Enter a three-digit integer: 121

121 is a palindrome

Enter a three-digit integer: 123

123 is not a palindrome

1. （计算税款，\*）编写程序，计算所有登记的婚姻状态的税款。

表2. 2009年美国国家联邦个人收入所得税税率表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 税率 | 单身纳税人 | 已婚共同纳税人或证实的鳏寡 | 已婚单独纳税人 | 家庭户主纳税人 |
| 10% | $0 ~ $8350 | $0 ~ $16700 | $0 ~ $8350 | $0 ~ $11950 |
| 15% | $8351 ~ $33950 | $16701 ~ $67900 | $8351 ~ $33950 | $11951 ~ $45500 |
| 25% | $33951 ~ $52250 | $67901 ~ $137050 | $33951 ~ $68525 | $45501 ~ $117450 |
| 28% | $52251 ~ $171550 | $137051 ~ $208850 | $68526 ~ $104425 | $117451 ~ $190200 |
| 33% | $171551 ~ $372950 | $208851 ~ $372950 | $104426 ~ $186475 | $190201 ~ $372950 |
| 35% | $372951+ | $372951+ | $186476+ | $372951+ |

1. （猜硬币的正反面）编写程序，让用户猜一猜是硬币的正面还是反面。这个程序随机产生一个整数0或者1，它们分别表示硬币的正面和反面。程序提示用户输入一个猜测值，然后显示这个猜测值是正确的还是错误的。
2. （彩票，\*\*）编写程序，产生三位整数的彩票。程序提示用户输入一个三位整数，然后依照下面的规则判定用户是否赢得奖金：
   1. 如果用户输入的所有数匹配彩票的确切顺序，那么赢得10000元奖金。
   2. 如果用户输入的所有数匹配彩票的所有数字，那么赢得3000元奖金。
   3. 如果用户输入的所有数中的某个数匹配彩票号码中的一个数字，那么赢得1000元奖金。
3. （随机点）编写程序，显示矩形中一个随机点的坐标。矩形中心位于(0,0)、宽100、高200。
4. （游戏石头剪刀布，\*）编写程序，实现游戏“剪刀-石头-布”。程序提示用户随机产生一个数字，这个数为0、1或者2，分别表示石头、剪刀和布。程序提示用户输入值0、1或者2，然后显示一条消息，表明用户和计算机谁赢了游戏、谁输了游戏，或是打成平手。下面是一些运行示例：

scissor (0), rock (1), paper (2): 1

The computer is scissor. You are rock. You won.

scissor (0), rock (1), paper (2): 2

The computer is paper. You are paper too. It is a draw.

1. （运输成本，\*）一家运输公司使用下面的函数，根据运输重量（w,磅）来计算运输成本（c,美元）:

。

编写程序，提示用户输入包裹重量，显示运输成本。如果重量大于20，显示一条消息“the package cannot be shipped”。

1. （计算三角形的周长，\*\*）编写程序，读取三角形的三条边。如果输入值合法就计算这个三角形的周长；否则，显示这些输入值不合法。如果任意两条边的和大于第三边，那么输入值都是合法的。
2. （风寒温度，\*）编程练习题28给出计算风寒温度的公式。这个公式只适用于温度在华氏-58度到41度之间，并且风速大于或等于2的情况。编写一个程序，提示用户输入一个温度值和一个风速值。如果输入值是合法的，那么显示风寒温度；否则，显示一条消息表明温度或风速是非法数值。
3. （某天是星期几，\*\*）泽勒一致性是由克里斯汀泽勒开发的用于计算某天是星期几的算法。这个公式是：。其中，
   * h是一个星期中的某一天（0为星期六；1为星期天；2为星期一；3为星期二；4为星期三；5为星期四；6为星期五）。
   * q是某月的第几天。
   * m是月份（3为三月，4为四月，，12为十二月）。一月和二月份分别记为上一年的13和14月。
   * j是世纪数，即 。
   * k是该世纪的第几年，即 year % 100。

注意，公式中的除法执行一个整数相除。编写程序，提示用户输入年、月和该月的哪一天，然后显示它是一周中的星期几。下面是一些运行示例：

Enter year (e.g., 2012): 2015

Enter month (1-12): 1

Enter the day of the month (1-31): 25

Day of the week is Sunday

Enter year (e.g., 2012): 2012

Enter month (1-12): 5

Enter the day of the month (1-31): 12

Day of the week is Saturday

提示：在这个公式中，一月和二月份分别采用13和14表示。所以，需要将用户输入的月份1转换为13；将用户输入的月份2转换为14。同时，将年份改为前一年。

1. （点是否在圆内，\*\*）编写程序，提示用户输入一个点（x,y），然后检查这个点是否在以原点（0,0）为圆心、半径为10的圆内。例如，（4,5）是圆内的一点，而（9,9）是圆外的一点。下面是一些运行示例：

Enter a point with two coordinates: 4 5

Point (4.0, 5.0) is in the circle

Enter a point with two coordinates: 9 9

Point (9.0, 9.0) is not in the circle

1. （点是否在矩形内，\*\*）编写程序，提示用户输入点(x, y)，然后检测该点是否在以原点(0, 0)为中心、宽为10、高为5的矩形中。例如，(2, 2)在矩形内，而(6, 4)在矩形外。提示，如果一个点到(0,0)的水平距离小于等于10/2且到(0,0)的垂直距离小于等于5.0/2，那么这个点就在矩形内。下面有两个运行示例：

Enter a point with two coordinates: 2 2

Point (2.0, 2.0) is in the rectangle.

Enter a point with two coordinates: 6 4

Point (6.0, 4.0) is not in the rectangle.

1. （挑一张牌，\*\*）编写程序，模拟从一副52张的牌中选择一张牌。程序应该显示牌的大小（Ace,2,3,4,5,6,7,8,9,10,Jack,Queen,King）以及牌的花色（Clubs（草花）、Diamonds（方块）、Hearts（红心）和Spades（黑桃））。下面是一个运行示例：

The card you picked is Jack of Hearts.

1. （交点，\*）第一条直线上的两个点是(x1,y1)和(x2,y2)，第二条直线的两个点是(x3,y3)和(x4,y4)。这两条直线的交点可以通过下面的线性方程组求解：

这个线性方程组可以应用Cramer规则求解。如果方程组无解，那么这两条直线平行。编写程序，提示用户输入这四个点，然后显示它们的交点。下面是一些运行示例:

Enter x1,y1,x2,y2,x3,y3,x4,y4: 2 2 5 -1.0 4.0 2.0 -1.0 -2.0

The intersecting point is at (2.88889, 1.1111)

Enter x1,y1,x2,y2,x3,y3,x4,y4: 2 2 7 6.0 4.0 2.0 -1.0 -2.0

The two lines are parallel.

1. （逻辑运算符）编写程序，提示用户输入一个整数值，然后判定它是否能被5和6整除，是否能被5或6整除，以及能否被5或6整除但不能同时被它们整除。下面是一个运行示例：

Enter an integer: 10

Is 10 divisible by 5 and 6? false

Is 10 divisible by 5 or 6? true

Is 10 divisible by 5 or 6, but not both? True

1. （点是否在三角形内，\*\*）假设一个直角三角形放在一个平面上。直角点在(0,0)处，其他两个点分别在(200,0)和(0,100)处。编写程序，提示用户输入一个点的x坐标和y坐标，然后判定这个点是否在该三角形内。下面是运行示例：

Enter a point’s x- and y-coordinates: 100.5 25.5

The point is in the triangle.

Enter a point’s x- and y-coordinates: 100.5 50.5

The point is not in the triangle.

1. （两个矩形，\*\*）编写一个程序，提示用户输入两个矩形中心点的x坐标和y坐标以及它们的宽度和高度，然后判定第二个矩形是在第一个矩形内，还是和第一个矩形交叠。下面是一些运行示例：



Enter r1’s center x-,y-coordinates,width,height: 2.5 4 2.5 43

Enter r2’s center x-,y-coordinates,width,height: 1.5 5 0.5 3

r2 is inside r1.

Enter r1’s center x-,y-coordinates,width,height: 1 2 3 5.5

Enter r2’s center x-,y-coordinates,width,height: 3 4 4.5 5

r2 overlaps r1.

Enter r1’s center x-,y-coordinates,width,height: 1 2 3 3

Enter r2’s center x-,y-coordinates,width,height: 40 45 3 2

r2 does not overlap r1.

1. （两个圆，\*\*）编写程序，提示用户输入两个圆的圆心坐标和各自的半径值，然后决定第二个圆是在第一个圆内，还是和第一个圆重叠。提示，如果两个圆心的距离 ，那么就认为circle2在circle1内；如果两个圆心的距离 ，那么就认为circle2和circle1重叠。下面是一些运行示例：

Enter circle1’s center x-,y-coordinates,radius: 0.5 5.1 13

Enter circle2’s center x-,y-coordinates,radius: 1 1.7 4.5

circle2 is inside circle1.

Enter circle1’s center x-,y-coordinates,radius: 3.4 5.7 5.5

Enter circle2’s center x-,y-coordinates,radius: 6.7 3.5 3

circle2 overlaps circle1.

Enter circle1’s center x-,y-coordinates,radius: 3.4 5.5 1

Enter circle2’s center x-,y-coordinates,radius: 5.5 7.2 1

circle2 does not overlap circle1.

1. （货币兑换，\*）编写一个程序，提示用户输入从美元到人民币的兑换汇率。提示用户输入0，表示从美元兑换为人民币；输入1，表示从人民币兑换为美元。继而提示用户输入美元数量或者人民币数量，分别兑换为另外一种货币。下面是一些运行示例：

Enter the exchange rate from dollars to RMB: 6.81

Enter 0 to convert dollars to RMB and 1 vice versa: 0

Enter the dollar amount: 100

$100.0 is 681.0 yuan

Enter the exchange rate from dollars to RMB: 6.81

Enter 0 to convert dollars to RMB and 1 vice versa: 1

Enter the RMB amount: 10000

10000 yuan is $1468.43

Enter the exchange rate from dollars to RMB: 6.81

Enter 0 to convert dollars to RMB and 1 vice versa: 5

Incorrect input.

1. （点的位置，\*）给定一个从点p0(x0,y0)到p1(x1,y1)的有向线段，我们可以使用下面的条件来确定点p2(x2,y2)是在线段的左侧、右侧或者在该直线上：

编写一个程序，提示用户输入三个点p0、p1和p2，显示p2是否在线段的左侧、右侧或在该直线上。下面是一些运行示例：

Enter three points for p0, p1, p2: 4.4 2 6.5 9.5 -5 4

(-5.0, 4.0) is on the left side of the line from (4.4, 2.0) to (6.5, 9.5)

Enter three points for p0, p1, p2: 1 1 5 5 2 2

(2.0, 2.0) is on the line from (1.0, 1.0) to (5.0, 5.0)

Enter three points for p0, p1, p2: 3.4 2 6.5 9.5 5 2.5

(5.0, 2.5) is on the right side of the line from (3.4, 2.0) to (6.5, 9.5)

1. （比较成本，\*）假设你要通过两种不同的包裹运输大米。编写一个程序来比较成本。该程序提示用户输入每个包裹的重量和价格，然后显示具有更好价格的包裹。下面是一些运行示例：

Enter weight and price for package 1: 50 24.59

Enter weight and price for package 2: 25 11.99

Package 2 has a better price.

Enter weight and price for package 1: 50 25

Enter weight and price for package 2: 25 12.5

Two packages have the same price.

1. （线段上的点，\*）编程练习题63显示了如何测试一个点是否在一条无限长的直线上。修改练习题63，测试一个点是否在一条线段上。编写程序，提示用户输入三个点p0、p1和p2，显示p2是在线段p0p1上。下面是一些运行示例：

Enter three points for p0, p1, p2: 1 1 2.5 2.5 1.5 1.5

(1.5, 1.5) is on the line segment from (1.0, 1.0) to (2.5, 2.5).

Enter three points for p0, p1, p2: 1 1 2 2 3.5 3.5

(3.5, 3.5) is not on the line segment from (1.0, 1.0) to (2.0, 2.0).

* 循环

1. （统计正数和负数的个数，然后计算这些数的平均值，\*）编写程序，读入未指定个数的整数，判读读入的正数有多少个，读入的负数有多少个，然后计算这些输入值的总和及其平均值（不对0计数）。当输入为0时，表明程序结束。将平均值以浮点数显示。下面是一个运行示例：

Enter an integer, the input ends if it is 0: 1 2 -1 3 0

The number of positives is 3

The number of negatives is 1

The total is 5.0

The average is 1.25

Enter an integer, the input ends if it is 0: 0

No numbers are entered except 0

1. （重复加分）编写程序，产生10个随机加法问题，加数是两个1到15之间的整数。显示（回答）正确答案的个数和测验时间。
2. （将千克转换成磅）编写程序，显示下面的表格。注，1千克为2.2磅。

千克 磅

1 2.2

3 6.6

…

197 433.4

199 437.8

1. （将英里转换成千米）编写程序，显示下面的表格。注，1英里等于1.609千米。

英里 千米

1. 1.609
2. 3.218

……

9 14.481

10 16.090

1. （千克与磅之间的互换）编写一个程序，并排显示下列两个表格。

千克 磅 磅 千克

1 2.2 20 9.09

2 6.6 25 11.36

…

197 433.4 510 231.82

199 437.8 515 243.09

1. （英里与千米之间的互换）编写一个程序，并排显示下列两个表格。

英里 千米 千米 英里

1 1.609 20 12.430

2 3.218 25 15.538

…

9 14.481 60 37.290

10 16.090 65 40.398

1. （计算未来的学费，\*\*）假设今年某大学的学费为10000元，学费的年增长率为5%。一年后，学费将是10500元。编写程序，计算10年后的学费，以及从现在开始的10年后算起，4年内总学费是多少（第11年、第12年、第13年和第14年，这四年的总学费）？
2. （找出最高分）编写程序，提示用户输入学生的个数、每个学生的名字及其分数，最后显示获得最高得分的学生的名字（和分数）。
3. （找出两个分数最高的学生，\*）编写程序，提示用户输入学生的个数、每个学生的名字及其分数，最后显示获得最高分的学生和第二高分的学生（姓名和成绩）。
4. （找出能被5或者6整除的数）编写程序，显示从100到1000之间所有能被5和6整除的数。每行显示10个。数字之间用一个空格字符隔开。
5. （找出能被5或者6整除，但不能被两者同时整除的数）编写程序，显示从100到200之间所有能被5或6整除，但不能被两者同时整除的数，每行显示10个数。数字之间用一个空格字符隔开。
6. （求满足的n的最小值）编写程序，使用while循环找出满足大于12000的最小正整数n。
7. （求满足的n的最大值）编写程序，使用while循环找出满足小于12000的最大正整数n。
8. （计算最大公约数，\*）下面是求两个正整数n1和n2的最大公约数的另一种解法：首先找出n1和n2的最小值d，然后依次检验d,d-1,d-2,…,2,1是否是n1和n2的公约数。第一个满足条件的公约数就是n1和n2的最大公约数。编写程序，提示用户输入两个正整数，然后显示最大公约数。
9. （显示ASCII码字符表，\*）编写一个程序，打印ASCII字符表中从’!’到’~’的字符。每行打印10个字符。符号之间用一个空格字符隔开。
10. （找出一个整数的因子，\*）编写程序，读入一个正整数，然后以升序显示它的所有最小因子。例如，若输入的整数是120，那么输出就应该是：2,2,2,3,5。
11. （使用循环语句打印4个图案，\*）编写程序，使用嵌套的循环语句，用四个独立的程序打印下面的图案：

图案1 图案2

1 1 2 3 4 5 6

1. 2 1 2 3 4 5

1 2 3 1 2 3 4

1 2 3 4 1 2 3

1 2 3 4 5 1 2

1 2 3 4 5 6 1

图案3 图案4

1 1 2 3 4 5 6

2 1 1 2 3 4 5

3 2 1 1 2 3 4

4 3 2 1 1 2 3

5 4 3 2 1 1 2

6 5 4 3 2 1 1

1. （显示金字塔，\*\*）编写程序，提示用户输入一个在1到15之间的整数，然后显示一个金字塔形状的图案。下面是一个运行示例：

Enter the number of lines : 7

1

2 1 2

3 2 1 2 3

4 3 2 1 2 3 4

5 4 3 2 1 2 3 4 5

6 5 4 3 2 1 2 3 4 5 6

7 6 5 4 3 2 1 2 3 4 5 6 7

1. （打印金子塔型的数字，\*\*）编写程序，使用嵌套的for循环，打印下面的输出：

1

1 2 1

1 2 4 2 1

1 2 4 8 4 2 1

1 2 4 8 16 8 4 2 1

1 2 4 8 16 32 16 8 4 2 1

1 2 4 8 16 32 64 32 16 8 4 2 1

1. （打印2到1000之间的素数，\*）编写程序，打印2到1000之间，包括2和1000的所有素数，每行显示8个素数。数字之间用一个空格字符隔开。
2. （比较不同利率下的贷款，\*\*）编写程序，让用户输入贷款总额和以年为单位的贷款期限，然后显示利率在从5%到8%、每次递增1/8的过程中，每月的支付额和总支付额。下面是一个运行示例：

Loan Amount: 10000

Number of Years: 5

Interest Rate Monthly Payment Total Payment

5.000% 188.71 11322.74

5.125% 189.29 11357.13

5.250% 189.86 11391.59

…

7.875% 202.17 12129.97

8.000% 202.76 12165.84

计算月支付额的公式：

1. （显示分期还贷时间表，\*\*）对于给定的贷款额的月支付额包括偿还本金及利息。月利息是通过月利率乘以余额（剩余本金）计算出来的。因此，每月偿还的本金等于月支付额减去月利息。编写一个程序，让用户输入贷款总额、贷款年数以及利率，然后显示分期还贷时间表。下面是一个运行示例：

Loan Amount: 10000

Number of Years：1

Annual Interest Rate: 7

Monthly Payment: 865.26

Total Payment: 10383.21

Payment# Interest Principal Balance

1 58.33 806.93 9193.07

2 53.62 811.64 8381.43

…

11 10.0 855.26 860.27

12 5.01 860.25 0.01

注，最后一次偿还后，余额可能不为0。如果是这样的话，那么最后一个月支付额应当是正常的月支付额加上最后的余额。

提示：编写一个循环来打印该表。因为每个月的还贷额都是相同的，所以应当在循环之前计算它。开始时，余额就是贷款总额。在循环的每次迭代中，计算利息及本金，然后更新余额。

1. （示例抵消错误，\*）当处理一个很大的数字以及一个很小的数字的时候，会产生一个抵消错误（cancellation error）。例如，100000000.0 + 0.000000001等于100000000.0。为了避免抵消错误，从而获得更加精确的结果，应该谨慎地选择计算的次序。例如，在计算下面的数列时，从右往左计算要比从左往右计算得到的结果更精确：。编写程序，对上面的数列分别按从左到右和从右到左顺序进行计算，并对结果进行比较。这里，取n=50000。
2. （数列求和，\*）编写程序，计算下面数列的和：。
3. （计算，\*\*）使用下面的数列可以近似计算：

编写程序，显示当 时 的值。

1. （计算e，\*\*）使用下面的数列可以近似计算e: 。编写程序，显示当 时 的值 。提示，因为，所以 。将e和通项item初始化为1，反复将新的item加到e上。新的item由前一个item除以i得到。其中， 。
2. （显示闰年，\*\*）编写程序，显示从101到2100期间所有的闰年，每行显示10个。数字之间用一个空格字符隔开，同时显示这期间闰年的数目。判断闰年的标准是：1、能整除4且不能整除100 2、能整除400。
3. （显示每月第一天是星期几，\*\*）编写程序，提示用户输入年份和代表该年第一天是星期几的数字，然后在控制台上显示该年每月第一天的日期（星期几）。例如，如果用户输入的年份是2013和代表2013年1月1日为星期二的数字2，那么程序应该显示如下输出：(算法见第53题)

Jan 1, 2013 is Tuesday

…

Dec 1, 2013 is Sunday

1. （显示日历，\*\*）编写程序，提示用户输入年份和代表该年第一天是星期几的数字，然后在控制台上显示该年的日历表。例如，如果用户输入年份2013和代表2013年1月1日为星期二的2，那么程序应该显示该年每个月的日历，如下所示：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| January 2013 | | | | | | |
| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|  |  | 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 27 | 28 | 29 | 30 | 31 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| December 2013 | | | | | | |
| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | 31 |  |  |  |  |

1. （复利值，\*）假设你每月在储蓄账户上存100元，年利率是5%。那么，每月利率是0.05/12 = 0.00417。在第一个月之后，账户上的值变成：

100 \* (1 + 0.00417) = 100.417

第二个月之后，账户上的值变成：

(100 + 100.417) \* (1 + 0.00417) = 201.252

第三个月之后，账户上的值变成：

(100 + 201.252) \* (1 + 0.00417) = 302.507

依次类推。编写程序，提示用户输入一个储蓄额（例如，100）、年利率（例如，5）以及月份数（例如，6），然后显示给定月份后账户上的余额。

1. （计算某项目的价值，\*）假设你向某个项目投资10000元，年获利率为5.75%。一个月后，这个项目的价值为

10000 + 10000 \* 5.75 / 1200 = 10047.91

两个月之后，这个项目的价值为

10047.91 + 10047.91 \* 5.75 / 1200 = 10096.06

三个月之后，这个项目的价值为

10096.06 + 10096.06 \* 5.75 / 1200 = 10144.44

依次类推。编写程序，提示用户输入一个总数（例如，10000）、年获利率（例如，5.75）以及月份数（例如，18），然后显示一个表格，如下面的运行示例所示：

Enter the initial deposit amount: 10000

Enter annual percentage yield：5.75

Enter maturity period (number of months): 18

Month Project Value

1. 10047.92
2. 10096.06

……

17 10846.57

18 10898.54

1. （随机数，\*）编程程序，产生两个不同的两位正整数。提示，产生第一个数，使用循环不断产生第二个数，直到它和第一个数不同为止。
2. （完全数，\*\*）如果一个正整数等于除它本身之外其他所有因子之和，那么就称它为完全数。例如，6是第一个完全数，6 = 1 + 2 + 3。第二个完全数是28,28 = 14 + 7 + 4 + 2 + 1。10000以下的完全数有四个。编写程序，找出这四个完全数。
3. （游戏：石头、剪刀、布，三星难度）修改游戏“石头-剪刀-布”，让用户可以连续地玩这个游戏，直到用户或者计算机（连续）赢对手两次以上为止。
4. （加法，\*）编写程序，计算下面的和：
5. （检测ISBN，\*\*）使用循环，简化编程练习题41。
6. （十进制到二进制，\*\*）编写程序，提示用户输入一个十进制正整数，然后显示对应的二进制值。
7. （十进制到八进制，\*\*）编写程序，提示用户输入一个十进制整数，然后显示对应的八进制值。
8. （求出销售总额，\*）假设售货员的工资包括基本工资和提成。基本工资是5000元，并且使用下面的方案确定售货员的提成率：

|  |  |
| --- | --- |
| 销售额 | 提成率 |
| 0.01 ~ 5000 | 8% |
| 5000.01 ~ 10000 | 10% |
| 10000.01 及以上 | 12% |

注意，提成采用渐进税率。第一个5000元的税率是8%，第二个5000元是10%，余下的是12%。如果销售额是25000，那么提成为

5000 \* 8% + 5000 \* 10% + 15000 \* 12% = 2700。

编写程序，找出为挣到30000元所必须完成的最小销售额。

1. (求销售额，\*)按要求改写练习104。
   * + 使用for循环。
     + 允许用户输入目标输入。
2. （正面或反面）编写程序，模拟抛硬币一百万次，显示出现正面和反面的次数。
3. （最大数的出现次数，\*）编写程序读取整数，找出它们的最大数，然后计算该数的出现次数。假设输入是以0结束的。如果输入为 3 5 2 5 5 5 0，那么程序找出最大数是5，而5出现的次数是4。提示，维护max和count两个变量。Max存储当前最大值，而count存储它的出现次数。在初始状态下，将第一个数赋值给max，count赋值为1。然后，将接下来的每个数字逐个地和max进行比较。如果这个数大于max，那么就把它赋值给max，同时将count重置为1。如果这个数等于max，那么就给count加1。

Enter numbers: 3 5 2 5 5 5 0

The largest number is 5

The occurrence count of the largest number is 4

1. (组合，\*)编写程序，显示从整数1到7中选择两个数字的所有组合，以及所有组合的总数（量）。
2. 2
3. 3

…

…

The total number of all combinations is 21

1. （计算平均值和和标准方差，\*\*）编写一个程序，提示用户输入10个数字，然后运用下面的公式，计算并显示这些数字的平均数和标准方差。

，

下面是一个运行示例：

Enter ten numbers: 1 2 3 4.5 5.6 6 7 8 9 10

The mean is 5.61

The standard deviation is 2.99794

1. (倒排一个字符串，\*)编写程序，提示用户输入一个字符串，然后以反序显示该字符串。下面是一个运行示例：

Enter a string: ABCD

The reversed string is DCBA

1. (检测ISBN-13，\*)ISBN-13是一个标识书籍的新标准。它使用13位数字。最后一位数字是校验和。它是使用下面的公式计算得到的：

如果校验和为10，那么将其替换为0。程序应该将输入作为一个字符串读入。下面是一些运行示例：

Enter the first 12 digits of an ISBN-13 as a string: 978013213080

The ISBN-13 number is 9780132130806

Enter the first 12 digits of an ISBN-13 as a string: 978013213079

The ISBN-13 number is 9780132130790

Enter the first 12 digits of an ISBN-13 as a string: 97801320

97801320 is an invalid input

1. (处理字符串，\*)编写一个程序，提示用户输入一个字符串，显示奇数位置的字符。下面是一个运行示例：

Enter a string: Beijing Chicago

BiigCiao

1. (对元音和辅音进行计数，\*)编写一个程序，提示用户输入一个字符串，然后显示字符串中元音和辅音的数目。元音字母为AEIOU。下面是一个运行示例：

Enter a string: Programming is fun

The number of vowels is 5

The number of consonants is 11

1. (对大写字母计数，\*)编写一个程序，提示用户输入一个字符串，然后显示该字符串中大写字母的数目。下面是一个运行示例：

Enter a string: Welcome to C++

The number of uppercase letters is 2

1. (最长的共同前缀，\*)编写一个程序，提示用户输入两个字符串，显示两个字符串最长的共同前缀。下面是一些运行示例：

Enter the first string: Welcome to C++

Enter the second string: Welcome to programming

The common prefix is Welcome to

Enter the first string: Beijing

Enter the second string: Shanghai

Beijing and Shanghai have no common prefix

* 数学函数、字符和字符串

1. （五边形的面积）编写程序，提示用户输入从正五边形中心到顶点的距离，计算五边形的面积。五边形面积的计算公式为：，其中 s 是边长。边长s的计算公式为：，其中 r 是从五边形中心到顶点的距离。结果保留两位小数。下面是一个运行示例。

Enter the length from the center to a vertex: 5.5

The area of the pentagon is 71.92

1. （最大球面距离，\*）最大球面距离是指球面上两个点之间的距离。假设(x1,y1)和(x2,y2)是两个点的地理经纬度。那么，这两个点之间的最大球面距离可以使用以下公式计算：

。

编写一个程序，提示用户以“度”为单位输入地球上两个点的经纬度，显示其最大球面距离。地球的平均半径为6371.01km。注意，需要将“度”转换为“弧度”。公式中的经纬度是相对北边和西边，负数表示相对南边和东边的度数。下面是一个运行示例。

Enter point1 (latitude and longitude) in degrees: 39.55 -116.25

Enter point2 (latitude and longitude) in degrees: 41.5 87.37

The distance between the two points is 10691.792 km

1. （估算面积，\*）从网址[www.gps-data-tem.com/map](http://www.gps-data-tem.com/map) 处找到Georgia州的Atlanta、Florida州的Orlando、Georgia州的Savannah和North Carolina的Charlotte。计算被这四个城市所围起来的区域的面积。提示，使用习题117中的公式来计算两个城市之间的距离；将多边形分为两个三角形，使用习题30中的公式计算三角形的面积。



三角形面积： 。

1. （六边形面积）六边形面积可以通过下面公式计算（s是边长）： 。编写程序，提示用户输入六边形的边长，然后显示它的面积。下面是一个运行示例。

Enter the side: 5.5

The area of the hexagon is 78.59

1. （正多边形的面积，\*）正多边形是一个n条边的多边形，它的每条边的长度都相等，并且所有角的度数也相等（即多边形既等边又等角）。计算正多边形面积的公式是： 。这里，s是边长。编写一个程序，提示用户输入边的个数以及正多边形的边长，然后显示它的面积。下面是一个运行示例。

Enter the number of sides: 5

Enter the side: 6.5

The area of the polygon is 72.69

1. （圆上的随机点，\*）编写程序，产生位于圆心在(0,0)、半径为30的圆上面的三个随机点，显示由这三个随机点组成的三角形的三个角的度数。提示，产生 之间的一个以弧度为单位的随机角度 ，由这个角度所确定的点为。



1. （顶点坐标，\*）假设一个正五边形的中心位于（0,0），其中一个点位于0点（12点）位置（如图所示）。编写程序，提示用户输入正五边形外切圆的半径，显示正五边形上五个顶点的坐标。下面是一个运行示例。

0点位置



Enter the radius of the bounding circle: 100

The coordinates of five points on the pentagon are

(95.1057, 30.9017)

(0.00013, 100)

(-95.1056, 30.9019)

(-58.7788, -80.9015)

(58.7782, -80.902)

1. （给出ASCII码对应的字符，\*）编写一个程序，输入一个ASCII码值（0 ~ 127之间的一个整数），然后显示该字符。下面是一个运行示例。

Enter an ASCII code: 69

The character for ASCII code 69 is E

1. （十进制转十六进制，\*）编写一个程序，提示用户输入 0 ~ 15 之间的一个整数，显示其对应的十六进制数。下面是一些运行示例。

Enter a decimal value (0 to 15): 11

The hex value is B

Enter a decimal value (0 to 15): 5

The hex value is 5

Enter a decimal value (0 to 15): 31

31 is an invalid input

1. （十六进制转二进制）编写程序，提示用户输入一个十六进制数，显示其对应的二进制数。下面是一些运行示例。

Enter a hex digit: 8

The binary value is 1011

Enter a hex digit: G

G is an invalid input

1. （判断元音还是辅音，\*）编写程序，提示用户输入一个字母，判断该字母是元音还是辅音。下面是是一些运行示例。

Enter a letter: B

B is a consonant

Enter a letter: a

A is a vowel

Enter a letter: #

# is an invalid input

1. （转换字母等级为数字，\*）编写程序，提示用户输入一个字母等级A、B、C、D或F，显示对应的数字值4、3、2、1或0。下面是一些运行示例。

Enter a letter grade: B

The numeric value for grade B is 3

Enter a letter grade: T

T is an invalid grade

1. （随机字符）编写程序，显示一个随机的大写字母。
2. （一个月中的日期，\*）编写一个程序，提示用户输入一个年份和一个月份名称的前三个字母（第一个字母使用大写形式），显示该月的天数。下面是一个运行示例。

Enter a year: 2001

Enter a month: Jan

Jan 2001 has 31 days

1. （学生的专业和状况，\*）编写一个程序，提示用户输入两个字符，显示这个两个字符代表的专业和年级。第一个字符表示专业，第二个是一个数字字符1、2、3和4。它们分别表示该学生是大一、大二、大三或者大四的学生。假设下面的字符用于表示专业：M，数学；C，计算机科学；I，信息技术。下面是一些运行示例。

Enter two characters: M1

Mathematics Freshman

Enter two characters: C3

Computer Science Junior

Enter two characters: T3

Invalid input

1. （检测ISBN-10）改写编程练习题41，将ISBN号作为一个字符串输入。
2. （字符串处理）编写程序，提示用户输入一个字符串，显示它的长度和第一个字符。
3. （检查SSN，\*）编写一个程序，提示用户输入一个社保号码，判断该号码是否合法。SSN的格式是DDD-DD-DDDD。其中，D是一个数字。下面是一个运行示例。

Enter a SSN： 232-23-5435

232-23-5435 is a valid social security number

Enter a SSN： 23-23-5435

23-23-5435 is a invalid social security number

1. （检测子串）编写程序，提示用户输入两个字符串，检测第二个字符串是否是第一个字符串的子串。下面是一些运行示例。

Enter string s1: ABCD

Enter string s2: BC

BC is a substring of ABCD

Enter string s1: ABCD

Enter string s2: BDC

BDC is not a substring of ABCD

1. （酬金，\*）编写一个程序，读取下面的信息，然后输出一个酬金声明：雇员姓名（如：Smith）、每周的工作小时数（如，10小时）、每小时的酬金（如，9.75元）、所得税税率（如，20%）和地方税税率（如，9%）。下面是一个运行示例。

Enter employee’s name: Smith

Enter number of hours worked in a week: 10

Enter hourly pay rate: 9.75

Enter federal tax withholding rate: 0.20

Enter state tax withholding rate: 0.09

Employee Name: Smith

Hours Worked: 10.0

Pay Rate: $9.75

Gross Pay: $97.5

Deductions:

Federal Withholding (20.0%): $19.5

State Withholding (9.0%): $8.77

Total Deduction: $28.77

Net Pay: $69.22

1. （对三个城市排序，\*）编写程序，提示用户输入三个城市的名称，然后以升序进行显示。下面是一个运行示例。

Enter the first city: Chicago

Enter the second city：Los Angeles

Enter the third city: Atlanta

The three cities in alphabetical order are Atlanta Chicago Los Angeles

1. （生成车牌号码，\*）编写一个程序，生成一个车牌号码。车牌号由三个大写字母和后面的3个数字组成。
2. （货币单位，\*）编写程序，将给定的钱数分成较小的货币单位。这个程序要求用户输入一个代表“总价”的字符串，如“11.56”。然后，输出一个清单，列出和“总价”相等的最大数量的元、5角、1角、5分和1分的数目。按照这个顺序，从而使得硬币最少。提示，使用indexof和substring方法分别抽取小数点前后的（钱数）数量；使用stoi方法将字符串转换为数字。

* 一维数组

1. （指定等级，\*）编写一个程序，读入学生成绩，获得最高分best，然后根据下面的规则赋等级值：
   1. 如果分数 >= best – 10，那么等级为 A。
   2. 如果分数 >= best – 20，那么等级为 B。
   3. 如果分数 >= best – 30，那么等级为 C。
   4. 如果分数 >= best – 40，那么等级为 D。
   5. 其他，等级为F。

程序提示用户输入学生总数，然后提示用户输入所有的分数，最后显示等级得出结论。下面是一个运行示例：

Enter the number of students: 4

Enter 4 scores: 40 55 70 58

Student 0 score is 40 and grade is C

Student 1 score is 55 and grade is B

Student 2 score is 70 and grade is A

Student 3 score is 58 and grade is B

1. (倒置输入的数)编写程序，读取10个整数，然后按照和读入顺序相反的顺序将它们显示出来。
2. (计算数字的出现次数，\*\*)编写程序，读取从1到100之间的整数，然后计算每个数出现的次数。假定输入是以0为结束标志。下面是一个运行示例：

Enter the integers between 1 and 100:2 5 6 5 4 3 23 43 2 0

2 occurs 2 times

3 occurs 1 time

4 occurs 1 time

5 occurs 2 times

6 occurs 1 time

23 occurs 1 time

43 occurs 1 time

注意，如果一个数出现的次数大于一次，那么在输出时使用复数”times”。

1. （分析成绩）编写一个程序，读入个数不确定的考试分数，并且判断有多少个分数是大于等于平均分，多少个分数是低于平均分的。输入一个负数表示输入结束。假设最高分为100。
2. (打印不同的数，\*\*)编写一个程序，读入10个数，然后显示互不相同的数（即便某个数出现多次，但也仅显示一次）。提示，读入一个数字。如果它是一个新数，那么把它存储在数组中。如果该数已经在数组，那么忽略它。在输入之后，数组包含的都是不同的数。下面是一个运行示例：

Enter ten numbers: 1 2 3 2 1 6 3 4 5 2

The number of distinct number is 6

The distinct numbers are: 1 2 3 6 4 5

1. (判断素数，\*)判断n是否是素数的另一个更有效的方法是：检验小于等于 的素数是否都能整除n。如果不能，那么n就是素数。编写程序，使用这个方法找出前50个素数。注，需要使用一个数组来存储这些素数，然后再检查它们是否是n的可能因子。
2. (统计数字的出现次数，\*)编写程序，生成100个0到9之间的随机整数，然后显示每一个数字出现的次数。提示，使用一个整型数组存放0,1,…,9的个数。
3. 编写程序，求数组的平均值。
4. 编写程序，找出整数数组中的最小值。
5. 编写程序，找出整数数组中最小元素的索引值。如果这样的元素个数大于1，那么返回最小的索引值。
6. (计算标准差，\*)编写程序，计算标准差。计算公式如下：

，

下面是一个运行示例：

Enter ten numbers: 1.9 2.5 3.7 2 1 6 3 4 5 2

The mean is 3.11

The standard deviation is 1.55738

1. (倒置数组，\*)编写程序，实现数组的倒置复制。例如，输入10个数字并保存至数组A中，然后将这10个数字按照相反顺序保存至新数组B中。
2. 编写程序，随机产生100000个整数值和一个查询值，然后进行线性搜索，显示搜索结果并估计执行时间。
3. （冒泡排序，\*\*）编写程序，读取10个double型的值，使用冒泡排序对它们进行排序，然后显示排好序的数字。冒泡排序算法遍历数组几次。在每次遍历中，对相邻的两个元素进行比较。如果这一对元素是降序，那么交换它们的值；否则，保持值不变。因为较小的值如气泡一样逐渐“浮向”顶部，同时较大的值“沉向”底部。所以，这种排序技术称为“冒泡排序法”（bubble sort）。
4. （选择排序法，\*）编写程序，读取10个整数，使用选择排序对它们进行排序，然后显示排序好的数字。选择排序法重复地在当前数组中找到最小值，然后把这个最小值与该数组中的第一个元素进行交换（或者，重复地在当前数组中找到最大值，然后将这个最大值和该数组中的最后一个元素进行交换）。
5. 编写程序，随机产生100000个整数值和一个查询值，然后进行折半搜搜，显示搜索结果并估计执行时间。注，先对100000个整数值进行排序。
6. （对学生排序，\*\*）编写一个程序，提示用户输入学生个数、学生姓名和他们的成绩，然后按照学生成绩的降序顺序打印学生的姓名。
7. （是否已经完成排序，\*\*）编写程序，提示用户输入一个列表（数组），然后显示该列表是否已经排好序。下面是一些运行示例。注意，输入中的第一个数表示列表中的元素个数。该数字不是列表的一部分。

Enter list: 8 10 1 5 16 61 9 11 1

The list is not sorted

Enter list: 10 1 1 3 4 4 5 7 9 11 21

The list is already sorted

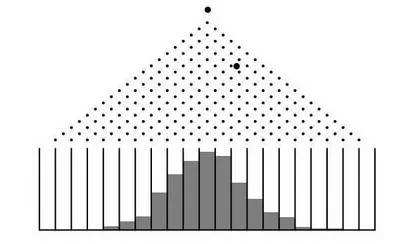
1. (游戏：豆机。三星题目)豆机，也称为梅花瓶或高尔顿瓶。它是一个用来做统计实验的设备，是用英国科学家“瑟弗兰克斯.高尔顿”的名字来命名的。它是一个均匀放置钉子（或钩子）的三角形直立板，如图1所示。

图1. 每个球都选取一个随机路径，然后掉入一个槽中

球都是从板子口落下的。每当球碰到钉子，它就有50%的机会落向左边或落向右边。在板子底部的槽子中都会累积一堆球。

编写程序，模拟豆机。程序应该提示用户输入球的个数以及机器的槽数。打印每个球的路径模拟它的下落（L左R右），并使用条形图显示槽中球的最终储备量。下面是程序的一个运行示例：

Enter the number of balls to drop: 5

Enter the number of slots in the bean machine: 8

LRLRLRR

RRLLLRR

LLRLLRR

RRLLLLL

LRLRRLR

0

0

000

提示，创建一个名为slots的数组。该数组中的每个元素存储的是相应槽中球的个数。每个球都经过一条路径落入一个槽中。路径上R的个数表示球落下的槽的位置。例如，对于路径LRLRLRR而言，球落到slots[4]中，而对路径RRLLLLL而言，球落在slots[2]中。

1. (储物柜难题，\*\*)一所学校有100个储物柜和100个学生。在开学第一天，所有的储物柜都是关着的。随着学生的到来，第一个学生（用S1表示）打开每个柜子。然后，第二个学生（用S2表示）从第二个柜子（用L2表示）开始，关闭相隔为1的柜子。学生S3从第三个柜子开始，改变每个（隔）第三个柜子的状态（如果它是开的就关上，如果它是关的就打开）。学生S4从柜子L4开始，改变每个（隔）第四个柜子的开闭状态。学生S5从L5开始，改变每个（隔）第5个柜子的状态。以此类推，直到学生S100改变L100位置。在所有学生都经过教学楼并且改变了柜子之后，哪些柜子是打开的？编写程序找出答案。提示，使用存放100个布尔型元素的数组。其中，每个元素都表明一个柜子是“开的”（true）还是“关的”（false）。在初始状态下，所有的柜子都是关的。
2. （优惠券收集人问题，\*\*）优惠券收集人问题是一个经典的统计问题。它有很多实际应用。这个问题是说，如果重复地从一组对象中拿出一个对象，然后找出要把所有需要拿出的对象都至少拿出来一次，那么需要拿多少次。从该问题衍生出的类似问题就是，从一副打乱的52张牌中重复选牌，找出在看到每种花色都有一张出现前，需要选多吃次。假设在选下一张牌之前的那张牌是背面向上的。编写程序，模拟要得到四张不同花色的牌所需要的选取次数。然后，显示选中的四张牌（有可能一张牌被选了两次）。下面是一个运行示例：

Queen of Spades（黑桃）

9 of Clubs（梅花）

Queen of Hearts

6 of Diamonds

Number of picks: 12

1. （组合，\*）编写一个程序，提示用户输入10个整数。然后，显示从这10个数中选出两个数的所有组合。
2. （选出四张牌，\*）编写一个程序，从一副52张的牌中选出4张牌，然后计算它们的和。Ace、King、Queen和Jack分别表示1、13、12和11。程序应该显示得到的和为24的选牌次数。
3. （模式识别方面：四个连续相等的数，\*）编写程序，测试某个数组是否有四个连续的元素的值相同（某个数组是否有连续四个相同的元素）。首先，程序应提示用户键入输入的大小，即列表中值的个数。然后，提示用户输入一个整数列表。如果这个列表中有四个连续的具有相同值的数，那么就显示true；否则，显示false。下面是一些运行示例。

Enter the number of values: 8

Enter the values: 3 4 5 5 5 5 4 5

The list has consecutive fours

Enter the number of values: 9

Enter the values: 3 4 5 5 6 5 5 4 5

The list has no consecutive fours

1. （合并两个有序列表，\*\*）编写程序，把两个有序列表合并成一个新的有序列表。程序提示用户输入两个有序列表，然后显示合并的列表。下面是一个运行示例。注意，输入的第一个数字表示列表中元素的个数。该数字不是列表的一部分。

Enter list1: 5 1 5 16 61 111

Enter list2: 4 2 4 5 6

The merged list is 1 2 4 5 5 6 16 61 111

1. （对字符串中的字符排序，\*\*）编写程序，对字符串的内容进行排序。例如，对“acb”排序后，得到“abc”。程序应提示用户输入一个字符串，显示排好序的字符串。
2. （猜字游戏，三星难度）编写一个猜字游戏。随机产生一个单词，提示用户一次猜测一个字母。单词中的每个字母显示为一个星号。当用户猜测正确后，显示正确的字母。当用户猜出一个单词时，显示猜错的次数，并且询问用户是否继续对另外一个单词进行游戏。提示，声明一个数组来存储单词。

// Add any words that you wish in this array

String words[] = {“write”,”that”,…};

下面是一个运行示例。

(Guess) Enter a letter in word \*\*\*\*\*\*\* > p

(Guess) Enter a letter in word p\*\*\*\*\*\* > r

(Guess) Enter a letter in word pr\*\*r\*\* > p

P is already in the word

(Guess) Enter a letter in word pr\*\*r\*\* > o

(Guess) Enter a letter in word pro\*r\*\* > g

(Guess) Enter a letter in word progr\*\* > n

n is not in the word

(Guess) Enter a letter in word progr\*\* > m

(Guess) Enter a letter in word progr\*m > a

The word is program. You missed 1 time

Do you want to guess another word? Enter y or n>

* 多维数组

1. （求矩阵各列数字的和，\*）编写程序，读取一个的矩阵，然后显示每列元素的和。下面是一个运行示例。

Enter a 3\*4 matrix row by row:

1.5 2 3 4

5.5 6 7 8

9.5 1 3 1

Sum of the elements at column 0 is 16.5

Sum of the elements at column 1 is 9.0

Sum of the elements at column 2 is 13.0

Sum of the elements at column 3 is 13.0

1. （求矩阵主对角线元素的和，\*）编写程序，读取一个的矩阵，然后显示它的主对角线上的所有元素的和。下面是一个运行示例。

Enter a 4\*4 matrix row by row:

1 2 3 4

5 6.5 7 8

9 10 11 12

13 14 15 16

Sum of the elements in the major diagonal is 34.5

1. （计算每个雇员每周工作的小时数，\*\*）假定所有雇员每周工作的小时数存储在一个二维数组中。每行将一个雇员7天的工作时间记录在7列中。例如，下面显示的数组存储了8位雇员的工作时间。编写一个程序，按照总工时降序的方式显示雇员和他们的总工时。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Su | M | T | W | Th | F | Sa |
| Employee 0 | 2 | 4 | 3 | 4 | 5 | 8 | 8 |
| Employee 1 | 7 | 3 | 4 | 3 | 3 | 4 | 4 |
| Employee 2 | 3 | 3 | 4 | 3 | 3 | 2 | 2 |
| Employee 3 | 9 | 3 | 4 | 7 | 3 | 4 | 1 |
| Employee 4 | 3 | 5 | 4 | 3 | 6 | 3 | 8 |
| Employee 5 | 3 | 4 | 4 | 6 | 3 | 4 | 4 |
| Employee 6 | 3 | 7 | 4 | 8 | 3 | 8 | 4 |
| Employee 7 | 6 | 3 | 5 | 9 | 2 | 7 | 9 |

1. 编写程序，实现两个矩阵的加法。提示用户输入两个的矩阵，然后显示它们的和。下面是一个运行示例。

Enter matrix1: 1 2 3 4 5 6 7 8 9

Enter matrix2: 0 2 4 1 4.5 2.2 1.1 4.3 5.2

The matrices are added as follows

1.0 2.0 3.0 0.0 2.0 4.0 1.0 4.0 7.0

4.0 5.0 6.0 + 1.0 4.5 2.2 = 5.0 9.5 8.2

7.0 8.0 9.0 1.1 4.4 5.2 8.1 12.3 14.2

1. （矩阵乘法，\*\*）编写程序，实现矩阵的乘法。提示用户输入两个的矩阵，然后显示它们的乘积。下面是一个运行示例。

Enter matrix1: 1 2 3 4 5 6 7 8 9

Enter matrix2: 0 2 4 1 4.5 2.2 1.1 4.3 5.2

The multiplication of the matrices is

1.0 2.0 3.0 0.0 2.0 4.0 5.3 23.9 24.0

4.0 5.0 6.0 \* 1.0 4.5 2.2 = 11.6 56.3 58.2

7.0 8.0 9.0 1.1 4.4 5.2 17.9 88.7 92.4

1. （距离最近的两个点，\*）编写程序，找出在三维空间上距离最近的两个点。使用一个二维数组表示这些点。使用下面的点来测试这个程序：double points[8][3] = {{-1,0,3},{-1,-1,-1},{4,1,1},{2,0.5,9},{3.5,2,-1},{3,1.5,3},{-1.5,4,2},{5.5,4,-0.5}};
2. （所有最近的点对，\*\*）编写程序，找出所有具有相同最小距离的点对。下面是一个运行示例。

Enter the number of points: 8

Enter 8 points: 0 0 1 1 -1 -1 2 2 -2 -2 -3 -3 -4 -4 5 5

The closest two points are (0.0, 0.0) and (1.0, 1.0)

The closest two points are (0.0, 0.0) and (-1.0, -1.0)

The closest two points are (1.0, 1.0) and (2.0, 2.0)

The closest two points are (-1.0, -1.0) and (-2.0, -2.0)

The closest two points are (-2.0, -2.0) and (-3.0, -3.0)

The closest two points are (-3.0, -3.0) and (-4.0, -4.0)

Their distance is 1.41421

1. （井字游戏，三星难度）在井字游戏中，两个玩家使用各自的标志（一方用X，另一方用O）轮流填写 网格中的某个空档。当一个玩家在网格的水平方向、垂直方向或者对角线方向上出现了三个相同的X或者三个相同的O时，游戏结束，该玩家获胜。平均（没有赢家）是指当网格中所有的空格都被填满时仍没有任何一方的玩家获胜（的情况）。创建一个玩井字游戏的程序。程序提示两个玩家可以选择X和O作为他们的标志。当输入一个标志时，程序在控制台上重新显示棋盘，然后确定游戏的状态（获胜、平局还是继续）。下面是一个运行示例：

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Enter a row (0, 1, or 2) for player X: 1

Enter a column (0, 1, or 2) for player X: 1

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | X |  |
|  |  |  |

Enter a row (0, 1, or 2) for player O: 1

Enter a column (0, 1, or 2) for player O: 2

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | X | O |
|  |  |  |

Enter a row (0, 1, or 2) for player X:

…

|  |  |  |
| --- | --- | --- |
| X |  |  |
|  | X | O |
|  |  | X |

X player won

1. （最大的行和列，\*）编写一个程序，在一个 的矩阵中随机填入0和1，打印该矩阵，找到（第一个）具有最多1的行和列。下面是一个程序的运行示例。

0011

0011

1101

1010

The largest row index: 2

The largest column index: 2

1. （游戏：九个正面和背面，\*\*）在一个的矩阵中放置9个硬币。这些硬币有些面向上，有些面向下。我们可以使用0（正面）或1（反面）表示硬币的状态。下面是一些例子：

0 0 0 1 0 1 1 1 0 1 0 1 1 0 0

0 1 0 0 0 1 1 0 0 1 1 0 1 1 1

0 0 0 1 0 0 0 0 1 1 0 0 1 1 0。

每个状态都可以用一个二进制数表示。例如，前面的矩阵对应到数字：000010000 101001100 110100001 101110100 100111110

这里，一共有512种可能性。所以，可以使用是十进制数0,1,2,3,…,511来表示这个矩阵的所有状态。编写一个程序，提示用户输入一个在0到511之间的数字，然后显示用字符H和T表示的对应矩阵。下面是一个运行示例。

Enter a number between 0 and 511: 7

H H H

H H H

T T T

用户输入7，它对应的二进制数是000000111。使用H代表0，T代表1，输出正确。

1. （探索矩阵，\*\*）编写程序，提示用户输入一个方阵的长度，随机地在矩阵中填入0和1，打印这个矩阵，然后找出整行、整列或者对角线都是0或1的行、列和对角线。下面是一个运行示例。

Enter the size for the matrix: 4

0111

0000

0100

1111

All 0s on row 1

All 1s on row 3

No same numbers on a column

No same numbers on the major diagonal

No same numbers on the sub-diagonal

1. （对二维数组排序，\*）编写一个程序，使用下面的方法对二维数组排序。这个方法首先按行排序，然后按列排序。例如，数组{{4,2},{1,7},{4,5},{1,2},{1,1},{4,1}} 将被排序为 {{1,1},{1,2},{1,7},{4,1},{4,2},{4,5}}。
2. （金融风暴，三星难度）银行间会相互借钱。在经济艰难时期，如果一个银行倒闭，那么它就不能偿还贷款。一个银行的总资产是它当前的余款减去它欠其他银行的贷款。下图是五个银行的状况图。每个银行的当前余额分别是2500万元、1亿2500万元、1亿7500万元、7500万元和1亿8100万元。从节点1到节点2的“方向边”表示“银行1”借给“银行2”共计4000万。



1



0



3



2



4



如果银行的总资产在某个限定范围以下，那么这个银行就是不安全的。它借的钱就不能返还给借贷方，而且这个借贷方也不能将这笔贷款算入它的总资产。因此，如果借贷方总资产在限定范围以下，那么它也不安全。

编写程序，找出所有不安全的银行。程序按照如下方式读取数据。首先，程序读取两个整数n和limit。这里，n表示银行个数，limit表示要保证银行安全的最小总资产。然后，程度会读取描述n个银行的n行信息，银行的id从0到n-1。每一行的第一个数字都是该银行的余额，第二个数字表明从该银行借款的银行（的数量）。其余的就都是由两个数字构成的数对。每对都描述一个借款方。每对数字中的第一个数字就是借款方的id，第二个数就是所借的钱数。例如，在上图中的五个银行的输入信息如下所示（注意，limit是201）：

5 201

25 2 1 100.5 4 320.5

125 2 2 40 3 85

175 2 0 125 3 75

75 1 0 125

181 1 2 125

银行3的总资产是75+125，这个数字在201以下。所以，银行3不安全。在银行3变得不安全之后，银行1的总资产也降为125+40。所以，银行1也不安全。程序的输出应该是：Unsafe banks are 3 1

提示，使用一个二维数组borrowers来表示贷款。borrowers[i][j]表明银行i贷款给银行j的贷款额。一旦银行j变得不安全，那么borrowers[i][j]就应该设置为0。

1. （连续四个相等的数，\*\*）编写程序，测试一个二维数组是否有四个连续的数字具有相同的值。这四个数可以是水平方向的、垂直方向的或者对角线方向的。程序提示用户输入一个二维数组的行数、列数以及数组中的值。如果这个数组有四个连续的数字具有相同的值，那么显示true；否则，显示false。下面是一些结果为true的例子：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 3 | 1 | 6 | 1 |  | 0 | 1 | 0 | 3 | 1 | 6 | 1 |  | 0 | 1 | 0 | 3 | 1 | 6 | 1 |
| 0 | 1 | 6 | 8 | 6 | 0 | 1 |  | 0 | 1 | 6 | 8 | 6 | 0 | 1 |  | 0 | 1 | 6 | 8 | 6 | 0 | 1 |
| 5 | 6 | 2 | 1 | 8 | 2 | 9 |  | 5 | 5 | 2 | 1 | 8 | 2 | 9 |  | 5 | 6 | 2 | 1 | 6 | 2 | 9 |
| 6 | 5 | 6 | 1 | 1 | 9 | 1 |  | 6 | 5 | 6 | 1 | 1 | 9 | 1 |  | 6 | 5 | 6 | 6 | 1 | 9 | 1 |
| 1 | 3 | 6 | 1 | 4 | 0 | 7 |  | 1 | 5 | 6 | 1 | 4 | 0 | 7 |  | 1 | 3 | 6 | 1 | 4 | 0 | 7 |
| 3 | 3 | 3 | 3 | 4 | 0 | 7 |  | 3 | 5 | 3 | 3 | 4 | 0 | 7 |  | 3 | 6 | 3 | 3 | 4 | 0 | 7 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 3 | 1 | 6 | 1 |
| 0 | 1 | 6 | 8 | 6 | 0 | 1 |
| 9 | 6 | 2 | 1 | 8 | 2 | 9 |
| 6 | 9 | 6 | 1 | 1 | 9 | 1 |
| 1 | 3 | 9 | 1 | 4 | 0 | 7 |
| 3 | 3 | 3 | 9 | 4 | 0 | 7 |

1. （游戏：四子连。三星难度。和158有关）四子连是一个两人棋盘游戏。在游戏中，玩家轮流把有颜色的棋子放入一个六行七列的垂直悬挂的网格中。这个游戏的目的是先于对手实现一行、一列或者一条对角线上有四个相同颜色的棋子。程序提示两个玩家交替地下红子Red或黄子Yellow。当放下一枚棋子时，程序在控制台重新显示这个棋盘，然后确定游戏的状态（赢、平局还是继续）。下面是一个运行示例。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Drop a red disk at column (0-6): 0

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |

Drop a yellow disk at column (0-6): 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| R |  |  | Y |  |  |  |

Drop a yellow disk at column (0-6): 6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  | R |  |  |  |
|  |  |  | Y | R | Y |  |
|  |  | R | Y | Y | Y | Y |
| R | Y | R | Y | R | R | R |

The yellow player won

1. （中心城市，\*）给定一组城市，中心城市是和所有其他城市之间具有最短距离的城市。编写一个程序，提示用户输入城市的数目以及城市的位置（坐标），找到中心城市以及和所有其他城市之间的总距离。下面是一个运行示例：

Enter the number of cities: 5

Enter the coordinates of the cities: 2.5 5 5.1 3 1 9 5.4 54 5.5 2.1

The central city is at (2.5, 5.0)

The total distance to all other cities is 60.81

1. （偶数个1，\*）编写一个程序，产生一个的二维矩阵，矩阵元素为随机的0或1，显示该矩阵，检测是否每行以及每列中有偶数个1。
2. （游戏，找到翻转的单元格，\*）假设给定一个填满0和1的矩阵，让用户翻转一个单元（即，从1翻成0或者从0翻成1），使得所有的行和列都有偶数个1。编写程序，找到哪个单元格被翻转了。程序提示用户输入一个的填满0和1的矩阵，找到不满足“具有偶数个1”特征的第一个r行以及第一个c列（即，1的数目不是偶数），则该翻转的单元格位于(r, c)上。下面是一个运行示例：

Enter a 6-by-6 matrix row by row:

1 1 1 0 1 1

1 1 1 1 0 0

0 1 0 1 1 1

1 1 1 1 1 1

0 1 1 1 1 0

1 0 0 0 0 1

The flipped cell is at (0, 1)

1. （马尔科夫矩阵，\*）一个的矩阵被称为一个“正马尔科夫矩阵”，当且仅当每个元素都是正数，并且每列元素的和为1。编写一个程序，检测一个的double矩阵是否是一个马尔科夫矩阵。下面是一些运行示例：

Enter a 3-by-3 matrix row by row:

0.15 0.875 0.375

0.55 0.005 0.225

0.30 0.12 0.4

It is a Markov matrix

Enter a 3-by-3 matrix row by row:

0.95 -0.875 0.375

0.65 0.005 0.225

0.30 0.12 -0.4

It is not a Markov matrix

1. （行排序，\*）编写一个程序，实现二维数组中的行排序。程序提示用户输入一个的double型矩阵，显示一个每行排好序的新矩阵。下面是一个运行示例：

Enter a 3-by-3 matrix row by row:

0.15 0.875 0.375

0.55 0.005 0.225

0.30 0.12 0.4

The row-sorted array is

0.15 0.375 0.875

0.005 0.225 0.55

0.12 0.30 0.4

1. （列排序，\*）编写一个程序，实现二维数组中的列排序。程序提示用户输入一个的double型矩阵，显示一个每列排好序的新矩阵。下面是一个运行示例：

Enter a 3-by-3 matrix row by row:

0.15 0.875 0.375

0.55 0.005 0.225

0.30 0.12 0.4

The column-sorted array is

0.15 0.005 0.225

0.3 0.12 0.375

0.55 0.875 0.4

1. （严格相同的数组）如果两个二维数组m1和m2中的相应元素都是相等的话，那么认为它们是“严格相同”（相等）。编写一个程序，提示用户输入两个的整型数组，显示这两个矩阵是否严格相同。下面是一些运行示例：

Enter list1: 51 22 25 6 1 4 24 54 6

Enter list2: 51 22 25 6 1 4 24 54 6

The two arrays are strictly identical

Enter list1: 51 25 22 6 1 4 24 54 6

Enter list2: 51 22 25 6 1 4 24 54 6

The two arrays are not strictly identical

1. （相同的数组）如果两个二维数组m1和m2具有相同的内容，那么它们是“相同”的。编写一个程序，提示用户输入两个的整型数组，显示这两个矩阵是否相同。下面是一些运行示例：

Enter list1: 51 25 22 6 1 4 24 54 6

Enter list2: 51 22 25 6 1 4 24 54 6

The two arrays are identical

Enter list1: 51 5 22 6 1 4 24 54 6

Enter list2: 51 22 25 6 1 4 24 54 6

The two arrays are not identical

1. （解线性方程组，\*）编写程序，求解线性方程组。例如，求解二元线性方程组。要求，使用二维数组保存系数，一维数组保存常数项。
2. （交点，\*）编写程序，求出两条直线的交点。假设(x1,y1)和(x2,y2)是直线1上的两个点，(x3,y3)和(x4,y4)位于直线2上。要求，把这些点的坐标保存在一个的矩阵points中。其中，points[0][0]、points[0][1] 代表 (x1,y1)。程序提示用户输入四个点的坐标，显示交点坐标。
3. （三角形面积，\*）编写程序，提示用户输入三角形的三个顶点的坐标，然后显示三角形的面积。如果三个点在一条直线上，那么面积记为0。要求，点坐标保存在一个矩阵points中。其中，points[0][0]、points[0][1] 代表 (x1,y1)。下面是一些运行示例：

Enter x1, y1, x2, y2, x3, y3: 2.5 2 5 -1.0 4.0 2.0

The area of the triangle is 2.25

Enter x1, y1, x2, y2, x3, y3: 2 2 4.5 4.5 6 6

The three points are on the same line

1. （多边形的子面积，\*）一个具有四个顶点的凸多边形可被分为四个三角形（如下图所示）。编写一个程序，提示用户输入四个顶点的坐标，然后以升序方式显示四个三角形的面积。下面是一个运行示例：



Enter x1, y1, x2, y2, x3, y3, x4, y4:

-2.5 2 4 4 3 -2 -2 -3.5

The areas are 6.17 7.96 8.08 10.42

1. （几何：最右下角的点，\*）在计算几何中，我们经常需要从一个点集中找出最右下角的点。编写程序，提示用户输入6个点的坐标，然后显示最右小角的点。下面是一个运行示例。

Enter 6 points: 1.5 2.5 -3 4.5 5.6 -7 6.5 -7 8 1 10 2.5

The rightmost lowest point is (6.5, -7.0)

1. （最大块，\*\*）给定一个元素为0或者1的方阵。编写程序，找到一个元素都为1的最大的子方阵。程序提示用户输入矩阵的行数。然后，显示最大子方阵的第一个元素（的位置），以及该子方阵的行数。下面是一个运行示例。

Enter the number of rows in the matrix: 5

Enter the matrix row by row:

1 0 1 0 1

1 1 1 0 1

1 0 1 1 1

1 0 1 1 1

1 0 1 1 1

The maximum square submatrix is at (2, 2) with size 3

1. （拉丁正方形，\*\*）拉丁正方形是一个的数组，由n个不同的拉丁字母填充，每个拉丁字母恰好只在每行和每列中出现一次。编写程序，提示用户输入数字n以及字符数组，如示例输出所示，检测该数组是否是一个拉丁正方形。字符是从A开始的前面n个字符。

Enter number n: 4

Enter 4 rows of letters separated by spaces:

A B C D

B A D C

C D B A

D C A B

The input array is a Latin square

Enter number n: 3

Enter 3 rows of letters separated by spaces:

A F D

Wrong input: the letters must be from A to C

1. （猜测首府，\*\*）编写程序，重复提示用户输入一个州的首府。当接收到用户输入后，程序报告答案是否正确。假设50个州以及它们的首府保存在一个二维数组中。程序提示用户回答所有州的首府，然后显示所有正确回答的数目（忽略英文字母的大小写）。

|  |  |
| --- | --- |
| Alabama | Montgomery |
| Alaska | Juneau |
| Arizona | Phoenix |
|  |  |
|  |  |

图n. 一个用于保存州以其首府的二维数组

下面是一个运行示例。

What is the capital of Alabama? Montogomery

The correct answer should be Montgomery

What is the capital of Alaska? Juneau

Your answer is correct

What is the capital of Arizona?

The correct count is 35

* 字符数组、正则表达式

1. 想不出什么题目让大家试试。用好正则表达式的关键在于设计出优秀的pattern。这个是艺术，不是说多写几遍代码就能出来的。使用按照规定走就好了。另外，对于大一的学生，是否需要介绍“正则表达式”？毕竟，教材里没有这方面的内容。或者，放到最后再介绍，免得没东西讲。

* 方法

1. （五角数）一个五角数被定义为 n(3n-1)/2，其中n=1,2,…。所以，前四个五角数分别是1,5,12,22。编写下面的函数返回一个五角数：

int getPentagonalNumber(int n)。

编写一个测试程序，显示前100个五角数，每行显示10个。

1. （求一个正整数各位数字之和）编写一个函数，计算一个正整数各位数字之和。函数的声明如下：int sumDigits(int n)。例如，sumDigits(234)返回9。编写测试程序，提示用户输入一个正整数，然后显示这个整数所有位数上的数字的和。
2. （回文整数，\*\*）使用下面的声明编写两个函数：

//返回一个整数的反序形式。例如，reverse(456)返回654

int reverse(int number);

//如果number是回文整数，那么返回true

bool isPalindrome(int number)

使用reverse函数实现isPalindromw。如果一个数字的反向倒置数和它一样，那么这个数就称作回文数。编写一个测试程序，提示用户输入一个整数值，然后报告这个整数是否是回文数。

1. （反向显示一个正整数，\*）使用下面的函数声明，编写函数，显示一个整数：void reverse(int number);。例如，reverse(3456)显示6543。编写一个测试程序，提示用户输入一个整数，然后显示它的反向数。
2. (显示图案，\*)编写函数，显示如下图案：

1

2 1

3 2 1

…

n n-1 … 3 2 1

该函数的函数头为 void displayPattern(int n)。

1. （计算未来投资价值，\*）编写一个函数，计算按照给定的年数和利率计算未来投资值。函数的声明语句如下：

double futureInvestmentValue(double investmentAmount,

double monthlyInterestRate,

int years);

例如，futureInvestmentValue(10000,0.05/12,5)返回12833.59。编写一个测试程序，提示用户输入投资额（例如，1000）、利率（例如，9%），然后打印年份从1到30年的未来投资值。下面是一个运行示例：

The amount invested: 1000

Annual interest rate: 9

Years Future Value

1 1093.80

2 1196.41

…

29 13467.25

30 14730.57

注：

1. (摄氏度和华氏度之间的转换)根据下面的函数声明，编写两个函数：

//摄氏度转换为华氏度

double celsiusToFahrenheit(double Celsius);

//华氏度转换为摄氏度

double fahrenheitTocelsius(double fahrenheit);

转换公式如下：

华氏度 = (9.0 / 5) \* 摄氏度 + 32

摄氏度 = (5.0 / 9) \* (华氏度 - 32)

编写一个测试程序，调用这两个函数来显示如下表格。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 摄氏度 | 华氏度 |  | 华氏度 | 摄氏度 |
| 40.0 | 104.0 |  | 120.0 | 48.89 |
| 39.0 | 102.2 |  | 110.0 | 43.33 |
| … |  |  |  |  |
| 32.0 | 89.6 |  | 40.0 | 4.44 |
| 31.0 | 87.8 |  | 30.0 | -1.11 |

1. （英尺和米之间的转换）编写如下的两个函数：

double footToMeter(double foot);//英尺转换为米

double meterToFoor(double meter);//米转换为英尺

转换公式如下：

米 = 0.307 \* 英尺

英尺 = 3.279 \* 米

编写一个测试程序，调用这两个函数来显示如下表格。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 英尺 | 米 |  | 米 | 英尺 |
| 1.0 | 0.305 |  | 20.0 | 65.574 |
| 2.0 | 0.610 |  | 25.0 | 81.967 |
| … |  |  |  |  |
| 9.0 | 2.745 |  | 60.0 | 196.721 |
| 10.0 | 3.050 |  | 65.0 | 213.115 |

1. （判断素数）编写一个函数，判断一个正整数是否是素数。函数声明如下所示：bool isPrime(int number)。编写一个测试程序，使用这个函数，求出小于10000的素数的个数。
2. （计算酬金）编写一个函数，计算酬金。函数声明如下所示：

double computeCommission(double salesAmount);。

酬金计算方案如下：

|  |  |
| --- | --- |
| 销售额 | 提成率 |
| 0.01 ~ 5000 | 8% |
| 5000.01 ~ 10000 | 10% |
| 10000.01 及以上 | 12% |

编写一个测试程序，显示下面表格：

|  |  |
| --- | --- |
| 销售总额 | 酬金 |
| 10000 | 900.0 |
| 15000 | 1500.0 |
| … |  |
| 95000 | 11100.0 |
| 100000 | 11700.0 |

1. （显示字符）编写一个打印字符的函数。函数的声明如下所示：

void printChars(char ch1, char ch2, int numberPerLine)

该函数打印从ch1到ch2之间的字符（含ch1和ch2），每行按指定个数numberPerLine打印。编写一个测试程序，打印从’1’到’Z’的字符，每行打印10个。字符之间使用一个空格字符隔开。

1. （数列求和，\*）编写一个函数，计算下列级数：

编写一个测试程序，显示下面的表格：

|  |  |
| --- | --- |
| i | m(i) |
| 1 | 0.5000 |
| 2 | 1.1667 |
| … |  |
| 19 | 16.4023 |
| 20 | 17.3546 |

1. （估算PI，\*） 可以使用下面的数列进行计算：

编写一个函数，对于给定的i返回m(i)。编写一个测试程序，显示如下表格：

|  |  |  |
| --- | --- | --- |
| i |  | m(i) |
| 1 |  | 4.0000 |
| 101 |  | 3.1515 |
| 201 |  | 3.1466 |
| 301 |  | 3.1449 |
| 401 |  | 3.1441 |
| 501 |  | 3.1436 |
| 601 |  | 3.1433 |
| 701 |  | 3.1430 |
| 801 |  | 3.1428 |
| 901 |  | 3.1427 |

1. （一年的天数，\*）编写一个函数，返回一年的天数。函数的声明如下所示：int numberOfDaysInAYear(int year)。编写一个测试程序，显示从2000年到2020年每年的天数。
2. （显示0和1构成的矩阵）编写一个函数，显示一个由0和1组成的n阶方阵。函数的声明如下所示：void printMatrix(int n)。每个元素都是随机产生的0或1。编写一个测试程序，提示用户输入n，显示如下所示的n阶方阵。

Enter n: 3

0 1 0

0 0 0

1 1 1

1. （检测密码，\*\*）一些网站对于密码的设定具有一些规则。编写一个函数，检测字符串是否是一个有效密码。假设密码规则如下：
   * 密码必须至少8位字符
   * 密码仅能包含字母和数字
   * 密码必须至少包含两个数字

编写一个程序，提示用户输入一个密码。如果符合规则，那么显示valid password；否则，显示invalid password。

1. （计算三角形的面积，\*）根据如下函数声明，编写两个函数：

// 判断三条边是否能组成一个三角形

bool isValid(double side1,double side2,double side3);

//计算三角形的面积

double area(double side1,double side2,double side3);

编写一个测试程序，读入三角形三边的值。若输入有效，那么计算面积；否则，显示输入无效。

注， 。

1. （计算一个字符串中字面的个数，\*）编写一个函数，计算字符串中的字母个数。函数的声明如下所示：int countLetters(String s);。编写一个测试程序，提示用户输入字符串，然后显示字符串中的字母个数。
2. （平方根的近似求法，\*\*）巴比伦法求解算术平方根。它通过反复使用下面的公式，近似计算得到 n 的算术平方根：

nextGuess = (lastGuess + n / lastGuess)/2

当nextGuess和lastGuess几乎相同的时候，nextGuess就是n的算术平方根的近似值。最初的猜测值可以是任意一个正值（例如，1）。这个值就是lastGuess的初始值。如果nextGuess和lastGuess的差小于一个很小的数（例如，0.0001），那么就可以认为nextGuess是n的平方根的近似值；否则，nextGuess就成为lastGuess，近似迭代过程继续执行。实现下面的函数，返回n的平方根：double sqrt(double n)。

1. （指定字符的出现次数，\*）编写一个函数，返回一个字符串中指定字符的出现次数。函数声明如下所示：int count(String str,char a)。例如，count(“Welcome”,’e’) 返回2。编写一个测试程序，提示用户输入一个字符串以及一个字符，显示该字符在字符串中出现的次数。
2. （将毫秒数转换成小时数、分钟数和秒数，\*\*）编写一个函数，将毫秒数转换成小时数、分钟数和秒数。函数的声明如下所示：

String convertMillis(long millis);。

这个函数返回形如“小时：分钟：秒”的字符串。例如，convertMillis(5500)返回字符串“0:0:5”， convertMillis(100000)返回字符串“0:1:40”， convertMillis(555550000)返回字符串“154:19:10”。注，1秒 = 1000 毫秒。

1. （回文素数，\*\*）回文素数是指一个数字同时为素数和回文数。例如，131是一个回文素数。数字313和757也是回文素数。编写程序，显示前100个回文素数。每行显示10个数，数字中间用空格隔开。下面是一个运行示例。
2. 3 5 7 11 101 131 151 181 191

313 353 373 383 727 757 787 797 919 929

……

1. （反素数，\*\*）反素数（反转拼写的素数）是指一个非回文素数，将其反转之后（得到的数字）也是一个素数。例如，17是一个素数，而71也是一个素数，所以17和71是反素数。编写程序，显示前100个反素数。每行显示10个，并且数字间用空格隔开。下面是一个运行示例。

13 71 31 37 71 73 79 97 107 103

149 157 167 179 199 311 337 347 359 389

…

1. （梅森素数，\*\*）如果一个素数可以写成 的形式，其中p是某个正整数，那么这个素数就称作梅森素数。编写程序，找出 的所有梅森素数，然后显示如下的输出结果：

|  |  |
| --- | --- |
| p | 2^p - 1 |
| 2 | 3 |
| 3 | 7 |
| 5 | 31 |
| … |  |

。

1. （双素数，\*\*）双素数是指一对差值为2的素数。例如，3和5就是一对双素数，5和7是一对双素数，11和13也是一对双素数。编写程序，找出小于1000的所有双素数。显示结果如下所示。

（3,5）

（5,7）

…

1. （双骰子游戏，\*\*）掷双骰子游戏是一种非常流行的骰子游戏。编写程序，玩这个游戏的一个变种。玩法如下所示：掷两个骰子，每个骰子有六个面，分别表示值1，2，…，6。检查这两个骰子的和。如果和为2、3或12（称为掷骰子（craps）），那么你就输了；如果和是7或者11（称作自然（natural）），那么你就赢了；但是，如果和是其他数字（例如，4、5、6、8、9或者10），那么就确定一个点。继续掷骰子，直到掷出一个7或者掷出和刚才相同的点数为止。如果掷出的是7，那么你就输了。如果掷出的点数和你前一次掷出的点数相同，那么你就赢了。程序扮演一个独立的玩家。下面是一些运行示例。

You rolled 5 + 6 = 11

You win

You rolled 1 + 2 = 3

You lose

You rolled 4 + 4 = 8

Point is 8

You rolled 6 + 2 = 8

You win

Your rolled 3 + 2 = 5

Point is 5

You rolled 2 + 5 = 7

You lose

1. （信用卡号的合法性，\*\*）信用卡号遵循下面的模式。一个信用卡号必须是13到16位的整数。它的开头必须是:
   * 4，指Visa卡
   * 5，指Master卡
   * 37，指American Express卡
   * 6，指Discover卡

在1954年，IBM的Hans Luhn提出一种算法，该算法可以验证信用卡号的有效性。这个算法在确定输入的卡号是否正确，或者这张信用卡是否被扫描仪正确扫描等方面是非常有用的。遵循这个合法性检测可以生成所有的信用卡号，通常称为Luhn检测或者Mod 10检测。具体算法如下所示（为了方便解释，假设卡号为4388576018402626）：

1. 从左到右，把每个奇数位上的数字翻倍。如果对某个数字翻倍之后的结果是一个两位数，那么就将这两位加在一起得到一位数。

4388576018402626

2\*2 = 4

2\*2 = 4

4\*2 = 8

1\*2 = 2

6\*2 = 12 (1 + 2 = 3)

5\*2 = 10 (1 + 0 = 1)

8\*2 = 16 (1 + 6 = 7)

4\*2 = 8

1. 把第一步得到的所有一位数相加：4+4+8+2+3+1+7+8=37。
2. 从左到右，把偶数位上的所有数字相加；6+6+0+8+0+7+8+3=38。
3. 把第二步和第三步得到的结果相加：37 + 38 = 75。
4. 如果第四步得到的结果能被10整除，那么卡号是合法的；否则，卡号是不合法的。例如，号码4388576018402626是非法的；号码4388576018410707是合法的。

编写程序，提示用户输入一个long型整数的信用卡号码，显示这个号码是否合法。使用下面的函数设计程序：

//return true if the card number is valid

bool isValid(long number);

//get the result from step2

int sumOfDoubleOddPlace(long number);

/\*return this x if it is a single digit, otherwise, return the sum of the two digits\*/

int getDigit(int x);

//return sum of even-place digits in number

int sumOfEvenPlace(long number);

//return true if the digit d is a prefix for number

bool prefixMatched(long number, int d);

//return the number of digits in d

int getSize(long d);

/\*return the first k number of digits from number. If the number of digits in number is less than k, return number.\*/

long getPrefix(long number,int k);

下面是一些运行示例（我们也可以将输入作为一个字符串读入，通过对字符串进行处理来验证信用卡卡号。）。

Enter a credit card number as a long integer: 4388576018402626

4388576018402626 is invalid

Enter a credit card number as a long integer: 4388576018410707

4388576018410707 is valid

1. （赢钱双骰子游戏的机会，\*\*）修改练习题222，使该程序运行10000次，然后显示赢得游戏的次数。
2. （五边形的面积）编写一个函数，返回五边形的面积。函数声明如下：double area(double side);。面积计算公式： 。编写一个测试程序，提示用户输入五边形的边长，然后显示它的面积。下面是一个运行示例。

Enter the side: 5.5

The area of the pentagon is 52.04

1. （正多边形的面积，\*）编写一个函数，返回正多边形的面积。函数声明如下：double area(int n,double side);。正多边形的面积计算公式： 。编写一个测试程序，提示用户输入正多边形边的个数以及边长，然后显示它的面积。下面是一个运行示例。

Enter the number of sides: 5

Enter the side: 6.5

The area of the polygon is 72.69

1. （格式化整数）编写一个函数，将整数格式化为指定宽度。函数声明如下：String format(int number, int width);。函数为数字number返回一个带有若干以0为前缀的字符串。字符串的长度就是宽度。例如，format(34,4)返回0034，format(34,5)返回00034。如果数字宽于指定宽度，那么函数返回该数字的字符串表示。例如，format(34,1)返回34。编写一个测试程序，提示用户输入一个数字以及宽度，显示通过调用format(number,width)返回的字符串。
2. （生成随机字符，\*）编写程序，随机打印100个大写字母和100个数字，每行显示10个。
3. （点的位置）编写程序，测试一个点是否在一个有向直线的左侧、右侧或在该直线上。使用下面的函数声明，实现相应的函数。

bool leftOfTheLine(double x0,double y0,double x1,double y1,double x2,double y2);

bool oneTheSameLine(double x0,double y0,double x1,double y1,double x2,double y2);

bool rightOfTheLine(double x0,double y0,double x1,double y1,double x2,double y2);

注，给定一个从点p0(x0,y0)到p1(x1,y1)的有向线段，我们可以使用下面的条件来确定点p2(x2,y2)是在线段的左侧、右侧或者在该直线上：

1. （找出最小元素）编写一个函数，求出一个整数数组中的最小元素。函数声明如下所示：double min(double a[],int n);。编写一个测试程序，提示用户输入10个数字，调用这个方法返回最小值，显示其最小值。下面是一个运行示例。

Enter ten numbers: 1.9 2.5 3.7 2 1.5 6 3 4 5 2

The minimum number is 1.5

1. （找出最小元素的下标）编写一个函数，求出整数数组中最小元素（第一次出现）的下标（索引值）。函数声明如下所示：

int indexOfSmallestElement(double a[],int n);。

1. （计算标准差，\*）编写函数，计算标准差。函数的声明如下所示：

// compute the deviation of double values

double deviation(double x[],int n);

//computer the mean of an array of double values

double mean(double x[],int n);

编写一个测试程序，提示用户输入10个数字，然后显示平均值和标准差。下面是一个运行示例。

Enter ten numbers: 1.9 2.5 3.7 2 1 6 3 4 5 2

The mean is 3.11

The standard deviation is 1.55738

1. （是否排好序，\*\*）编写函数，判断参数中的list数组是否已经按照升序方式排好。函数声明如下所示：

bool isSorted(int list[],int n);。

编写一个测试程序，提示用户输入一个数组，显示该数组是否已经排好序。下面是一些运行示例。

Enter list: 10 1 5 16 61 9 11 1

The list is not sorted

Enter list: 1 1 3 4 4 5 7 9 11 21

The list is already sorted

1. （求矩阵各列数字的和，\*）编写一个函数，计算整数矩阵中特定列的所有元素的和。函数声明如下：double sumColumn(double m[][4],int rowSize, int columnIndex);。

编写测试程序，读取一个的矩阵，然后显示每列元素的和。下面是一个运行示例。

Enter a 3\*4 matrix row by row:

1.5 2 3 4

5.5 6 7 8

9.5 1 3 1

Sum of the elements at column 0 is 16.5

Sum of the elements at column 1 is 9.0

Sum of the elements at column 2 is 13.0

Sum of the elements at column 3 is 13.0

1. （求矩阵主对角线元素的和，\*）编写一个函数，求n阶方阵中主对角线上所有元素的和。函数声明如下所示：

double sumMajorDiagonal(double m[][4],int rowSize);。

编写程序，读取一个的矩阵，然后显示它的主对角线上的所有元素的和。下面是一个运行示例。

Enter a 4\*4 matrix row by row:

1 2 3 4

5 6.5 7 8

9 10 11 12

13 14 15 16

Sum of the elements in the major diagonal is 34.5

* 递归

1. （计算阶乘，\*）编写一个函数，使用递归计算阶乘。编写一个测试程序，提示用户输入一个（10以内的）整数，然后显示它的阶乘。
2. （斐波那契数列，\*）编写一个函数，使用递归计算斐波那契数列的第n项。其中，f0=1，f1=1。编写一个测试程序，提示用户输入一个索引，然后显示它的斐波那契数。
3. （求最大公约数，\*）求最大公约数的gcd(m,n)方法也可以如下递归地定义：
   * 如果 m % n 为0，那么gcd(m,n)的值为n；
   * 否则，gcd(m,n)就是gcd(n, m%n)。

编写一个递归方法，求m和n的最大公约数。编写一个测试程序，提示用户输入两个正整数，显示它们的最大公约数。

1. （对数列求和）编写一个递归方法，计算下面的级数：

编写一个测试程序，为 i=1,2,…,10显示m(i)。

1. （对数列求和）编写一个递归方法，计算下面的级数：

编写一个测试程序，为 i=1,2,…,10显示m(i)。

1. （斐波那契数列，\*）修改练习238，使程序可以找出调用fib函数的次数。提示，使用一个静态变量，每当调用这个方法的时候，该变量就加1。
2. （以逆序输出一个整数中的数字，\*）编写一个递归方法，在控制台上以逆序显示一个int型的值。函数的声明如下所示：

void reverseDisplay(int value);。

例如，reverseDisplay(12345)显示的是54321。编写一个测试程序，提示用户输入一个正整数，然后显示它的逆序数字。

1. （以逆序输出一个字符串中的字符，\*）编写一个递归函数，在控制台上以逆序显示一个字符串。函数的声明如下所示：

void reverseDisplay(String value);。

例如，reverseDisplay(“abcd”)显示的是dcba。编写一个测试程序，提示用户输入一个字符串，然后显示它的逆序字符串。

1. （字符串中某个指定字符出现的次数，\*）编写一个递归函数，返回一个指定字符在字符串中出现的次数。函数的声明如下所示：

int count(String str, char a);。

例如，count(“Welcome”,’e’)返回2。编写一个测试程序，提示用户输入一个字符串和一个字符，显示该字符在字符串中出现的次数。

1. （使用递归求一个正整数各位数之和，\*）编写一个递归方法，计算一个正整数中各位数之和。函数声明如下所示：int sumDigits(int n);。例如，sumDigits(234)返回的是9（2+3+4=9）。编写一个测试程序，提示用户输入一个正整数，然后显示各位数字之和。
2. （找出数组中的最大数，\*）编写一个递归方法，返回一个数组中的最大值。编写一个测试程序，提示用户输入一个包含8个整数的数组，然后显示数组的最大值。
3. （求字符串中大写字母的个数，\*）编写一个递归方法，返回一个字符串中大写字母的个数。编写一个测试程序，提示用户输入一个字符串，然后显示该字符串中大写字母的数目。
4. （汉诺塔，\*）编写一个递归方法，实现汉诺塔，并且可以计算n个盘子移动的次数。提示，使用静态变量，每当调用方法一次，该变量就加1。
5. （将十进制数转换为二进制数，\*）编写一个递归方法，将一个十进制数转换为二进制数的字符串。函数的声明如下所示：

String dec2Bin(int value);。

编写一个测试程序，提示用户输入一个十进制数，然后显示等价的二进制数。

1. （将十进制数转换为十六进制数，\*）编写一个递归方法，将一个十进制数转换为一个十六进制数的字符串。函数的声明如下所示：

String dec2Hex(int value);。

编写一个测试程序，提示用户输入一个十进制数，然后显示等价的十六进制数。

1. （将二进制数转换为十进制数，\*）编写一个递归方法，将一个字符串形式的二进制数转换为一个十进制数。函数的声明如下所示：

int bin2Dec(String binaryStr);。

编写一个测试程序，提示用户输入一个二进制字符串，然后显示等价的十进制数。

1. （将十六进制数转换为十进制数，\*）编写一个递归方法，将一个字符串形式的十六进制数转换为一个十进制数。函数的声明如下所示：

int hex2Dec(String hexStr);。

编写一个测试程序，提示用户输入一个十六进制字符串，然后显示等价的十进制数。

1. （字符串排列，\*\*）编写一个递归方法，输出一个字符串的所有排列。例如，对于字符串abc，输出为：abc acb bac bca cab cba。提示，定义下面两个方法。其中，第二个方法是一个辅助方法。

void displayPermutation(String s);

void displayPermutation(String s1, String s2);

第一个方法简单地调用displayPermutation(“”,s)。第二个方法使用循环，将一个字符从s2移到s1，并使用新的s1和s2递归调用该方法。基础情况是当s2为空的时候，将s1打印到控制台上。

编写一个测试程序，提示用户输入一个字符串，然后显示它的所有排列。

1. 指针、引用、链表
2. 采用教材上的例题就好了。动态数组、修改排序顺序、计算不同被积函数的定积分，建立链表、删除结点、插入结点、遍历结点。这些让学生自己摸一下，有个概念就好了。
3. 文件读写
4. （统计一个文件中的字符数、单词数和行数，\*）编写一个程序，统计一个文件中的字符数、单词数和行数。单词由空格符分隔，文件名通过键盘输入。下面是一个运行示例。

Enter file name: Loan.data

File Load.data has

1919 characters

210 words

71 lines

1. （处理文本文件中的成绩，\*）假定一个文本文件中包含未指定个数的学生成绩，成绩之间以空格分开。编写一个程序，提示用户输入文件，然后从文件中读入成绩，显示它们的和以及平均值。
2. （写/读数据，\*）编写程序，把随机产生的100个整数写入文件exercise\_240.txt，数字之间由空格分开。然后，从文件中读取这些数据并以升序显示它们。
3. （替换文本，\*\*）编写程序，用一个新字符串替换文本文件中的某个字符串文本，并将替换后的新文本存储到一个新文件中。
4. （猜单词游戏，\*\*\*）改写练习题165。程序读取存储在一个名为 hangman.txt 的文本文件中的单词，这些单词用空格隔开。
5. （统计单词，\*）编写一个程序，统计Abraham Lincoln总统的Gettysburg演讲中的单词数。该演讲的网址为 http://cs.armstrong.edu/liang/data/Lincoln.txt
6. （数据排好序了吗，\*）编写一个程序，从文件sortedStrings.txt中读取字符串，并且报告“文件中的字符串是否以升序的方式进行存储”。如果文件中的字符串没有排好序，那么显示没有排序的前两个字符串。
7. （处理web上的文本文件中的成绩，\*\*）假定Web上的一个文本文件<http://cs.armstrong.edu/liang/data/Scores.txt>中包含了不确定数目的成绩。编写一个程序，从该文件中读取成绩，显示它们的总和以及平均数。文件中的成绩用空格隔开。
8. （创建大的数据集，\*）编写程序，创建一个具有1000行的数据文件。文件中的每行包含了一个教职员的姓、名、级别以及薪水。第i行的教职工的姓和名为FirstNamei和LastNamei。级别随机产生为assistant（助理）、associate（副）以及full（正）。薪水为随机产生的数字，并且保留两位小数。对于助理而言，薪水应该在50000 到 80000之间；副职为60000 到110000之间；正职为75000到130000之间。保存文件为Salary.txt。下面是一些示例数据。

FirstName1 LastName1 assistant 60055.95

FirstName2 LastName2 associate 81112.45

…

FirstName1000 LastName1000 full 92255.21

1. （处理大的数据集，\*）一个大学将其教职员工的薪水发布在<http://cs.armstrong.edu/liang/data/Salary.txt>中。文件中的每行包含一个教职员工的姓、名、级别以及薪水（见练习246）。编写一个程序，分别显示（所有教职员工）三个级别（助理、副职、正职）的总薪水和平均薪水。
2. （每个字母出现的次数，\*\*）编写一个程序，提示用户输入一个文件名，然后显示该文件中每个字母出现的次数。字母是大小写敏感的。下面是一个运行示例。

Enter a filename: Loan.data

Number of A’s: 56

…

Number of Z’s: 9

1. 对象和类
2. （矩形类Rectangle）设计一个名为Rectangle的类，表示矩形。这个类包括：
   1. 两个名为width和height的double型数据域。它们分别表示矩形的宽和高。width和height的默认值都为1。
   2. 创建默认构造函数（无参构造函数）
   3. 创建一个width和height为指定值的构造函数
   4. 一个名为getArea()的方法，返回这个矩形的面积。
   5. 一个名为getPerimeter()的方法，返回周长。

编写一个测试程序，创建两个Rectangle对象：一个矩形的宽为4、高为40；另一个矩形的宽为3.5、高为35.9。然后，按照这个顺序，显示每个矩形的宽、高、面积和周长。

1. （股票类Stock）设计一个名为Stock类，表示股票。这个类包括：
2. 一个名为symol的字符串数据域，表示股票代码。
3. 一个名为name的字符串数据域，表示股票名字。
4. 一个名为previousClosingPrice的double型数据域，存储前一天的股票价格。
5. 一个名为currentPrice的double型数据域，存储当天的股票价格。
6. 创建一个可指定股票代码和股票名字的构造函数。
7. 一个名为getChangePercent()的方法，返回从previousClosingPrice变化到currentPrice的百分比。

编写一个测试程序，创建一个Stock对象，它的股票代码是600519，股票名字为“贵州茅台”，前一日收盘价是679.96。设置新的当前值为688.95，然后显示市值变化的百分比。

注，上面的这道两题不考虑访问控制级别。使用struct创建新类型。

1. （秒表，\*）设计一个名为StopWatch的类。这个类包括：
   1. 具有访问器方法（get方法）的私有数据域startTime和endTime。
   2. 一个无参构造函数，使用当前时间来初始化startTime。
   3. 一个名为start()的方法，将startTime重设为当前时间。
   4. 一个名为stop()的方法，将endTime设置为当前时间。
   5. 一个名为getElapsedTime()的方法，以毫秒为单位返回秒表记录的流逝。

编写一个测试程序，用于测量使用选择排序对100000个数字进行排序的执行时间。

1. （账户类Account）设计一个名为Account的类。这个类包括：
   1. 一个名为id的int型私有数据域，默认值为0。
   2. 一个名为balance的double型私有数据域，默认值为0。
   3. 一个名为annualInterestRate对的double型私有数据域，用于存储当前利率，默认值为0。假设所有的账户都有相同的利率。
   4. 一个名为dateCreated的Date类型的私有数据域，存储用户的开户日期。
   5. 一个用于创建默认账户的无参构造函数。
   6. 一个用于创建带特定id和初始余额的构造函数。
   7. id、balance和annualInterestRate的访问器（get方法）和修改器（set方法）。
   8. dateCreated的访问器。
   9. 一个名为getMonthlyInterestRate()的方法，返回月利率。
   10. 一个名为getMonthlyInterest()的方法，返回月利息。
   11. 一个名为withDraw()的方法，从账户提取特定数额的钱款。
   12. 一个名为deposit()的方法，向账户中存储特定数额的钱款。

编写一个测试程序，创建一个账户ID为1122、余额为20000元、年利率为4.5%的Account对象。然后，使用withdraw方法取款2500元；再使用deposit方法存款3000元。最后，打印余额、月利息以及这个账户的开户日期。

提示：方法getMonthlyInterest()用于返回月利息，而不是利率。月利息是balance \* monthlyInterestRate。monthlyInterestRate是annualInterestRate / 12。annualInterestRate是一个百分数，例如4.5%。因此，我们需要将其除以100。

1. （风扇类Fan）设计一个名为Fan的类，表示一个风扇。这个类包括：
   1. 三个名为SLOW、MEDIUM和FAST而值为1、2和3的常量，表示风扇的速度。
   2. 一个名为speed的int型私有数据域，表示风扇的速度，默认值为SLOW。
   3. 一个名为on的布尔型私有数据域，表示风扇是否打开，默认值为false。
   4. 一个名为radius的double型私有数据域，表示风扇的半径，默认值为5。
   5. 一个名为color的string型数据域，表示风扇的颜色，默认值为blue。
   6. 这四个数据域的访问器（get方法）和修改器（set方法）。
   7. 一个创建默认风扇的无参构造函数
   8. 一个名为toString()的方法，返回描述风扇的字符串。如果风扇是打开的，那么该方法返回一个由风扇速度、颜色和半径等内容组成的字符串。如果没有打开，那么返回一个由“fan is off”和风扇颜色及半径等内容组成的字符串。

编写一个测试程序，创建两个Fan对象。将第一个对象设置为最大速度、半径为10、颜色为yellow、状态为打开。将第二个对象设置为中等速度、半径为5、颜色为blue、状态为关闭。通过调用它们的toString()方法，显示这些对象。

1. （正n边形，\*\*）在一个正n边形中，所有边的长度都相同，且所有角的度数都相同（即，这个多边形是等边等角的）。设计一个名为RegularPolygon的类。这个类包括：
   1. 一个名为n的int型私有数据域，用于定义多边形的边数，默认值为3。
   2. 一个名为side的double型私有数据域，用于存储边的长度，默认值为1。
   3. 一个名为x的double型私有数据域，用于存储多边形中心点的x坐标，默认值为0。
   4. 一个名为y的double型私有数据域，用于存储多边形中心点的y坐标，默认值为0。
   5. 一个创建带默认值的正多边形的无参构造函数。
   6. 一个能创建带指定边数、边长、中心在(0,0)处的构造函数。
   7. 一个能创建带指定边数、边长、中心在(x,y)处的构造函数。
   8. 所有数据域的访问器和修改器。
   9. 一个返回多边形周长的方法getPerimeter()。
   10. 一个返回多边形面积的方法getArea()。计算正多边形面积的公式是： 。

编写一个测试程序，分别使用无参构造方法、RegularPolygon(6,3)和RegularPolygon(10,4,5.6,7.8)，创建三个RegularPolygon对象。然后，显示每个对象的周长和面积。

1. （二次方程式，\*）为一元二次方程，设计一个名为QuadraticEquation的类。这个类包括：
   1. 代表三个系数的私有数据域a、b和c。
   2. 一个参数为a、b和c的构造函数。
   3. 数据域a、b和c的三个get()方法。
   4. 一个名为getDiscriminant()的方法，返回判别式 的结果。
   5. 两个名分别为getRoot1()和getRoot2()的方法，返回方程的两个根。这些方法只有在 的时候，才有效；否则，这些方法返回0。

编写一个测试程序，提示用户输入a、b和c的值。然后，显示判别式的结果。如果判别式大于0，那么显示两个根；如果判别式等于0，那么显示一个根；否则，显示“The equation has no roots.”。

1. （位置类Location，\*\*）设计一个名为Location的类，定位二维数组中的最大值及其位置。这个类包括公共的数据域row、column和maxValue，二维数组中最大值及其索引值分别用double型的maxValue、int型的row和column存储。编写下面的方法，返回一个二维数组中最大值的位置：Location locateLargest(double a[][4],int row);。注，返回值是一个Location的实例（对象）。

编写一个测试程序，提示用户输入一个二维数组，然后显示这个数组中最大元素及其位置。下面是一个运行示例。

Enter the array：

23.5 35 2 10

4.5 3 45 3.5

35 44 5.5 9.6

The location of the largest element is 45 at (1,2).

* 面向对象思考

1. （时间类Time，\*）设计一个名为Time的类。这个类包括：
   1. 表示时间的数据域hour、minute和second。
   2. 一个以当前时间创建Time对象的无参构造函数。其中，数据域的值表示当前时间。
   3. 一个构造Time对象的构造函数。这个对象有一个特定的时间值，这个值是以毫秒表示的、从1970年1月1日午夜零点开始到现在流逝的时间段（数据域的值表示这个时间）。
   4. 一个构造带特定的小时、分钟和秒的Time对象的构造函数。
   5. 三个数据域hour、minute和second各自的get方法。
   6. 一个名为setTime(long elapseTime)的方法，使用流逝的时间给对象设置一个新时间。例如，如果流逝的时间为555550000毫秒，那么转换为10小时、10分钟、10秒。

编写一个测试程序，分别使用new Time()和new Time(555550000)，创建两个Time对象。然后，显示它们的小时、分钟和秒。

1. （BMI类）设计一个名为BMI的类。其中，必须包括如下公共构造函数：BMI(String name, int age, double weight, double feet, double inches);。
2. （MyInteger类）设计一个名为MyInteger的类。这个类包括：
   1. 一个名为 value的int型数据域，存储这个对象表示的int值。
   2. 一个为指定的int值创建MyInteger对象的构造函数。
   3. 一个返回int值的get方法
   4. 如果（数据域的）值分别是偶数、奇数或素数，那么isEven()、isOdd()和isPrime()方法都会返回true。
   5. 如果指定值分别是偶数、奇数或素数，那么相应的静态方法isEven(int)、isOdd(int)和isPrime(int)都会返回true。
   6. 如果指定值分别是偶数、奇数或素数，那么相应的静态方法isEven(MyInteger)、isOdd(MyInteger)和isPrime(MyInteger)都会返回true。
   7. 如果该对象的值与指定的值相等，那么equals(int)和equals(MyInteger)方法都返回true。
   8. 静态方法parseInt(char[])将由数字字符构成的数组转换为一个int值。
   9. 静态方法parseInt(String)将一个字符串转换为一个int值。

编写测试程序，测试这个类中的所有方法。

1. （MyPoint类）设计一个名为MyPoint的类，表示一个带x坐标和y坐标的点。这个类包括：
   1. 两个带get方法的数据域x和y，分别表示它的坐标。
   2. 一个创建点(0,0)的无参构造函数。
   3. 一个创建特定坐标点的构造函数。
   4. 一个名为distance的方法，返回从该点到MyPoint类型的指定点之间的距离。
   5. 一个名为distance对的方法，返回从该点到指定x和y坐标的指定点之间的距离。

编写一个测试程序，创建两个点（0,0）和(10,30.5)，显示它们之间的距离。

1. （ATM机，\*\*）使用练习题270中创建的Account类，模拟一台ATM机。首先，创建一个有10个账户的数组。其中，id为0，1，…，9，并初始化收支为100元。然后，系统提示用户输入一个id。 如果输入的id不正确，那么要求用户输入正确的id。如果一旦接受一个id，那么就显示如运行示例所示的主菜单。其中，选择1，表示查看当前的收支；选择2，表示取钱；选择3，表示存钱；选择4，表示退出主菜单。一旦退出，系统就会提示再次输入id。所以，系统一旦启动就不会停止。下面是一个运行示例。

Enter an id : 4

Main menu

1: Check balance

2: withdraw

3: deposit

4: exit

Enter a choice: 1

The balance is 100.0

Main menu

1: Check balance

2: withdraw

3: deposit

4: exit

Enter a choice: 2

Enter an amount to withdraw: 3

Main menu

1: Check balance

2: withdraw

3: deposit

4: exit

Enter a choice: 1

The balance is 97.0

Main menu

1: Check balance

2: withdraw

3: deposit

4: exit

Enter a choice: 3

Enter an amount to deposit: 10

Main menu

1: Check balance

2: withdraw

3: deposit

4: exit

Enter a choice: 1

Enter an amount to deposit: 107.0

Main menu

1: Check balance

2: withdraw

3: deposit

4: exit

Enter a choice: 4

Enter an id:

1. （课程类Course，\*\*）设计一个类Course，来对课程建模。假设需要处理课程信息。每门课程都有一个名字以及选课的学生，需要能够向/从这门课程中添加/删除一个学生。这个类包括：
   1. 一个存储该课程学生的数组。
   2. 学生的个数，默认值为0，最大人数100人。
   3. 创建一个带特定名称的课程的构造函数
   4. 返回课程名
   5. 一个名为addStudent(String student)的方法，向这门课程添加一位新学生。
   6. 一个名为dropStudent(String student)的方法，从这门课程中删除一位学生。
   7. 一个名为getStudents()的方法，返回修读这门课程的所有学生（的姓名列表）。
   8. 一个名为getNumberOfStudents()的方法，返回修读这门课程的学生人数。
   9. 一个名为clear()的方法，删除修读这门课程的所有学生。

编写一个测试程序，创建一门课程，添加三个学生，删除一个学生，然后显示这门课程的学生。附加问题：现在，（修读学生的）数组的大小是固定的。尝试对其进行改进，通过创建一个新的更大的数组并复制当前数组的内容，实现数组大小的自动增长。

1. （Queue类，\*）设计一个名为Queue（队列）的类，用于存储整数。和数组一样，队列也由元素组成。在队列中，元素以“先进先出”的方式获取。这个类包括：
   1. 一个名为element的int[]类型的数据域，保存队列中的int值。
   2. 一个名为size的数据域，保存队列中的元素个数。
   3. 一个构造函数，使用默认的容量8来创建一个Queue对象。
   4. 方法enqueue(int v)，将v加入到队列的尾部中。
   5. 方法dequeue()，用于从队列中移除（首）元素并返回该元素。
   6. 方法empty()，如果队列是空的，那么该方法返回true。
   7. 方法getSize()，返回队列的大小。

编写一个测试程序，增加从1到20的21个成员，然后将这些数字移除并显示它们。注，初始数组的大小为8。如果队列中元素的个数超过了当前的大小，那么数组大小将会翻倍。如果一个元素从数组的开始部分进行移除，那么我们需要把数组中的所有剩余元素往前（往左）移动一个位置。

1. （Circle2D类，\*）定义Circle2D类。这个类包括
   1. 两个带有get方法的名为x和y的double型数据域，表示圆心。
   2. 一个带get方法的数据域radius，表示圆的半径。
   3. 一个无参构造函数，创建一个圆形为(0,0)、radius为1的单位圆。
   4. 一个构造函数，创建带指定的x、y和radius的圆。
   5. 一个返回圆面积的方法getArea()。
   6. 一个返回圆周长的方法getPerimeter()。
   7. 如果给定的点(x,y)在圆内，那么方法contains(double x,double y)返回true。
   8. 如果给定的圆在这个圆内容，那么方法contains(Circle2D one)返回true。
   9. 如果给定的圆和这个圆有重叠部分，那么方法overlaps(Circle2D one)返回true。

编写一个测试程序，使用new Circle2D(2,2,5.5)，创建一个Circle2D对象c1，显示它的面积和周长，还要显示c1.contains(3,3)、c1.contains(new Circle2D(4,5,10.5))和c1.overlaps(new Circle2D(3,5,2.3))的（判断）结果。

1. （Triangle2D类，\*\*\*）定义Triangle2D类。它包括：
   1. 三个名为p1、p2和p3的MyPoint类型（练习278）数据域，这个三个数据域都带有get和set方法。
   2. 一个无参构造函数，该方法创建三个坐标为(0,0)、(1,1)和(2,5)的点组成的默认三角形。
   3. 一个创建带指定点的三角形的构造函数。
   4. 一个返回三角形面积的方法getArea()。
   5. 一个返回三角形周长的方法getPerimeter()。
   6. 如果给定的点p在这个三角形内，那么方法contains(MyPoint p)返回true。
   7. 如果给定的三角形在这个三角形内，那么方法contains(Triangle2D one)返回true。
   8. 如果给定的三角形和这个三角形重叠（存在交叠部分），那么方法overlaps(Triangle2D one)返回true。

编写测试程序，使用构造方法new Triangle2D(new MyPoint(2.5,2), new MyPoint(4.2,3), new MyPoint(5,3.5))创建一个Triangle2D对象t1，显示它的面积和周长，并显示t1.contains(3,3)、t1.contains(new Triangle2D(new MyPoint(2.9,2), new MyPoint(4,1), new MyPoint(1,3.4)))和t1.overlaps(new Triangle2D(new MyPoint(2,5.5), new MyPoint(4,-3), new MyPoint(2,6.5)))的结果。

提示：三角形面积的计算公式见练习30。为了检测一个点是否在三角形中，可以分别连接该点和三个顶点，得到三条连线。如果点在三角形中，那么每条连线应该和边相交一次。如果连线和边相交两次，那么这个点肯定在这个三角形外（这个条件有点不理解。如果点在外，那么交点就是三角形的顶点。这是我的理解）。找到两条线交点的算法见练习57。

1. （MyRectangle2D类，\*）定义MyRectangle2D类。这个类包括：
   1. 两个名为x和y的double型数据域，表示矩形的中心点。这两个数据域都带有get和set方法。假设这个矩形的边与x轴和y轴平行。
   2. 带get和set方法的数据域width和height。
   3. 一个无参构造函数，创建一个中心点为(0,0)且width和height为1的矩形。
   4. 一个构造函数，创建带指定的x、y、width和height的矩形。
   5. 方法getArea()，返回矩形的面积。
   6. 方法getPerimeter()，返回矩形的周长。
   7. 如果给定的点(x,y)在矩形内，那么方法contains(double x, double y)返回true。
   8. 如果给定的矩形在这个矩形内，那么方法contains(MyRectangle2D one)返回true。
   9. 如果给定的矩形和这个矩形重叠（存在交叠部分），那么overlaps(MyRectangle2D one)返回true。

编写测试程序，使用new MyRectangle2D(2,2,5.5,4.9)，创建一个MyRectangle2D对象a，显示它的面积和周长、a.contains(3,3)、a.contains(new MyRectangle2D(4,5,10.5,3.2))和a.overlaps(new MyRectangle2D(3,5,2.3,5.4))的结果。

1. （MyDate类，\*）设计一个名为MyDate的类。这个类包括：
   1. 表示日期的数据域year，month和day。月份从0开始，0表示一月份。
   2. 一个无参构造方法，创建当前日期的MyDate对象。
   3. 一个构造方法，创建以从1970年1月1日午夜零点开始流逝的毫秒数为时间的MyDate对象。
   4. 一个构造函数，创建一个带指定年、月、日的MyDate对象。
   5. 三个数据域year、month和day的get方法。
   6. 一个名为setDate(long elapsedTime)的方法，使用流逝的时间（毫秒数）设置新的日期。

编写测试程序，分别使用new Date()和new Date(34355555133101L),创建两个Date对象，然后显示它们的小时、分钟和秒。提示，前两个构造函数从流逝时间的毫秒数中提取出年、月、日。例如，如果流逝的时间是561555550000毫秒，那么年就是1987，月就是9，天是18。

1. （边界矩形，\*）边界矩形是指包围一个二维平面上一系列点的矩形，如下图所示。



图1. 包围在矩形中的点（边界矩形）

编写一个方法，为二维平面上的一系列点返回一个边界矩形。方法声明如下所示：

MyRectangle2D getRectangle(double points[][2],int n)。

编写一个测试程序，提示用户输入5个点，然后显示边界矩形的中心、宽度以及高度。下面是一个运行示例。

Enter five points: 1.0 2.5 3 4 5 6 7 8 9 10

The bounding rectangle’s center (5.0, 6.25), width 8.0, height 7.5

* 继承与多态

1. （Person、Student、Employee、Faculty和Staff类）设计一个名为Person的类和它的两个名为Student和Employee的子类。Employee类又有两个子类：教员类Faculty和职员类Staff。每个人都有姓名、地址、电话号码和电子邮件地址。学生有班级状态（大一、大二、大三或大四），将这些状态定义为常量。一个雇员涉及办公室、工资和受聘日期。使用练习285题定义的MyDate类，为受聘日期创建一个对象。教员有办公时间和级别。职员有职务称号。覆盖每个类中的toString方法，显示相应的类别名字和人名。

编写一个测试程序，创建Person、Student、Employee、Faculty和Staff对象，并且调用它们的toString()方法。

1. （账户类Account的子类）在练习题270中定义的Account类模拟一个银行账户。一个账户有账号、余额、年利率、开户日期等属性，以及存款和取款等方法。创建Account类的两个子类：支票账户（checking account）和储蓄账户（saving account）。支票账户有一个透支限定额；储蓄账户不能透支。

编写一个测试程序，创建Account、SavingAccount和CheckingAccount的对象并测试它们。

1. （课程类Course）重写Course类，使用ArrayList（或者，Vector）代替数组来存储学生。注，不应该改变Course类的原始合约（即，构造函数和公共方法的声明都不应该发生变化，但私有成员可以改变。）。
2. （新的Account类，\*\*）根据练习题270中定义的Account类，设计一个新的Account类。这个类包括：
   1. 添加一个String的新数据域name，存储客户的名字。
   2. 添加一个新的构造方法，创建一个具有指定名字、id和收支额的账户。
   3. 添加一个名为transactions的ArrayList（或者，Vector）类型的新数据域，用于为账户存储交易。每笔交易都是一个Transaction类的对象。
   4. 修改withdraw和deposit方法，向transactions数据域线性添加一笔交易（添加在transactions的尾部）。
   5. 其他所有属性和方法同练习270。

Transaction类的定义：

1. 交易的日期date。
2. 交易的类型type。例如，“W”代表取款，“D”代表存款。
3. 交易量amount。
4. 交易后的新余额balance。
5. 交易描述description
6. 构造方法，使用给定日期、类型、余额以及描述，创建一个Transaction对象。

编写一个测试程序，创建一个年利率为1.5%、存款额为1000、id为1122、名字为George的Account对象。依次向该账户存入30元、40元和50元。然后，依次从该账户中取出5元、4元和2元。打印账户清单，显示账户持有者的名字、利率、存款额和所有的交易记录。

1. （凸多边形面积，\*）如果连接一个多边形中任意两个顶点的线段都包含在这个多边形中，那么称它为凸多边形。编写一个程序，提示用户输入一个凸多边形中的顶点数，并顺时针输入点，然后显示多边形的面积信息。下面是一个运行示例。

Enter the number of the points: 7

Enter the coordinates of the points:

-12 0 -8.5 10 0 11.4 5.5 7.8 6 -5.5 0 -7 -3.5 -13.5

The total area is 292.575

1. （加分测试，\*\*）如果用户重复输入相同的答案，那么给出用户警告。下面是一个运行示例。注，使用一个数组列表来存储输入的答案。

What is 6 + 9? 12

Wrong answer. Try again. What is 6 + 9? 34

Wrong answer. Try again. What is 6 + 9? 12

You already entered 12

Wrong answer. Try again. What is 6 + 9? 15

You got it！

1. （完全平方，\*\*）编写一个程序，提示用户输入一个整数m，然后找到最小的整数n，使得 m \* n 是一个完全平方（若一个数能表示成某个整数的平方的形式，则称这个数为完全平方数。）。下面是一些运行示例。提示，存储所有m的最小因子到一个数组列表，n是列表中出现奇数次的因子的乘积。例如，考虑 m = 90的情况，保存因子2，3，3，5到一个数组列表中。在列表中，2和5出现了奇数次数。因此，n是10。

Enter an integer m: 1500

The smallest number n for m \* n to be a perfect square is 15

m \* n is 22500

Enter an integer m: 63

The smallest number n for m \* n to be a perfect square is 7

m \* n is 441