

# 历年NOC操作题精选

## 2020年（赛制没设置Nemo组别）

### 小学Kitten

#### 操作题1

六一儿童节，小可一家人去商场游玩，爸爸的车停在停车场。已知停车场的收费标准是：停车1小时，收费5元；如果超过了1个小时，则超出部分按6元/小时收费；如果超过了5个小时，则超出部分按7元/小时收费。（若不足1小时按照1小时计算）

请设计一个程序：输入停车时间，自动计算停车费。

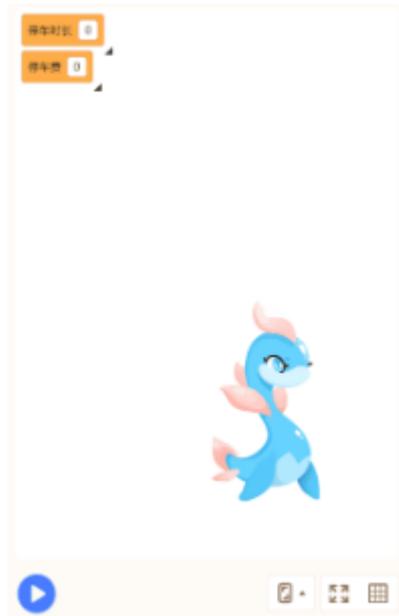
程序要求：

- 1.运行程序，弹出询问框，询问：“停车几个小时？”；
- 2.输入停车时长（输入的数字为正数）；
- 3.使用“对话（）”积木，显示需要缴纳的停车费用。

示例：

请输入停车时长	3	3.5	4	6
停车费用	17	23	23	36

程序效果：



## 预置文件

[https://education.codemao.cn/27/public-edu/1/2020NOC/停车费计算预置\(6\).bcm](https://education.codemao.cn/27/public-edu/1/2020NOC/停车费计算预置(6).bcm)

## 完整bcm代码文件

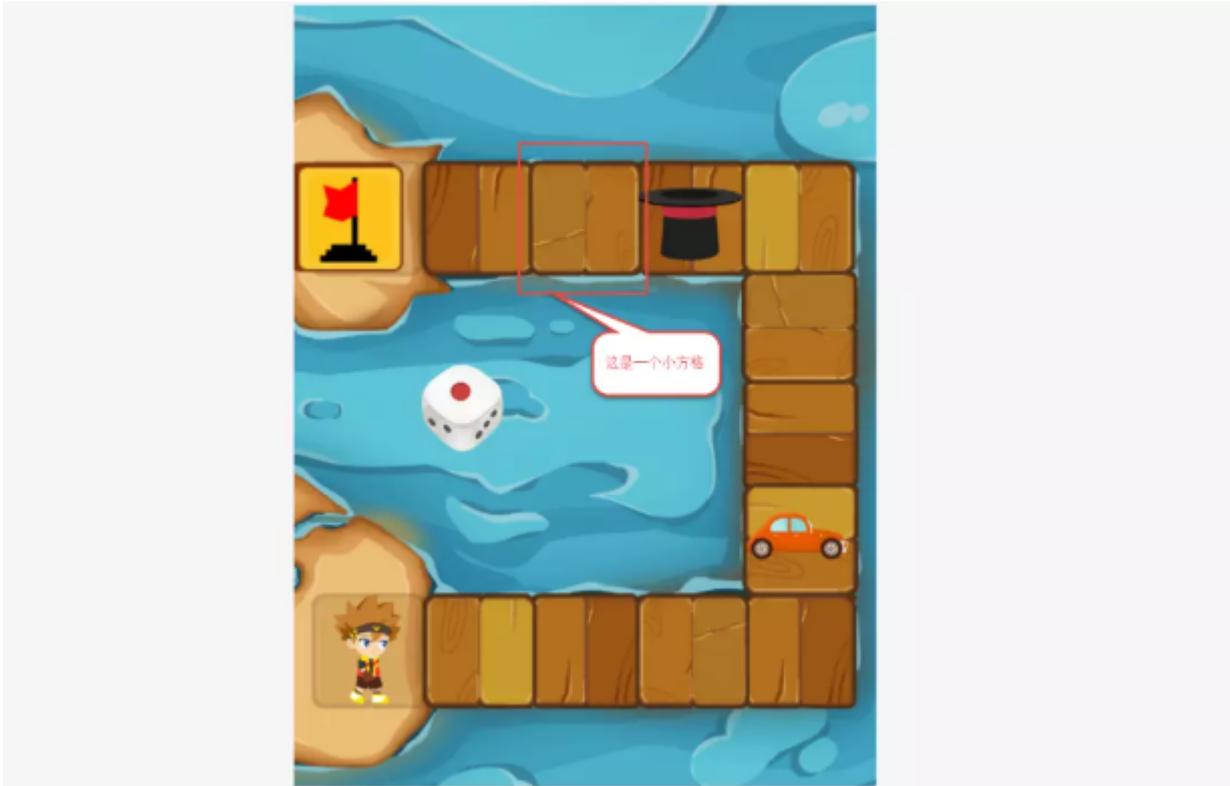
<https://education.codemao.cn/27/public-edu/1/2020NOC/计算停车费完整BCM.bcm>

测试范例：

组别	第一组	第二组	第三组	第四组	第五组	第六组	第七组
输入	0.5	1	2.5	3	5	5.5	6
输出	5	5	17	17	29	36	36

## 操作题2

阿短要去小岛（红旗所在位置）游玩。他需要经过一座由11个小方格组成的桥（方格的边长为115）。来帮帮他吧！



程序要求：

(1) 程序开始，阿短在图示的起点位置点，红旗在图示的终点位置。界面上有一个骰子（位置、大小自由调整）；

桥的格子上随机出现魔术帽和车子两个角色，但二者不能重叠；

(2) 点击骰子，进行骰子投掷，骰子有点数的滚动变换效果，然后随机产生一个1到6之间的点数；

(3) 阿短沿着桥梁行走，如投掷点数为2，则前进两格。能重复进行投掷，直到程序结束；前进过程中，阿短有走路的动态效果，且面向前进的方向；

(4) 阿短前进完后，如果停留在车子所在格子上，显示“胜利”，程序结束；

如果停留在魔术帽所在格子上，显示“失败”，程序结束；

(5) 当阿短碰到红旗，显示“胜利”，程序结束。

预置文件

[https://education.codemao.cn/27/public-edu/1/2020NOC/操作题2预置\(1\).bcm](https://education.codemao.cn/27/public-edu/1/2020NOC/操作题2预置(1).bcm)

完整bcm代码文件

<https://education.codemao.cn/27/public-edu/1/2020NOC/操作题2--完整.bcm>

### 操作题3

闰年怎么算？判定公历闰年遵循的规律为：四年一闰，百年不闰，四百年再闰。

公历闰年的简单计算方法是：

- 1.能被4整除而不能被100整除（如2004年就是闰年,1900年不是闰年）；
- 2.能被400整除（如2000年是闰年,1998年不是闰年）。

（注意：只要符合以上条件之一的年份即为闰年）

阿短想设计一个计算闰年的小程序。输入一个随机四位整数，程序就会对这整数进行判断是否为闰年，最后将结果以新建对话框的形式输出。

#### 【程序要求】

- 1、程序运行，询问输入测试的四位整数；
- 2、根据以上两种闰年判断规则，判断用户输入的整数是否为闰年；
- 3、以新建对话框的形式输出判断结果，如果符合判断规则，输出“闰年”，不符合则输出“平年”。

### 完整bcm代码文件

<https://education.codemao.cn/27/public-edu/1/4894bc66d56e4d83bc84e37aca7cf3130668.bcm>

### 操作题4

宇宙中有一种叫“黑洞”的天体，由高密度物质组成，引力极其强大，连光线都无法逃脱。数学中也有一种神奇的“数字黑洞”——123，也叫作西西弗斯数。

任意选取一个自然数，如“1234567890”，该自然数中包含5个偶数、5个奇数、该数是10位数，将统计出的这3个数按照“偶-奇-总”的顺序排列得到一个新数：5510，接着将新数5510按以上规则重复进行，可得到新数:134，又将新数134按以上规则重复进行，最终得到数字123。继续重复以上规则的，得到的数字始终是123，像物质进入黑洞一样再也逃不出去了，因此123叫做数字黑洞。

请用编程验证西西弗斯黑洞（123数字黑洞），其规则如下：

- 1、点击开始后，输入任意一个自然数，求出这个自然数中所包含的偶数的个数、奇数的个数和这个数的位数；
- 2、将求出的这3个数按照“偶-奇-总”的顺序排列得到一个新数，并以新建对话框的形式输出得到的新数；
- 3、对得到的新数重复前面的做法，直到得到的最终数为123，并将最终数以对话的形式显示在舞台上。

## 完整bcm代码文件

<https://education.codemao.cn/27/public-edu/1/c04e4b0bd5c94b11ab32e2be0b0515d80668.bcm>

## 中学Kitten

### 操作题1

编写一个程序，输入整数n，判断n的各数位上的数字是奇数还是偶数（0是偶数），并输出奇数的个数和偶数的个数。

示例1：

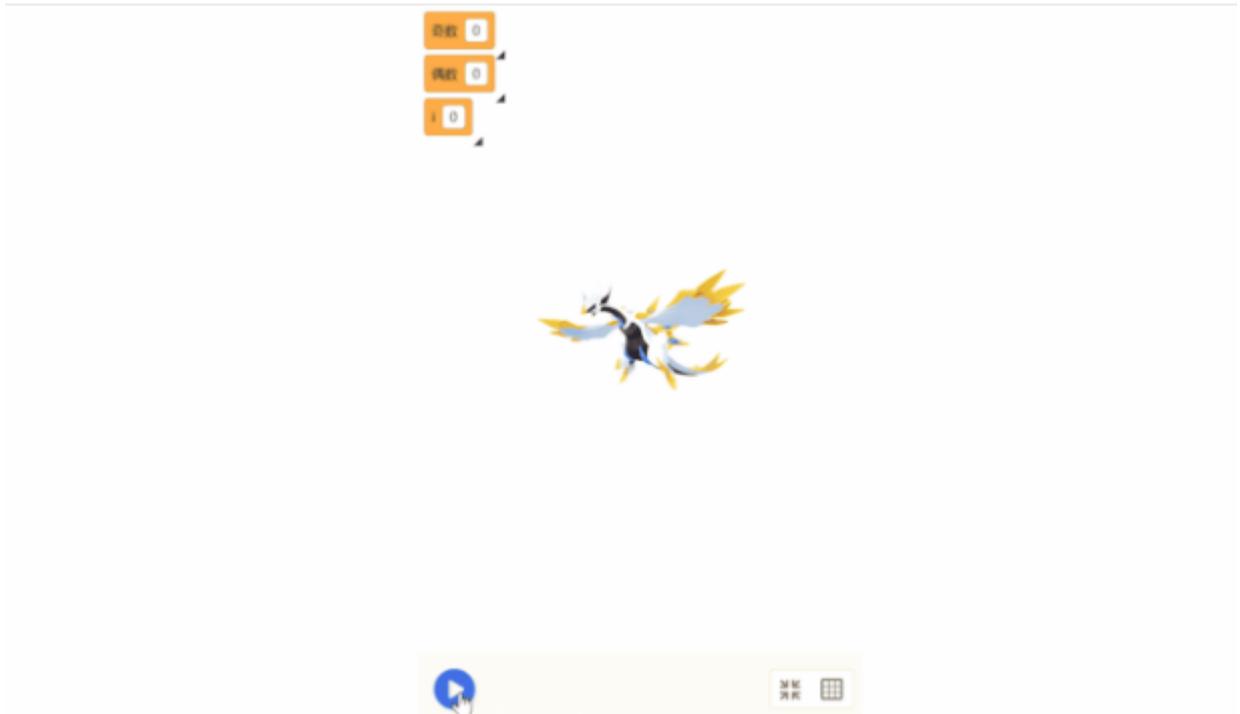
输入：18

输出：奇数：1，偶数：1

示例2

输入：1234567

输出：奇数：4，偶数：3



预置文件

[https://education.codemao.cn/27/public-edu/1/2020NOC/奇数与偶数的个数判断预置\(1\).bcm](https://education.codemao.cn/27/public-edu/1/2020NOC/奇数与偶数的个数判断预置(1).bcm)

完整bcm代码文件

<https://education.codemao.cn/27/public-edu/1/2020NOC/奇数与偶数的个数判断.bcm>

测试范例：

组别	第一组	第二组	第三组
输入	25	243	1024
输出	奇数： 1， 偶数： 1	奇数： 1， 偶数： 2	奇数： 1， 偶数： 3

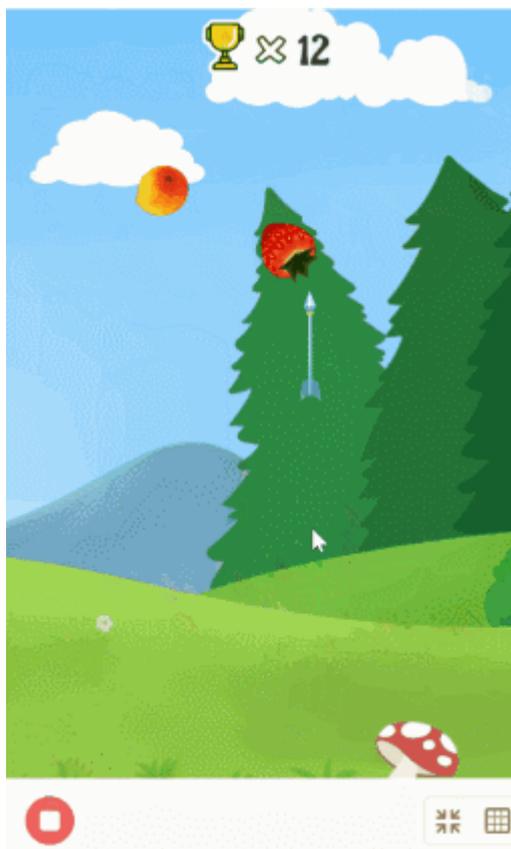
操作题2

《士兵突击》是一部热血激荡的军旅片，片中很多人物都有一手好枪法，其中有一段剧情是一位神枪手在草原上射击空中的移动靶，百发百中。阿短受此启发，想设计一个射箭的游戏。

程序要求：

- 1.按要求导入对应的素材（桃子、草莓、西瓜、箭）和背景；
- 2.每次按下鼠标就有一支箭出现在鼠标处，并能随着鼠标水平移动，松开鼠标后，箭射出。
- 3.西瓜、草莓和桃子随机出现在舞台区左边缘，并从左至右移动。当它们移出舞台区后，再次出现在舞台区左边缘的随机位置。
- 4.被箭射中的水果，随箭一同消失，且变量“得分”加1。

程序效果：



预置文件

[https://education.codemao.cn/27/public-edu/1/2020NOC/射箭游戏预置4\(1\).bcm](https://education.codemao.cn/27/public-edu/1/2020NOC/射箭游戏预置4(1).bcm)

完整bcm代码文件

## 中学Python

### 操作题1

请用海龟编辑器设计程序，将用户输入的字符串中出现的数字以列表的形式进行输出，并能够统计出各汉字、字母、空格等其他符号（字母不区分大小写）出现的次数，统计完成后以字典的形式进行输出。

输入输出示范：

1.请输入字符串：1231['1', '2', '3', '1']

{}

2.请输入字符串：你A#saS

[]

{'你':1, 'a':2, '#':1, 's':2}

3.请输入字符串：121你你Aasasa

['1', '2', '1']

{'你':2, 'a':4, 's':2}

### 【参考代码】

```
1 s = input("请输入字符串")
2 s = s.lower()
3 num = []
4 dic = {}
5 for i in s:
6     if i.isdigit():
7         num.append(i)
8     else:
9         if i in dic:
10            continue
11        else:
12            dic[i] = s.count(i)
13 print(num)
14 print(dic)
```

## 操作题2

给定一个正整数  $n$  ( $0 < n < 1\,000\,000\,000$ ), 每个数位上数字均不为 0。正整数  $n$  的位数为  $m$ , 现在从  $m$  位中删除  $k$  位 ( $0 < k < m$ ), 求组成的新整数最小为多少?

例如:  $n = 9128456$ ,  $k=2$ , 则生成的新整数最小为 12456;  $n = 92415$ ,  $k=2$ , 则生成的新整数最小为 215;  $n = 2318$ ,  $k=1$ , 则生成的新整数最小为 218;  $n = 1333$ ,  $k=3$ , 则生成的新整数最小为 1。

### 【程序要求】

- 1、控制台分别输入正整数  $n$  和删除位数  $k$  且有输入提示信息;
- 2、控制台输出最小新整数;
- 3、运行时间不能超过 3 秒。

### 【参考程序】

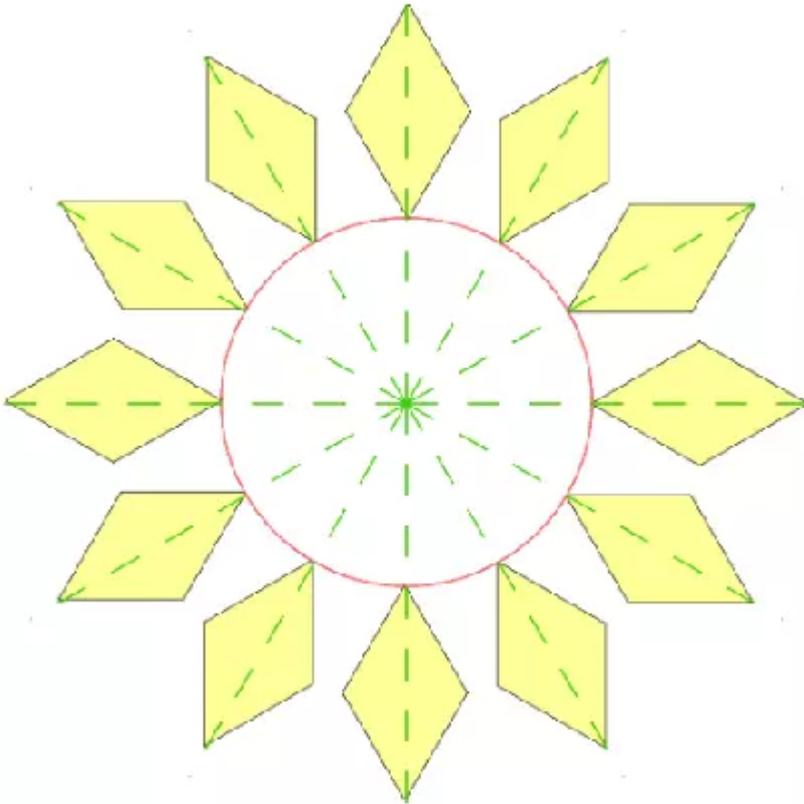
```
1 n = input('请输入数字：')
2 k = int(input('请输入删除的位数：'))
3 n += '0'
4 while k > 0:
5     i = 0
6     l = len(n)
7     while(i < l-1):
8         if(n[i] > n[i+1]):
9             n = n[:i] + n[i+1:]
10            break
11        i += 1
12    k -= 1
13 n = int(n[:-1])
14 print(n)
```

## 2021年

### 小学Nemo

#### 操作题1

轴对称图形，是指在平面内，沿着一条直线折叠之后直线两旁的部分能够完全重合的图形，这条直线也叫作对称轴。有些特殊的轴对称图形，其对称轴不止1条，例如下图所示的图形。请你仔细观察图形的特点，然后编写程序将这个图形绘制出来。



### 【程序要求】

- 1、中间圆的半径为120，轮廓线为红色，不填充；
- 2、周围菱形为黑色轮廓线，边长为80，菱形锐角为60度，黄色填充；
- 3、圆周围总共有12个形状形同的菱形组成，菱形长对角线延长线经过圆心(如图中虚线所示，注：虚线不用绘制)；
- 4、绘图过程中隐藏画笔，能清楚地看到图形绘制过程。

### 操作题2

几何图形看似简单，却能延伸出最复杂多变的逻辑。在自然界，有六边形的蜂巢和雪花，有黄金比例的鹦鹉螺。在人类文明伊始，便有线形的长城，三角的胡夫金字塔，和长方状的帕特农神庙。几何图形有一种难以言表的美感，圆形的圆润，方形的刚硬，三角形的锐利都是图形给我们的感觉，而正多边形是几何图形中一类特殊的存在。正多边形是指二维平面内各边相等，各角也相等的多边形，也叫正多角形。请用编程制作一个可以画出指定边数的正多边形的程序（推荐边长为100）。

参考公式：

n代表边数

正n边形的内角和度数为： $(n-2)\times 180^\circ$

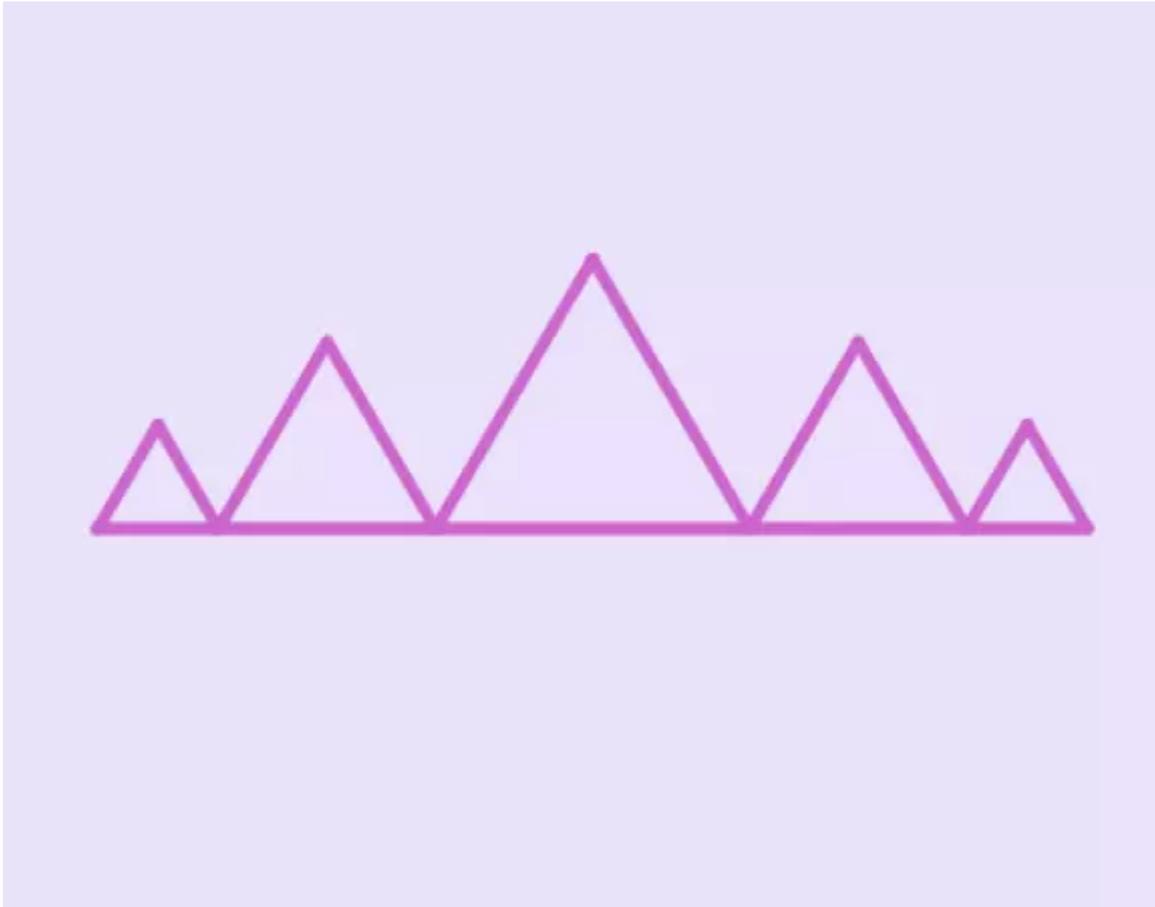
正n边形的一个外角为： $360^\circ \div n$

### 【程序要求】

- 1、创建一个“绘制一个正多边形”的有参函数，参数为“边数”；
- 2、程序运行时，先接收用户输入的边数，如果边数是小数，那么程序提示用户“边数不得是小数”，如果边数小于3，那么程序提示用户“边数不得小于3”。这两种情况都没出现时，程序才调用“绘制一个正多边形”函数，同时将用户输入的边数作为函数的参数；
- 3、对“绘制一个正多边形”函数体编程，结合参考公式设计程序，实现绘制指定边数的正多边形。

### 操作题3

鲁班是我国古代的一位创造发明家，他偶然被一种叶子边缘呈波浪状轮廓的植物划伤了自己的手，由此受到启发，最终发明出了锯子。这种由多个三角形组成的波浪形状也被称为锯齿，请你用NEMO画出一条特别的锯齿，如下图所示。这个锯齿是由5个正三角形（每条边长都相等的三角形）组成的图形，整个图形左右对称。



### 【程序要求】

- 1、设置画笔角色初始位置为  $(-550,0)$  ，画笔粗细为5，画笔颜色任意；
- 2、能够在舞台绘制一个边长为50的正三角形，且舞台上不能有其他图形或者角色；
- 3、能够在要求2的基础上，继续向右连续绘制2个正三角形，其边长依次增加40；
- 4、能够在要求3的基础上，继续向右连续绘制2个正三角形，其边长依次减小40。

### 操作题4

数字排序是程序中经常出现的操作，请你按照下面的程序要求，完成排序程序。

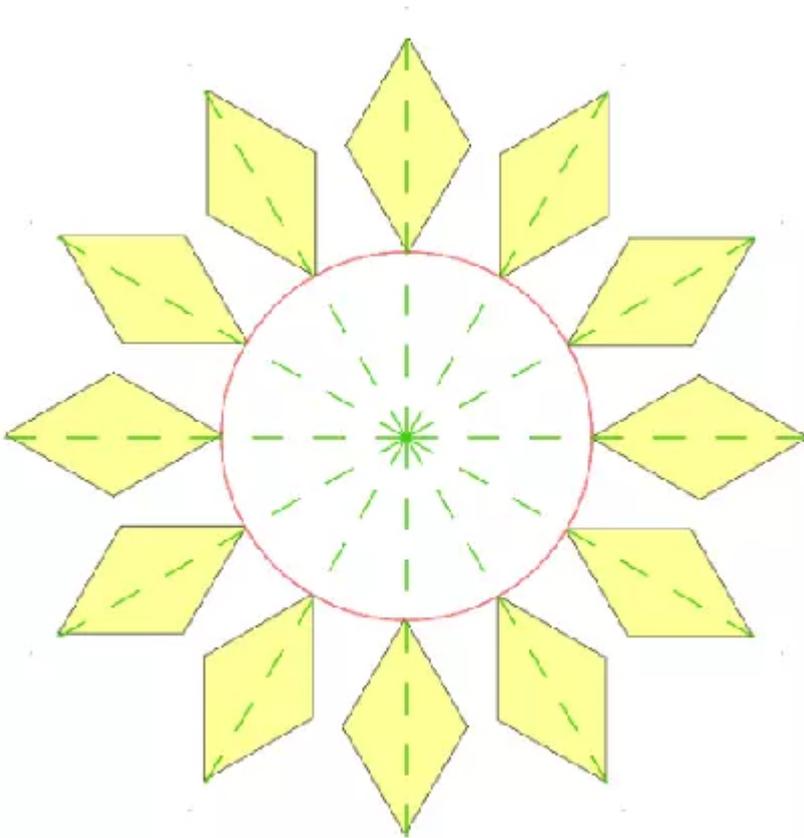
### 【程序要求】

- 1、在1~9范围内随机生成5个整数（不可重复），保存到列表1，之后等待1秒；
- 2、每间隔1秒，就把列表1中当前最大的一个数字移动到列表2中，注意：是移动（移动之后列表1中就没有该数字了），不是复制；
- 3、重复进行要求2的操作，直到列表1为空，列表2中的5个数字按照从大到小排列。

## 小学Kitten

### 操作题1

轴对称图形，是指在平面内，沿着一条直线折叠之后直线两旁的部分能够完全重合的图形，这条直线也叫作对称轴。有些特殊的轴对称图形，其对称轴不止1条，例如下图所示的图形。请你仔细观察图形的特点，然后编写程序将这个图形绘制出来。



请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

### 【程序要求】

- 1、中间圆的半径为120，轮廓线为红色，不填充；
- 2、周围菱形为黑色轮廓线，边长为80，菱形锐角为60度，黄色填充；
- 3、圆周围总共有12个形状形同的菱形组成，菱形长对角线延长线经过圆心(如图中虚线所示，注：虚线不用绘制)；
- 4、绘图过程中隐藏画笔，能清楚地看到图形绘制过程。

### 【预置】



画太阳-预置程序.bcm

附件大小：32.2KB

## 【参考】



画太阳.bcm

附件大小: 39.3KB

### 操作题2

阿短是一位小学5年级学生，本学期即将结束，阿短打算创作一个能够进行四则运算练习的出题程序。程序能够随机出10道题，题目皆是一位数的加减乘除四则运算，每答对一题得1分。请你帮助阿短，编写出这个程序吧。



确定

特别说明：

- 1、在计算机中，“\*”表示乘法运算，等同于数学中“ $\times$ ”；“/”表示除法运算，等同于数学中“ $\div$ ”；
- 2、对于除法运算，要求除数不得为0，对于结果为小数的情况，进行四舍五入取整；
- 3、小学数学大纲中不涉及负数运算，所以在进行减法运算时，要求被减数不小于减数。

请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

### 【程序要求】

- 1、参与运算的两个数值都是从0到9之间随机获取，运算符是从“+”、“-”、“\*”、“/”中随机产生，并作为询问的内容，如题干配图所示；
- 2、在要求1的基础上进一步对数值做出限制，如果运算符为“-”，那么要求被减数不小于减数；如果运算符为“/”，那么除数不得为0；
- 3、程序对随机生成的算式进行运算，并对结果进行四舍五入取整。如果用户输入答案与结果相等，那么说出“回答正确”，然后将得分增加1分；否则说出“回答错误”；
- 4、程序总共随机生成10道题目。

### 【预置】



四则运算练习-预置程序.bcm  
附件大小：22.8KB

### 【参考】



四则运算-参考.bcm

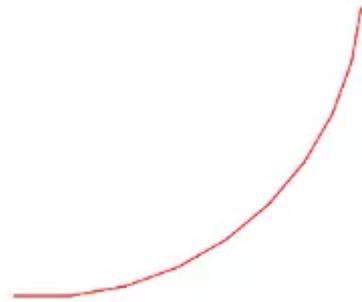
附件大小: 84.6KB

### 操作题3

莲花是一种洁白端庄的花朵，“出淤泥而不染，濯清涟而不妖”形容的就是莲花的品质。请你编写程序，在舞台上用画笔画出一朵美丽的莲花，如下图所示。



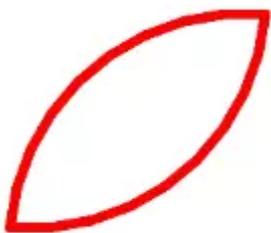
注：绘制一半花瓣线条的程序已经在预置程序中给出，其绘制的效果如下：



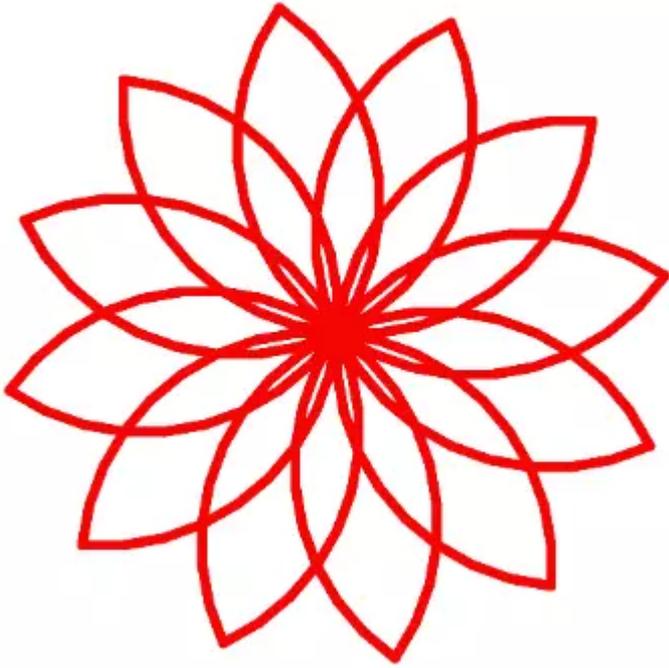
请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

### 【程序要求】

1、设置画笔粗细为7，初始颜色为红色，以坐标 (0,0) 为中心点绘制大小为40的花瓣；



2、在要求1基础上，以坐标 (0,0) 为中心点绘制一个如图所示的莲花（由12个花瓣组成）；



3、在要求2基础上，绘制如图所示的两层结构的莲花，第二层边长减少5，粗细减少1，颜色增加30；



4、在要求3基础上，绘制一个四层结构的莲花，与题干给出图形结构相同。

**【预置】**



画莲花-预置程序.bcm

附件大小: 23.4KB

## 【参考】



画莲花.bcm

附件大小: 33.6KB

### 操作题4

“数字黑洞153”又叫做“圣经数”，这个奇妙的数字黑洞是一个叫科恩的以色列人发现的。数感极好的科恩无意中发现153是3的倍数，并且它的各位数字的立方和仍然是153。无比兴奋之余，他又用另外一些3的倍数来做同样的运算，最后得数也都是153。于是科恩就把他发现的这个数字称之为“圣经数”。圣经数（数字黑洞153）的规则如下：任意取一个是3的倍数的自然数，求出这个数各个数位上数字的立方和，得到一个新数；然后再求出这个新数各个数位上数字的立方和，又得到一个新数。重复运算下去，最后一定会掉入数字黑洞153之中。

例如，69是3的倍数，按照数字黑洞153的规则，它的变换过程如下：

$$6^3+9^3=945,$$

$$9^3+4^3+5^3=918,$$

$$9^3+1^3+8^3=1242,$$

$$1^3+2^3+4^3+2^3=81,$$

$$8^3+1^3=513,$$

$$5^3+1^3+3^3=153$$

.....

注：“^”表示次方，例如 $6^3$ 表示6的3次方，即 $6 \times 6 \times 6$ 。

请用编程验证圣经数（数字黑洞153），其过程如下：

请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

### 【程序要求】

- 1、创建一个“数字黑洞153”有参函数，参数为“自然数”；
- 2、程序开始时，先询问用户想验证的自然数（必须是3的倍数）是多少。如果用户输入的自然数是3的倍数，就调用“数字黑洞153”函数，同时将该数字作为函数的参数，否则就提示用户重新输入一个3的倍数；
- 3、对“数字黑洞153”函数体编程，按照数字黑洞153的规则对输入的数字进行运算，并将计算的过程保存到列表中，直到达到153为止。

例：当用户输入69时，则程序结束时列表中应该依次存在数字：69，945，918，1242，81，513，153。

### 【预置】



圣经数预置程序.bcm  
附件大小：22.0KB

### 【参考】



圣经数.bcm  
附件大小：35.6KB

## 操作题1

排行榜在游戏中非常常见，通过排行榜，我们知道当前谁是高水平的玩家。请你用kitten编程，制作一个能够自动更新排名的排行榜，要求该排行榜统计排名前5玩家的得分和昵称。

请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

### 【程序要求】

- 1、当点击开始时，程序将“得分榜”列表和“玩家”列表前5项设置为空；
- 2、当按下空格键时，程序分别询问用户的游戏得分和游戏昵称，并保存；
- 3、程序将玩家的得分与“得分榜”列表中元素依次进行比较，如果发现得分比“得分榜”列表某一项数值大，则将得分插入到列表这一项位置，同时将游戏昵称插入到“玩家”列表同样的项数位置，最后说出玩家的名次；
- 4、如果列表中5项元素全部比较结束仍然没有找到插入位置，那么说出“玩家没有进入排行榜”。

### 【预置】



成绩排行榜-预置程序.bcm  
附件大小：23.2KB

### 【参考】



成绩排行榜.bcm  
附件大小：35.1KB

## 操作题2

如下图（左）所示函数“圆”可以画出一个圆，函数圆的参数“步数”决定圆的大小。请利用“圆”函数，设计程序，画出下图（右）所示的图形（图形全部位于舞台范围内），该图形从外数第二层圆与第一层相内切于最上方，但是第三层圆与第二层圆相内切于最下方，以此类推，一共5层。请你编写程序，绘制出5层内切圆吧。



请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

### 【程序要求】

- 1、设定“笔”角色初始位置为 (0,0) ，画笔粗细为5，画笔初始颜色不做要求；
- 2、能够在舞台上画出一个步数为3的圆，作为最外层圆；
- 3、能够绘制第二层圆，其步数减小0.5，颜色值增加50，与外层圆在最上方相内切，
- 4、总共在舞台上绘制出5个圆，相邻圆之间步数相差0.5，画笔颜色值相差50，相切位置与题干要求一致。

### 【预置】



绘制内切圆-预置程序.bcm

附件大小：24.1KB

## 【参考】



绘制内切圆.bcm  
附件大小: 27.3KB

### 操作题3

在4年一度的奥林匹克运动会（以下简称“奥运会”）上，有这么一个优美的体育项目——跳水。跳水运动员会站在离水面几米高的跳台上，起跳并完成自己选定的动作，裁判们会根据运动员选定的动作的完成程度和入水时水花的大小（水花越小分数越高）来给运动员们的表现评分（最低0分，最高10分）。为了公平起见，在单人跳水中一共会有7个裁判对运动员进行评分，但是会去掉其中的一个最高分和一个最低分，再计算剩下的5个裁判评分的平均值，作为这个运动员本次跳水的分数。

示例：七个裁判的评分为：8.9， 7.1， 9.8， 9.0， 9.4， 9.8， 9.2。

去掉一个最高分9.8， 去掉一个最低分7.1， 这名运动员本次跳水最终得分为：

$$(8.9+9.8+9.0+9.4+9.2) / 5=9.26$$

请写一个程序，能够帮助裁判组算出选手最终得分。

### 【程序要求】

- 1、点击开始后，“编程猫”角色分别询问7个裁判的评分，“请输入第1个裁判的打分”，“请输入第2个裁判的打分”，……；并依次将分数保存到列表中；
- 2、找出所有评分中的最高分，然后让“编程猫”角色说“去掉一个最高分：X分”2秒（句中X为最高分）；

3、找出所有评分中的最低分，然后让“编程猫”角色说“去掉一个最低分，Y分”2秒（句中Y为最低分）；

4、计算剩余评分的平均分，然后让“编程猫”角色说“该选手最终得分：S分”2秒（句中S为剩余5个分数的平均分）。

### 【预置】



计算最终得分-预置程序.bcm

附件大小：22.0KB

### 【参考】



计算最终得分.bcm

附件大小：36.4KB

### 操作题4

尼科彻斯定理可以描述为“任何一个整数的立方都可以表示成一串连续的奇数的和”。根据尼科彻斯定理的叙述举例如下：

$$3*3*3=7+9+11=27$$

$$4*4*4=13+15+17+19=64$$

请你编写程序，验证尼科彻斯定理。

注：三个相同的数相乘,叫做这个数的立方。如 $5\times 5\times 5$ 叫做5的立方。

请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

### 【程序要求】

- 1、点击开始后，程序询问“请输入一个大于1的整数”，如果用户输入的数字小于等于1，或者输入的不是整数，则输出“输入的数值有误”并停止整个程序；
- 2、计算用户输入的数值的立方，并说出“X的立方是Y”2秒（X为用户输入的数值，Y为计算出的立方值）；
- 3、找到满足尼科彻斯定理的奇数数列，并将它们保存到列表“数值列表”列表中。

### 【预置】



尼科彻斯定理-预置程序.bcm

附件大小：22.3KB

### 【参考】



尼科彻斯定理.bcm

附件大小：39.6KB

## 中学Python

### 操作题1

完全数是一些特殊的自然数。它除了自身以外的约数的和，恰好等于它本身。例如，第一个完全数是6，它有约数1、2、3、6，除去它本身6外，其余3个数相加，即 $1+2+3=6$ 。第二个完全数是28，它有约数1、2、4、7、14、28，除去它本身28外，其余5个数相加，即 $1+2+4+7+14=28$ ，以此类推。

请使用海龟编辑器设计程序，输入一个正整数，计算出该数字范围内的所有完全数，并存储在一个列表中，最后输出该列表。

输入输出示范：

1.输入：请输入一个数，求该数范围内的完全数：5000

输出：5000范围内完全数：[6, xx, xxx]

2.输入：请输入一个数，求该数范围内的完全数：20000

输出：20000范围内完全数：[6, xx, xxx, xxxxx]

请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

### 【程序要求】

- 控制台显示“请输入一个数，求该数范围内的完全数：”；
- 通过控制台输入一个整数，求该数范围内的完全数；
- 控制台显示结果，格式为“xxx范围内完全数：列表”

### 【参考代码】

```

: 1 def Per_Num(n):
2     a = 1
3     b = n
4     total = 0
5     while a < b:
6         if n % a == 0:
7             total += a + b
8             a += 1
9             b = n / a
10    if a == b and a * b == n:
11        total += a
12    if total - n == n:
13        return 1
14    else:
15        return 0
16 Per_list = []
17 n = int(input("请输入一个数, 求该数范围内的完全数: "))
18 for i in range(2, n):
19     if Per_Num(i):
20         Per_list.append(i)
21 print(f"{n}范围内的完全数:", Per_list)

```

## 操作题2

有一群小猫，意外获得了一堆小鱼干，于是他们决定玩一个游戏，获胜者将获得全部小鱼干。

游戏规则如下：全部小猫围成一圈，从 1 开始进行编号，任选 1 只小猫开始报数，报到  $k$  的小猫退出圈子，从下一只小猫继续游戏，留在最后的那只小猫获胜。例如有 5 只小猫，编号依次为 1 到 5 的正整数，从 2 号小猫开始报数，每当报到 3 时，那只小猫淘汰出圈。那么依次淘汰编号为 4、2、1、3 的小猫，最终编号为 5 的小猫获胜。

请使用海龟编辑器设计程序，依次输入3个正整数，分别是小猫总数量 num，开始报数小猫的编号 begin 和每次淘汰出圈的号码 k，输出最终获胜小猫的编号。

输入输出示范：

1、输入：

请输入小猫总数量：5

请输入开始报数的小猫编号：2

请输入每次淘汰出圈的号码：3

输出：

win: 5

2、输入：

请输入小猫总数量：10

请输入开始报数的小猫编号：3

请输入每次淘汰出圈的号码：6

输出：

win: 5

请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

### 【程序要求】

1、能够出现输入的提示信息；

2、能够打印出最后获胜小猫的编号。

### 【参考代码】

```
1 def f(num, begin, k):
2     queue = list(range(1, num + 1)) # 创建报数编号列表
3     go_out = begin - 1 # 定义出局者下标
4     for i in range(num - 1):
5         go_out = (go_out + k - 1) % len(queue)
6         queue.pop(go_out) # 删除出局者
7     print('win:', queue[0]) # 列表中剩下的唯一一个元素就是获胜者
8
9 num = int(input("请输入小猫总数量: "))
10 begin = int(input("请输入开始报数的小猫编号: "))
11 k = int(input("请输入每次淘汰出圈的号码: "))
12
13 f(num, begin, k)
```

### 操作题3

对于一个给定的自然数M，求出所有的连续自然数段，这些自然数段中的数字之和为M

例：1998+1999+2000+2001+2002 = 10000

### 【输入格式】

一个整数 M ( $10 \leq M \leq 100000$ )

### 【输出格式】

每行两个自然数，第一个数小于第二个数，这两个自然数及之间所有的整数之和为M

两个整数之间用一个空格隔开，所有输出行的第一个数按从大到小排列

请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

输入输出示范：

- 程序样例1

输入：

10

输出：

1 4

- 程序样例2

输入：

10000

输出：

18 142

297 328

388 412

1998 2002

## 【参考代码】

```
1 num = int(input())
2 for i in range(1,num):
3     sum1 = 0
4     ge = 0
5     for j in range(i, num):
6         sum1 += j
7         ge += 1
8         if sum1 == num:
9             print(i,ge+i-1)
10            break
11        elif sum1 >= num:
12            break
```

### 操作题4

有5个砝码，质量分别为1、3、9、27、81，这5个砝码可以组合称出1到121之间任意重量的物体（注：左盘为置物盘，右盘为砝码盘，且砝码允许放在置物盘和砝码盘）。

现要求使用海龟编辑器设计程序，从控制台输入物体的重量（正整数），然后计算置物盘和砝码盘各使用了哪些砝码，并存储在相应的列表中输出。

输入输出示范：

- 输入：请输入物体的重量（正整数）：11

输出：置物盘中的砝码为:[1]

砝码盘中的砝码为:[9, 3]

- 输入：请输入物体的重量（正整数）：20

输出：置物盘中的砝码为:[9, 1]

砝码盘中的砝码为:[27, 3]

请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

### 【程序要求】

- 控制台显示“请输入物体的重量（正整数）：”；
- 通过控制台输入物体的重量，然后计算置物盘和砝码盘各使用了哪些砝码，并存储在相应的列表中输出；
- 控制台显示结果，格式为"置物盘中的砝码为:[9, 1]"

"砝码盘中的砝码为:[27, 3]"

### 【参考代码】

```

1 def reckon():
2     weight = []
3     a = [0, 1, -1]
4     b = [0, 3, -3]
5     c = [0, 9, -9]
6     d = [0, 27, -27]
7     e = [0, 81, -81]
8     n = int(input("请输入物体的重量 (正整数) : "))
9     for i in range(3):
10        for j in range(3):
11            for k in range(3):
12                for m in range(3):
13                    for h in range(3):
14                        if a[i] + b[j] + c[k] + d[m] + e[h] == n:
15                            weight.append(e[h])
16                            weight.append(d[m])
17                            weight.append(c[k])
18                            weight.append(b[j])
19                            weight.append(a[i])
20
21     left = []
22     right = []
23     for q in range(5):
24         if weight[q] > 0:
25             right.append(weight[q])
26         elif weight[q] < 0:
27             left.append(-weight[q])
28     print('置物盘中的砝码为:', left)
29     print('砝码盘中的砝码为:', right)
30     return 0

```

2022年

小学Nemo

操作题1

**【题干描述】**

绘制雪花：观察下边的图形，编写程序把它画出来



**【程序要求】**

- ①隐藏画笔角色；
- ②用画笔相关积木绘制雪花。雪花枝干和枝桠的边长、画笔的粗细没有限制，但要画出如图所示结构一模一样的蓝色雪花图案。

**【程序效果】**

[https://static.codemao.cn/new\\_question/rymWzg375](https://static.codemao.cn/new_question/rymWzg375)

**【程序预置版BCM】**



**【程序完整版BCM】**



## 操作题2

### 【题干描述】

需要做出这么一个程序，能够询问多少数字并且每个数字添加到列表中。找出出现次数最多的数字，如果有多个数字出现次数相同，则答案为后输入的数字。



### 【程序要求】

(1) 运行程序，圣诞猫询问“请问要输入几个数字？”

(2) 圣诞猫依次询问“请输入第x个数字”，将输入的各个数字存放在列表中。（x是指第几个，例如：请输入第1个数字1；请输入第2个数字）

(3) 找出其中出现次数最多的数字，小猫说“出现次数最多的数字是xxx”。如果有多个数字出现的次数相等，都是最多的，小猫说出最后一个。（例如：小猫说“出现次数最多的数字是26”）

**【程序效果】**

[https://static.codemao.cn/new\\_question/HklzxVI3m5](https://static.codemao.cn/new_question/HklzxVI3m5)

**【程序预置版BCM】**



**【程序完整版BCM】**



### 操作题3

#### 【题干描述】

暑假来了！小明开始了暑期的阅读计划，他从图书馆借了一本书，这本书一共有249页，在前三天他每天以相同的速度看完了一部分内容，并想要在一周之内读完这本书。为了帮助小明计算自己的阅读速度，请你设计一个“阅读速度计算”的程序，自动判定小明的阅读速度，并给出合理规划建议。

#### 【程序要求】

- 1、运行程序，弹出询问框，询问：“前三天一共看了多少页书？”，用户输入前三天看书的总页数。
- 2、判断用户输入总页数是否满足要求，页数要求为整数且在书本页数的范围，若不满足要求对话提示“页数输入不符合要求。”。
- 3、系统**自动计算**这本书剩余页数以同等速度看完所需要的天数，如果剩下的天数大于4天，则使用“对话（）”积木，显示“从第4天起每天需要看（）页”；如果剩下的天数 $\leq 4$ 天，则使用“对话（）”积木，显示“保持阅读速度，能在一周内阅读完毕，加油喔！”

**备注：**默认后4天每天看同样的页数，页数可采取向上舍入进行取整。

- 4、此程序在显示了“对话（）”积木内容后，等待2秒，可以无限重复进行阅读速度的计算。

#### 【输入输出范例】

组别	1	2	1	2	5
输入	-1	250	35	81	127
输出	页数输入不符合要求	页数输入不符合要求	从第4天起每天需要看54页	从第4天起每天需要看42页	保持阅读速度，能在一周内阅读完毕，加油喔！

**【程序效果】**



**【程序预置版BCM】**



### 【程序完整版BCM】



## 操作题4

### 【题干描述】

中位数 (Median) 又称中值，统计学中的专有名词，是按**顺序排列**的一组数据中居于中间位置的数，代表一个样本、种群或概率分布中的一个数值，其可将数值集合划分为相等的上下两部分。对于有限的数集，可以通过把所有观察值高低排序后找出正中间的一个作为中位数。如果数据的个数是奇数，则中间那个数据就是这组数据的中位数；如果数据的个数是偶数，则中间那2个数据的平均数就是这组数据的中位数。

例如：3、2、1的中位数是2；1、4、6、7的中位数是5。

请你使用给定的素材，根据要求设计一个“判断中位数”程序。

### 【程序要求】

- 1.程序开始，通过询问的方式让用户依次输入5个随机的数据，数据存储到列表里，对应列表显示在舞台上。
- 2.程序能对所输入的5个数据按照小到大排序。
- 3.排序完成后，以新建对话框的形式输出这组数的中位数。如输入1、9、4、10、5，则最终对话内容为“这组数据的中位数是：5。”

### 【输入输出范例】

	输入	输出
样例1	10,3,7,5,1	这组数据的中位数是：5
样例2	89,24,55,192,100	这组数据的中位数是：89
样例3	-15,12,0,-2,15	这组数据的中位数是：0

### 【程序效果】



**【程序预置版BCM】**



**【程序完整版BCM】**

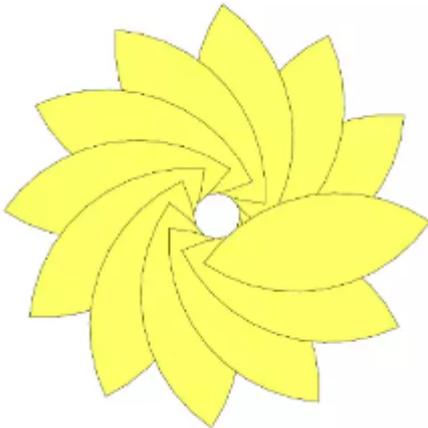


## 小学Kitten

### 操作题1

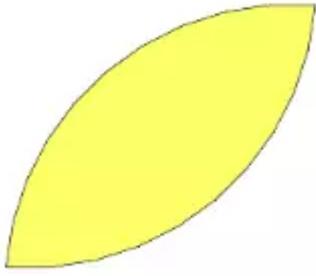
#### 【题干描述】

请使用源码编辑器的“画笔”功能，绘制出下图所示的花朵图案。

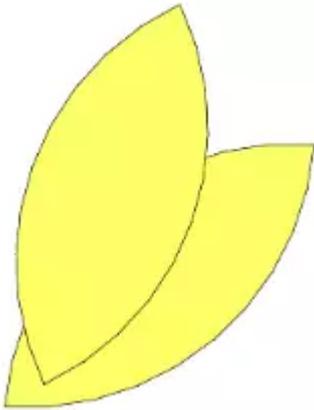


#### 【程序要求】

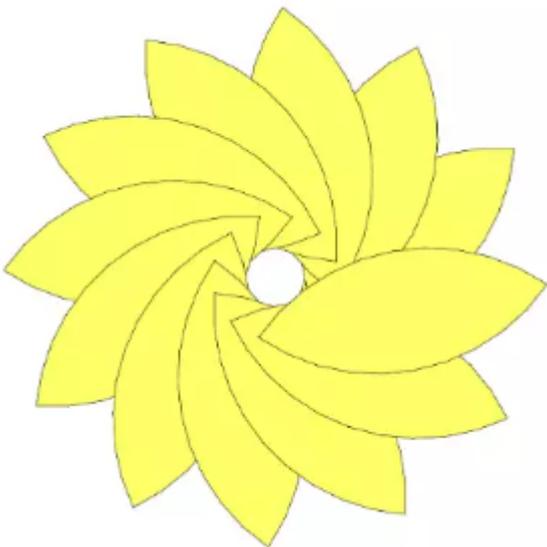
- 1、先绘制出一片花瓣，轮廓为黑色，填充颜色自定；



2、如下图所示，相邻花瓣大小相同，顶点需要相隔一段距离，不能重合；



3、在1、2的基础上，绘制出12个大小相同的花瓣，组合成为花朵图案；



4、绘制完成后，“画笔”角色需隐藏。

## 【预置程序】



画花朵-预置.bcm

附件大小：22.2KB

## 【参考程序】



花朵完成版.bcm

附件大小：26.5KB

## 操作题2

如果三组数字 $a$ 、 $b$ 、 $c$  ( $a < b < c$ ) 满足 $a^2 + b^2 = c^2$ ，则称这三组数字为勾股数。

例如3、4、5就是一组勾股数 ( $3^2 + 4^2 = 5^2$ )、5、12、13也是一组勾股数

( $5^2 + 12^2 = 13^2$ )，但是1、2、3就不是一组勾股数 ( $1^2 + 2^2 \neq 3^2$ )。

请你设计一个程序，当用户从小到大依次输入三个数字后，程序能够判断这三个数字是否为一组勾股数。

请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

## 【程序要求】

- 1、用户能够依次输入三个数字；
- 2、程序依照勾股数的定义，对这三个数字的数量关系进行计算；
- 3、如果这三个数字是勾股数，则新建对话框提示“是勾股数”，否则提示“不是勾股数”；
- 4、提示：以下数字组合也是勾股数，供你调试程序：(6, 8, 10)、(8, 15, 17)、(7, 24, 25)。

## 【预置程序】



勾股数-预置.bcm  
附件大小: 32.8KB

### 【参考程序】



勾股数完成.bcm  
附件大小: 40.5KB

### 操作3

#### 【题干描述】

请编写一个程序，它的功能是这样的：在对话框里任意输入一句话，再输入这句话中出现的字，程序能自动查找这个字在这句话中出现的位置并显示，如果没有则显示“不存在”。例如：输入“今天的天气真不错。”，再输入“天”，输出“在第2，4位”。例如：输入“今天的天气真不错。”，再输入“我”，输出“不存在”。

请点击下方编辑器进行答题，答题过程中注意点击“保存作品”，完成后点击右侧“退出全屏”按钮，继续回答其他题目。

#### 【程序要求】

- 1、运行程序时，程序通过询问方式让用户依次输入一句话和一个要查询的字；
- 2、如果这个字不在句子中出现，则通过对话框显示“不存在”；
- 3、如果这个字在句子中出现，则通过对话框并按照规定格式显示该字的位置，例如，“在第2位”（这个字在句中只出现一次）、“在第1，5，7位”（这个字在句中多次出现）。

#### 【效果参考】



#### 【程序预置】



查字的位置-预置程序.bcm  
附件大小: 86.5KB

#### 【程序参考】



查字的位置-参考程序.bcm  
附件大小: 150.4KB

### 操作4

#### 【题目描述】

某个猴子族群选新猴王的方法是: 让所有候选猴子围成一圈, 从某位置起顺序编号为1, 2, 3, 4, .....号, 直到最后一只猴子; 接着从第1号开始报数, 每轮从1报到6, 凡报到6的猴子就退出; 下一只猴子从1开始以同样的方式报数。如此不断循环, 所有报到6的猴子全部淘汰, 直到最后只剩下一只猴子就将它选为猴王。请用编程计算出一定会当选猴王的编号。

请点击下方编辑器进行答题, 答题过程中注意点击“保存作品”, 完成后点击右侧“退出全屏”按钮, 继续回答其他题目。

### **【程序要求】**

- 1、当程序开始时, 通过询问方式让用户输入猴子的数量;
- 2、根据用户输入的猴子数量, 向“队列”列表中按顺序添加对应的编号1,2,3.....;
- 3、依次将报数报到6的猴子从“队列”中删除, 直到最后只剩一只猴子; 例如, 当猴子数量为10时, 最后剩下的是编号为3的猴子;
- 4、“编程猫”角色说出当选猴王的猴子的编号。

### **【效果参考】**



### 【程序预置】



猴子选大王-预置程序.bcm

附件大小：33.1KB

### 【程序参考】



猴子选大王-参考程序.bcm

附件大小：43.0KB

## 中学Kitten

### 操作1

### 【题干描述】

几何图形看似简单，却能延伸出最复杂多变的逻辑。在自然界，有六边形的蜂巢和雪花，有黄金比例的鹦鹉螺。在人类文明伊始，便有线形的长城，三角的胡夫金字塔，和长方状的帕特农神庙。几何图形有一种难以言表的美感，圆形的圆润，方形的刚硬，三角形的锐利都是图形给我们的感觉，而正多边形是几何图形中一类特殊的存在。正多边形是指二维平面内各边相等，各角也相等的多边形，也叫正多角形。请用编程制作一个可以画出指定边数的正多边形的程序（推荐边长为100）。

参考公式：

n代表边数

正n边形的内角和度数为： $(n-2) \times 180^\circ$

正n边形的一个外角为： $360^\circ \div n$

### 【程序要求】

- 1、创建一个“绘制一个正多边形”的有参函数，参数为“边数”；
- 2、程序运行时，先接收用户输入的边数，如果边数是小数，那么程序提示用户“边数不得是小数”，如果边数小于3，那么程序提示用户“边数不得小于3”。这两种情况都没出现时，程序才调用“绘制一个正多边形”函数，同时将用户输入的边数作为函数的参数；
- 3、对“绘制一个正多边形”函数体编程，结合参考公式设计程序，实现绘制指定边数的正多边形

### 【参考程序】



正多边形

附件大小: 39.3KB

## 操作2

### 【题干描述】

用1、2、3、4共4个数字能组成多少个互不相等、且无重复数字的三位数，请你编写程序，将所有符合要求的结果统计出来吧。

### 【程序要求】

- 1、在“结果”列表中只保留满足题干要求的数字排列情况；
- 2、需使用算法生成符合要求的数据，不要直接在“结果”列表中手工填写数字；
- 3、用“编程猫”角色说出共有多少个符合条件的数字。

### 【参考程序】



结果运算

附件大小: 83.7KB

## 操作3

### 【题目描述】

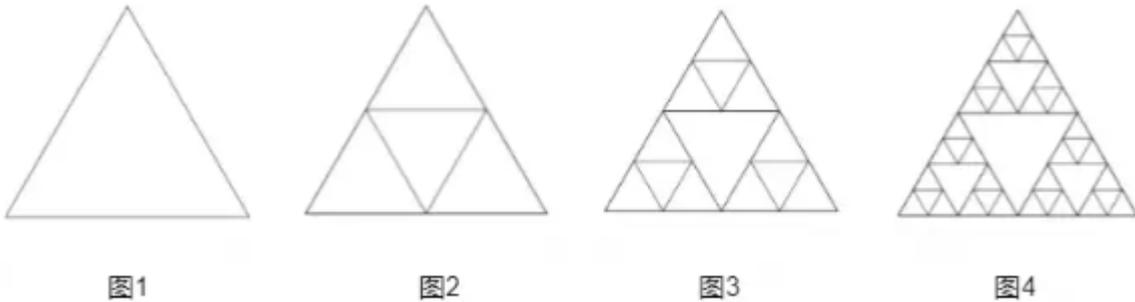
谢尔宾斯基三角形是最经典的分形图形之一，它由波兰数学家谢尔宾斯基在1915年提出，它由许多大小不等的等边三角形构成。

谢尔宾斯基三角形的画法如下：

- (1) 画一个等边三角形，并沿三条边中点的连线将它等分为4个小三角形；
- (2) 排除中间的小三角形，对其余3个小三角形再执行4等分的操作；

(3) 重复上述步骤，可以得到更多、更小的等边三角形。最终这些大小不同的等边三角形就构成了谢尔宾斯基三角形。

其绘制过程如下图所示。



### 【程序要求】

- 1、绘制一个边长为400的等边三角形，如图1所示；
- 2、在等边三角形基础上，沿三条边中点的连线将它等分为4个小等边三角形，如图2所示；
- 3、按照谢尔宾斯基三角形的画法，最终绘制出图4所示图形；
- 4、图形不能超出屏幕范围。

注：能够绘制出图1、图2、图3图形，可适当得分。

### 【程序预置】



谢尔宾斯基三角形-预置程序.bcm  
附件大小：32.4KB

### 【程序参考】



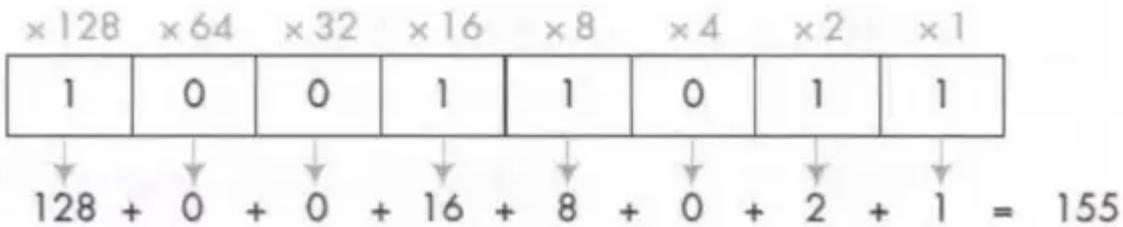
谢尔宾斯基三角形-参考程序.bcm  
附件大小：38.6KB

#### 操作4

##### 【题干描述】

二进制数仅由两个数字组成：0和1。计算机使用二进制数进行操作和通信。相反，人类更擅长使用由数字0到9组成的十进制数。整数的二进制转换为十进制的转换规则如下：

在整数二进制中，我们称最右边的数是第0位，其左边的为第1位，向左依次增加，因此，最左边为最高位7。在二进制转换为十进制时，用相应位次的数值乘以2的位次幂并依次相加。如图中即为二进制数10011011的转换方法：



计算过程： $(1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = 155$ 。

请你编写程序，制作将整数二进制数转换为十进制数的转换器。具体要求如下：

##### 【程序要求】

- 1、当程序开始后，舞台区上方显示8位数字0的造型，同时“转换机器人”通过对话框提示用户“输入二进制数（最长8位）”；
- 2、系统对用户输入的二进制数字进行判断：
  - 二进制数是否超过八位；不满足条件则对话提示“二进制数字超过了8位”；
  - 二进制数的每一位是否均为有效数字（0或1）；不满足条件则对话提示“二进制数包含无效位数”；
  - 判定用户输入合理后，将舞台上8位数字0的造型转换为用户输入的二进制

数。

(如：用户输入1100，则舞台上数字由“00000000”转换为“00001100”)

3、根据转换规则计算对应的十进制数，并通过角色“转换机器人”对话显示转换结果，如“1100=12”；

#### 【参考效果】



#### 【预置程序】



进制转换-预置.bcm

附件大小：335.9KB

#### 【参考程序】



进制转换-完整.bcm

附件大小: 361.7KB

## 中学Python

### 操作题1

#### 【题干描述】

请编写一个程序，完成要求的功能。输入某年某月某日，判断这一天是这一年的第几天。

#### 【程序要求】

- 提示用户输入待查询的年月日；
- 程序计算从1月1日起，输入日期是当年的第几天；
- 输出查询日期是这一年的第几天。

#### 【程序效果】

##### 控制台

```
请输入查询日期:20211213  
您查询的日期是这一年的第347天  
程序运行结束
```

#### 【程序完整版】

```

1 import datetime
2 import time
3
4 dtstr = str(input('请输入查询日期:'))
5 dt = datetime.datetime.strptime(dtstr, "%Y%m%d")
6 another_dtstr = dtstr[:4] + '0101'
7 another_dt = datetime.datetime.strptime(another_dtstr, "%Y%m%d")
8
9 print("您查询的日期是这一年的第{}天".format(int((dt-another_dt).days) + 1))

```

### 【测试数据】

输入	输出
20211213	347
20220207	38
20300715	196
19951218	352
19000526	146

### 操作题2

### 【题干描述】

一群猴子要选新猴王。新猴王的选择方法是：让N只候选猴子围成一圈，从某位置起顺序编号为1~N号。从第1号开始报数，每轮从1报到3，凡报到3的猴子即退出圈子，接着又从紧邻的下一只猴子开始同样的报数。如此不断循环，最后剩下的一只猴子就选为猴王。请问是原来第几号猴子当选猴王？

### 【程序要求】

1. 让用户输入猴子总数（1至100之间）
2. 利用列表操作完成选猴王的算法
3. 输出当选猴王的猴子编号
4. 如果用户输入猴子总数不在要求范围内，要有相应的提示。

### 【程序效果】

```
请输入猴子总数（1-100之间）： 11  
当选猴王的猴子编号是： 7  
程序运行结束
```

### 【程序完整版】

```

1 n = int(input("请输入猴子总数 (1-100之间): "))
2 monkey = []
3 timer = 0
4 count = 0
5 if(n > 0 and n <= 100):
6     for i in range(1, n+1):
7         monkey.append(i)
8     while(len(monkey) > 1):
9         timer += 1
10        count += 1
11        if(count > len(monkey)):
12            count = 1
13        if(timer == 3):
14            timer = 0
15            monkey.pop(count-1)
16            count -= 1
17        print("当选猴王的猴子编号是: " + str(monkey[0]))
18 else:
19     print("请输入正确的猴子总数")

```

### 【测试数据】

输入	输出
11	7
110	请输入正确的猴子总数
0	请输入正确的猴子总数

100	91
50	11

### 操作题3

输入一行字符（汉字除外），分别统计出其英文字母、空格、数字、和其他字符的个数。

#### 【程序要求】

- 1、定义一个main()函数，实现以上统计输出的功能。
- 2、函数功能需要提示用户输入内容，输入的内容不能包含汉字。
- 3、函数对输入内容统计英文字母、空格、数字和其他字符的个数。
- 4、函数格式化输出字母、空格、数字和其他字符的个数。
- 5、调用main()函数，成功实现题目要求的功能。

#### 【程序效果】

##### 控制台

```
请输入内容（除汉字外）： I like Python3!  
字母有11个，空格有2个，数字有1个，其他字符有1个  
程序运行结束
```

#### 【程序参考】

```

1 def main():
2     s = input('请输入内容 (除汉字外) : ')
3     letter = 0 # 统计字母
4     space = 0 # 统计空格
5     digit = 0 # 统计数字
6     other = 0 # 统计其他字符
7     for c in s:
8         if c.isalpha():
9             letter += 1
10        elif c.isspace():
11            space += 1
12        elif c.isdigit():
13            digit += 1
14        else:
15            other += 1
16    print('字母有{}个, 空格有{}个, 数字有{}个, 其他字符有{}个'.format(letter, s
17
18 main()

```

#### 操作题4

小明和妈妈去超市购物，但是有些商品没有价格，此时小明相当一个价格查询的程序，这样非常方便的可以查找的商品的价格。具体要求如下：

1、目前有商品(可直接使用)：

```

1 dic = {"水果": {"桔子": 4.5, "香蕉": 6.9, "荔枝": 23.8},
2        "蔬菜": {"蘑菇": 6.5, "菠菜": 3.6, "白菜": 4.5, "土豆": 1.4},
3        "肉制品": {"猪肉": 27.0, "牛肉": 60.0}}

```

2、运行程序时，控制台会显示'欢迎光临'和三个商品种类；

3、输入水果、蔬菜、或肉制品，将显示对应种类的商品名称，输入其他内容，需要重新输入；

4、输入控制台显示的商品名称，将按规定格式显示对应商品的价格，例如显示“土豆的单价是 1.4元”。若输入的商品名未出现在控制台，则需再次输入，例如输入豆角，则提示再次输入。

```
-----欢迎光临-----  
水果  
蔬菜  
肉制品  
请输入商品类别：豆制品  
请输入商品类别：其他  
请输入商品类别：蔬菜  
-----欢迎光临-----  
蘑菇  
菠菜  
白菜  
土豆  
请输入要查询的商品：豆角  
请输入要查询的商品：土豆  
土豆的单价是1.4元  
程序运行结束
```

### 【程序参考】

```
1 dic = {"水果": {"桔子": 5.0, "香蕉": 3.0, "荔枝": 13.8},
2       "蔬菜": {"茄子": 3.5, "白菜": 2.5},
3       "肉制品": {"猪肉": 38.0, "牛肉": 80.0}}
4 choice = ''
5 current_layer = dic
6 print("-----欢迎光临-----")
7
8 for key in current_layer:
9     print(key)
10 while choice not in current_layer:
11     choice = input("请输入商品类别: ")
12 current_layer = current_layer[choice]
13 print("-----欢迎光临-----")
14 for key in current_layer:
15     print(key)
16 while choice not in current_layer:
17     choice = input("请输入要查询的商品: ")
18 print("{}的单价是{}".format(choice, current_layer[choice]))
```