

2025 信息素养大赛小高组复赛真题 — 选择题答案解析

选择题

第 1 题

[题目] 运行下面程序，舞台上能看到几只小猫？

[答案] B

[解析]

- 本体+克隆体：绿旗被点击时，舞台上原本就有一只小猫（本体）。
- 本体没有移动位置，所以产生的克隆体都堆在一起，只能看见两只。

[知识点]

克隆体原理：克隆体继承本体的属性，如果本体位置不变，所有克隆体都会堆叠在同一位置。

青少年科创教育领航者

第 2 题

[题目] 运行下面程序，哪个说法是错误的？

[答案] D（花朵最终位置为(170, 0)）

[解析]

- 这道题考查的是「重复执行直到」的结束条件。

- 初始状态：x 坐标是 0，大小是 10。
- 条件判断：只要「大小 > 80」，循环就会停止。
- 最后一次循环前：当执行了 14 次时，x 坐标增加到了 140，大小增加到了 80 ($10 + 14 \times 5 = 80$)。此时 80 不大于 80，条件不满足，所以要再执行一次。
- 最后一次循环后：第 15 次执行，x 坐标增加到 150，大小增加到 85 ($80 + 5 = 85$)。此时 $85 > 80$ ，循环停止。
- 验证选项：最终大小是 85 (A 对)，说出最终 x 坐标 150 (B 对)，向右移动并变大 (C 对)。最终位置的 x 应该是 150，而不是 170，所以 D 说法错误。

[知识点]

循环控制：「重复执行直到」在条件满足时停止。注意最后一次循环的执行时机。

第 3 题

[题目] 运行下面程序，最后说出的内容是？

[答案] C (丝瓜)

[解析]

- 我们一步步模拟列表的变化：
- 初始加入 6 项：白菜、萝卜、西红柿、黄瓜、菠菜、西兰花。
- 删除第 2 项 (萝卜)：列表变成：1.白菜、2.西红柿、3.黄瓜...
- 末尾加入「茄子」：茄子排在最后，不影响前几项。
- 将第 3 项替换为「芹菜」：原本的「黄瓜」被替换了。列表变成：1.白菜、2.西红柿、3.芹菜。
- 在第 3 项前插入「丝瓜」：「丝瓜」挤到了第 3 个位置，原来的「芹菜」及后面的往后退一格。

- 最终第 3 项：丝瓜。

[知识点]

列表操作：删除会改变后续项的索引；插入会将原位置及后面的项后移；替换只改变指定项。

第 4 题

[题目] 运行下面程序，哪个说法是错误的？

[答案] D (变量 n 最终存储 4 个字符)

[解析]

- 字符串内容：「全国青少年信息素养大赛成功举办」，一共 15 个字。
- 提取规则：从最后 ($m=15$) 往前找，条件是「 m 除以 3 的余数 = 2」。
- 符合条件的数字 m ：从 15 往下数，除以 3 余数是 2 的数字分别是：14、11、8、5、2。
- 对应的文字：
 - 第 14 个字是：举
 - 第 11 个字是：赛
 - 第 8 个字是：素
 - 第 5 个字是：年
 - 第 2 个字是：国
- 组合结果：变量 n 最终连接起来是「举赛素年国」（C 对）。这明明是 5 个字符，而不是 4 个，所以 D 说法错误。

[知识点]

字符串处理：掌握字符索引、余数判断、倒序遍历等技巧。注意索引从 1 开始还是从 0 开始。

第 5 题

[题目] 运行下面程序，变量 num 的值是？

[答案] A (557)

[解析]

- 这道题考查的是「或（或者）」逻辑运算。
- 只要满足下面任何一个条件，数字就会被加进 num 里：
- 包含 5：数字里有「5」（比如 65、85、51）。
- $\text{金额}/8 > 5$ ：这个条件简化一下就是 $\text{金额} > 40$ 。
- 我们来挨个检查列表里的数字（大于 40 或者 包含 5）：
- 第 1 项 100： > 40 ，加！
- 第 2 项 34：都不满足，不加。
- 第 3 项 65：有 5 且 > 40 ，加！
- 第 4 项 85：有 5 且 > 40 ，加！
- 第 5 项 76： > 40 ，加！
- 第 6 项 49： > 40 ，加！
- 第 7 项 12：都不满足，不加。
- 第 8 项 43： > 40 ，加！
- 第 9 项 51：有 5 且 > 40 ，加！
- 第 10 项 88： > 40 ，加！
- 最终计算： $100 + 65 + 85 + 76 + 49 + 43 + 51 + 88 = 557$ 。

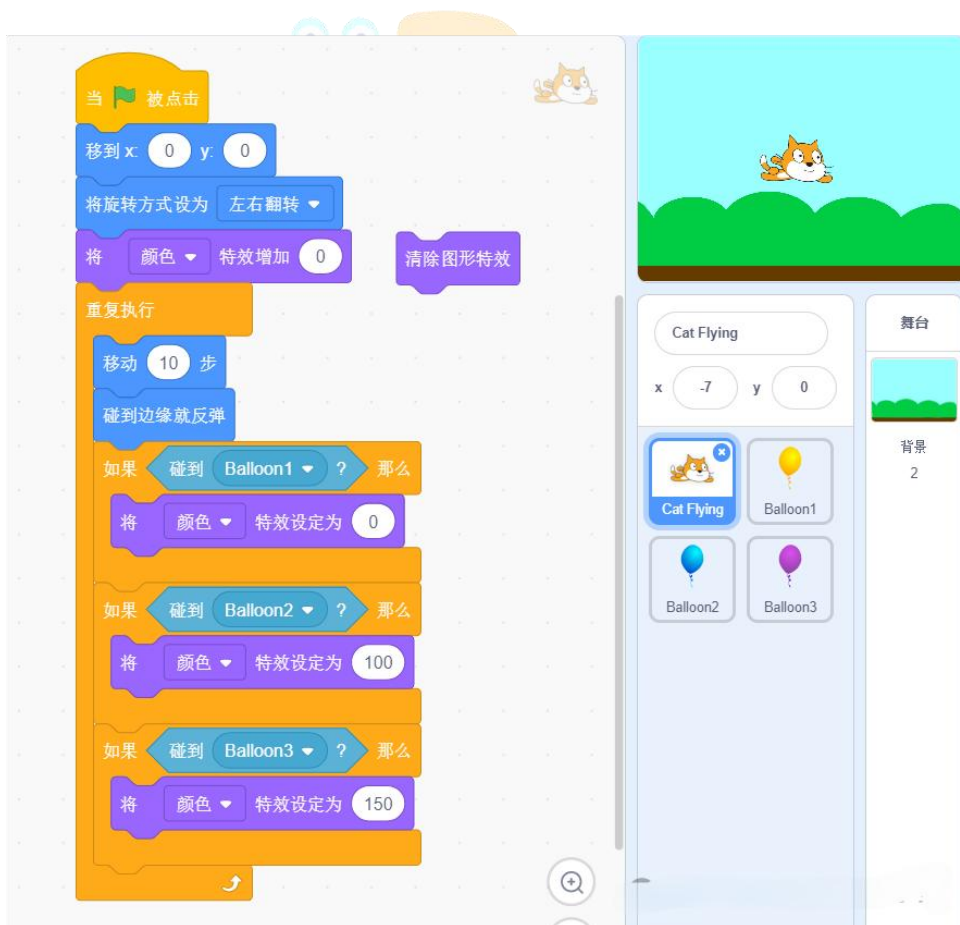
[知识点]

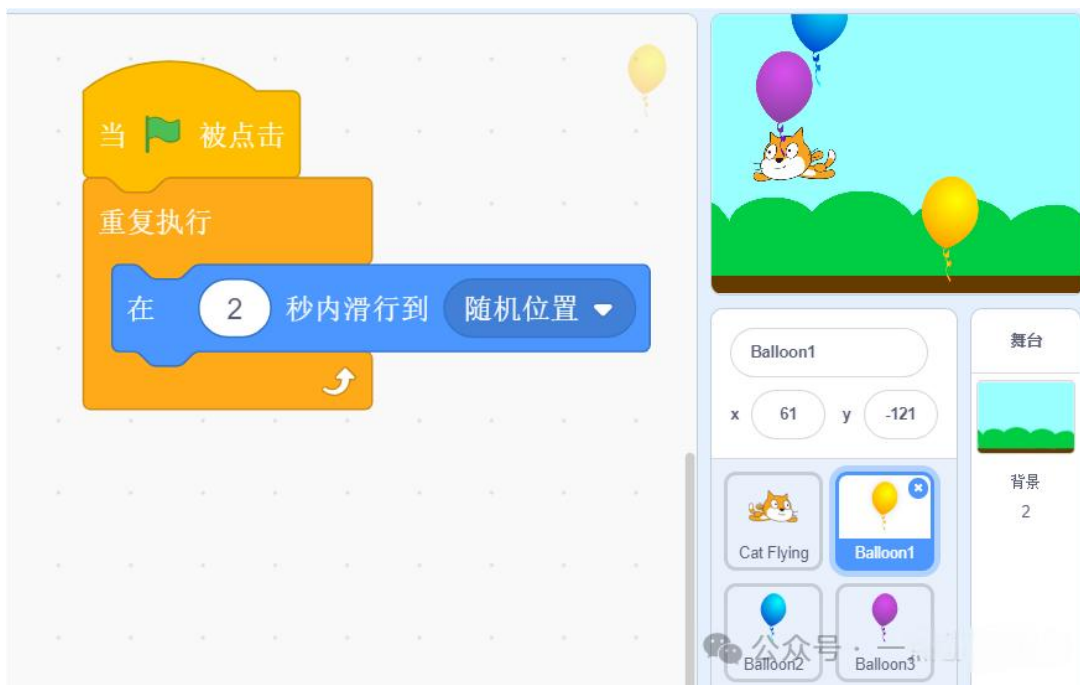
逻辑运算：「或」运算只要有一个条件满足就为真。列表遍历结合条件筛选是常见考点。

编程题

第 1 题

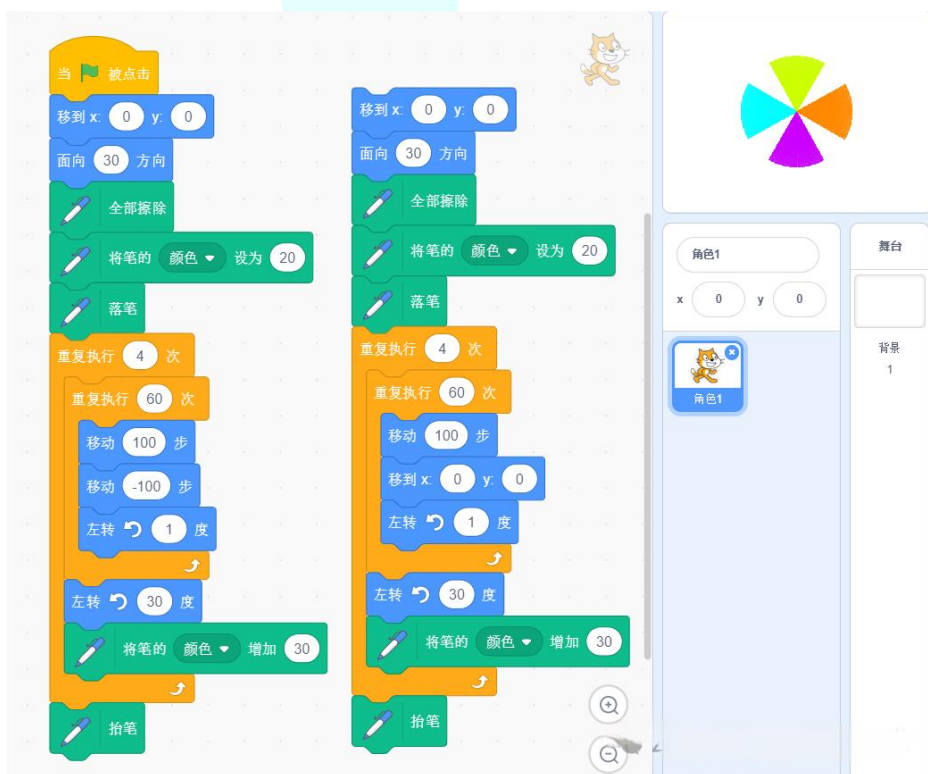
参考程序：



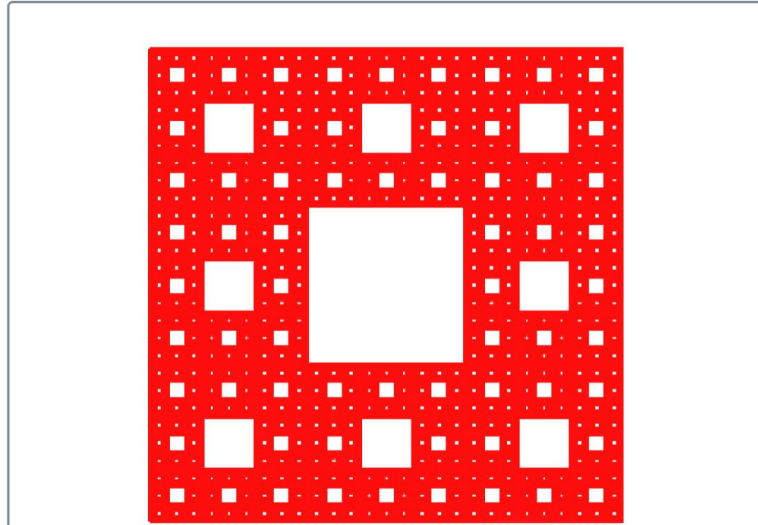


第 2 题

参考程序：

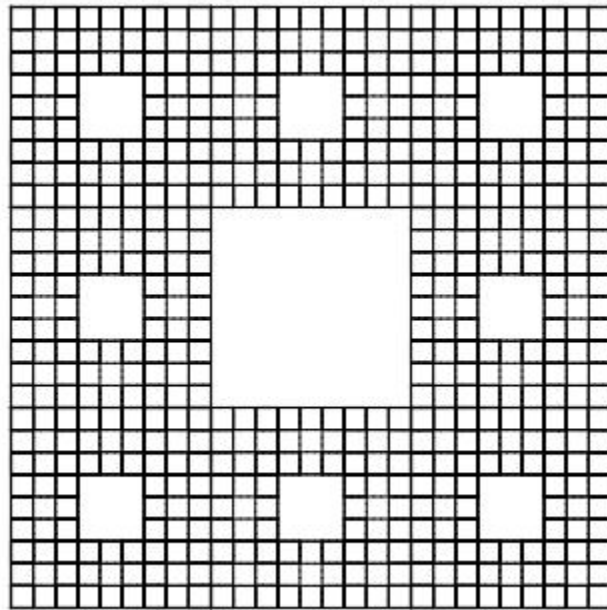


第 3 题



参考程序:

谢尔宾斯基地毯图案如下图所示:



问题描述:

该图案是将一个大的正方形平均分成 9 个小的正方形。

除中间块之外的所有正方形再进行 9 等分，依次类推，可以画出更多小正方形。

了解递归：

要解决上述问题，需要用到递归的方法。

递归方法就是将一个大的问题替换为若干个相同的小问题来解决。例如，我们可以把谢尔宾斯基地毯问题转化为绘制小正方形的问题。

递归方法可以分为两个步骤：递（递去）和归（归来）。

递去：将递归问题分解为若干个规模较小，与原问题形式相同的子问题，这些子问题可以用相同的解题思路来解决。

归来：当你将问题不断缩小规模递去的时候，必须有一个明确的结束递去的临界点（递归出口，在本程序中临界点就是最小正方形的边长），一旦达到这个临界点即就从该点原路返回到原点，最终问题得到解决。

青少年科创教育领航者

编程思路：

首先制作一个新积木，叫谢尔宾斯基地毯，并添加一个变量——边长。



这个问题的基本形状是一个正方形，我们可以写出如下代码绘制正方形的四条边。



以上代码可以绘制出的是一个正方形，我们可以将边长分成 3 段，重复执行 3 次来完成正方形绘制，为后续递归做铺垫（分三段的原因是一个大正方形分成 9 个小正方形，小正方形边长是大正方形 $1/3$ ）。



使用递归调用，在重复执行 3 次里面加入代码“谢尔宾斯基地毯（边长/3）”。



加入的这一个程序的意思是，在绘制大正方形时，先绘制边长/3 的小正方形，并且边长为边长/3（递归中的递去）。

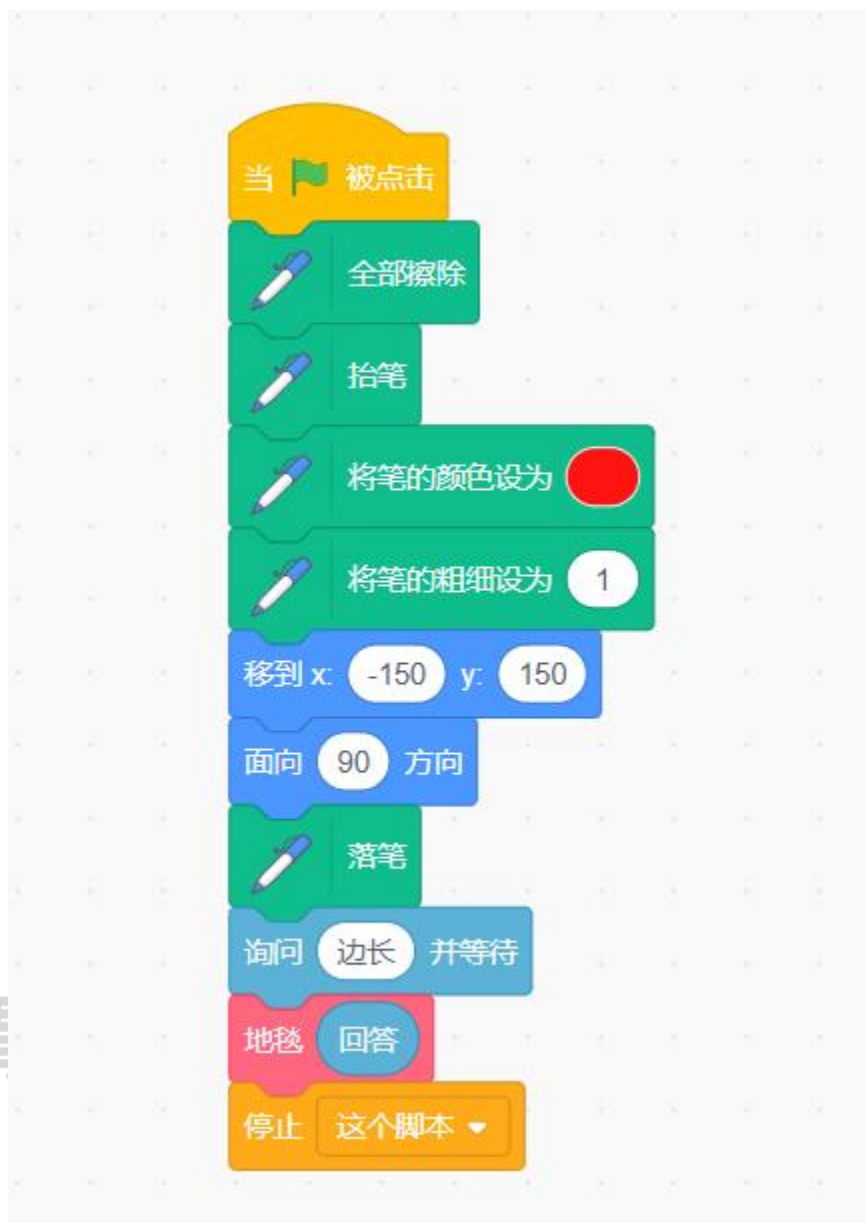
递去程序之后，边长不能无限缩小，没有最小值，也绘制不出最小的正方形，因此需要在绘制过程中限制最小边长，让程序回归原来要解决的问题，有了最小边长之后，程序就会从最小正方形开始绘制，逐渐将整个图形绘制完成。示例如下：

青少年科创教育领航者



通过上述程序就可以完成谢尔宾斯基地毯的图案方法，当我们需要时，写一个程序调用即可。
调用函数示例如下：

青少年科创教育领航者



第 4 题

参考程序：

(1) 计算真因子之和

输入数字：让用户输入一个正整数。

初始化变量：

输入的数：存储用户输入的数。

计数（当前检查的数）：用于循环（从 1 到输入的数 - 1）。

Sum（真因子之和）：初始化为 0，累加符合条件的因子。

遍历检查可能的因子：从 1 到该数-1，检查是否能整除。

累加真因子：如果能整除，将数字添加到 Sum 变量中。

（如果输入的数能被当前检查的数整除，则将其加到真因子之和。）

显示结果：显示真因子之和 Sum

(2) 判断盈利数

比较真因子之和和输入的数：

如果真因子之和 $>$ 输入的数 $\rightarrow$ 盈利数。

如果真因子之和 $=$ 输入的数 $\rightarrow$ 完全数。

如果真因子之和 $<$ 输入的数 $\rightarrow$ 亏损数。

输出结果：显示是否为盈利数，如果是输出 1，否则输出 0；

(3) 优化：

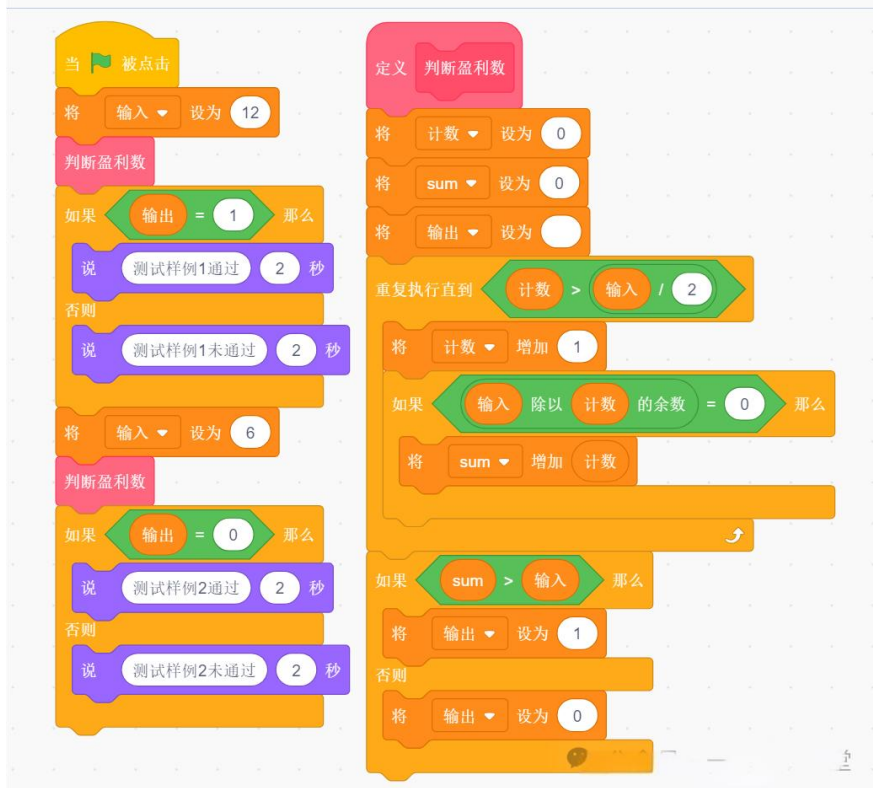
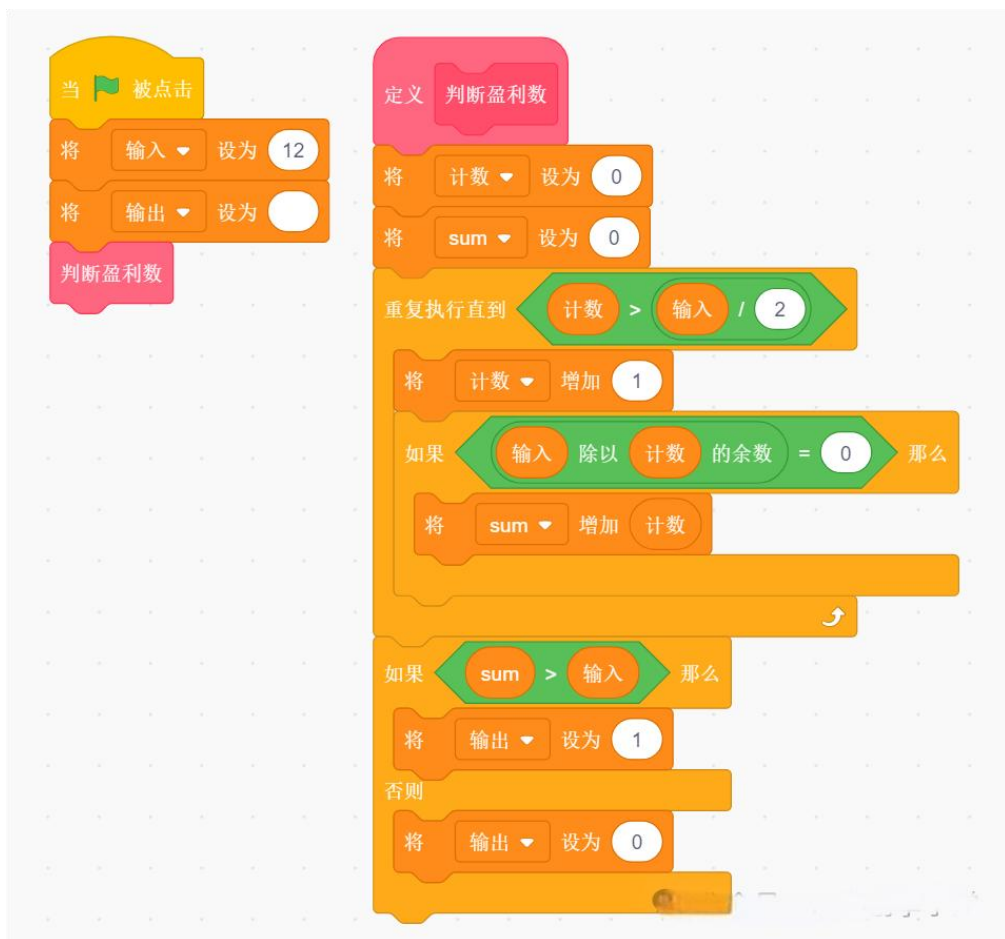
减少循环次数：只需检查到输入的数 / 2（因为大于一半的数不可能是因子）

重复直到 $<(\text{当前检查的数}) > ((\text{输入的数}) / (2)) >$

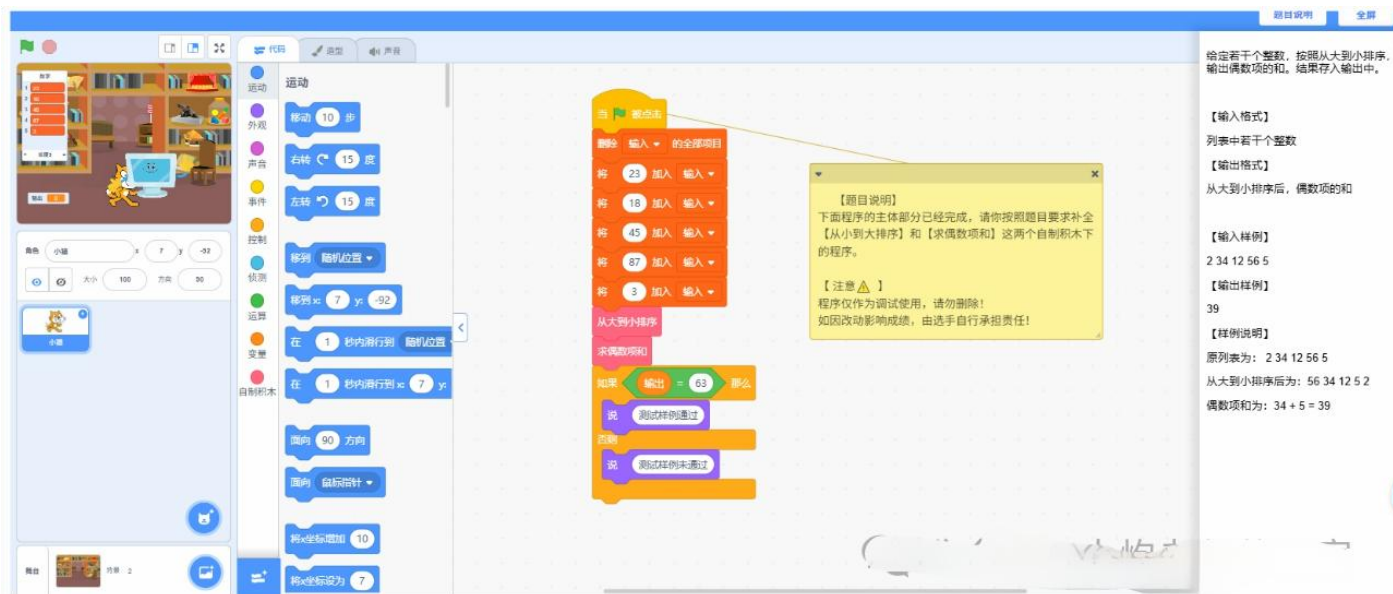
处理无效输入：添加判断确保输入是正整数。



青少年科创教育领航者



航者



青少年科创教育领航者

第 5 题

参考程序：

从大到小-降序排序

冒泡排序：

- (1) 若相邻的两个数中前一个数较小，则进行交换，使较小的数向后移动；
- (2) 数字列表相邻的两个元素进行比较，n 个数据，两两比较，比较 n-1 次；
- (3) 把每一轮比较交换的过程称为一次起泡，完成一次起泡后，已排好序的数就增加一个，要排序的数就减少一个，从而使下次起泡过程的比较运算减少

一次。因此，对于 n 个数据，各轮比较次数依次为 $(n-1)$, $(n-2)$, ..., 2 , 1 。
(每比较完一轮，下一轮少比较一个元素)

(4) **当前序号** 变量开始设为 1 ，每次从无序部分的第一个数据开始比较，每比较完一对数据，**当前序号** 变量增加 1 ，比较下一对数据，**当前序号** 变量表示需要比较的第一对数据中前一个(左边/前边)那个数据的编号，所有需要比较的数据对中前一个(左边/前边)的编号范围为 1 -【**数字**的项目数- 1 】（比较到倒数第二个元素）。

外层循环（控制轮次）作用：

每一轮将当前未排序部分的最大值“冒泡”到末尾。

内层循环（比较与交换）作用：

遍历列表，比较相邻元素：如果前一个元素 $<$ 后一个元素（降序排序），则交换它们。

每轮结束后，最大的元素已到位，末尾元素已排序，下一轮可减少比较次数。

定义 从大到小排序

将 轮次 设为 0

重复执行 数字 的项目数 - 1 次

将 当前序号 设为 1

重复执行 数字 的项目数 - 1 - 轮次 次

如果 数字 的第 当前序号 项 < 数字 的第 当前序号 + 1 项 那么

将 中转 设为 数字 的第 当前序号 项

将 数字 的第 当前序号 项替换为 数字 的第 当前序号 + 1 项

将 数字 的第 当前序号 + 1 项替换为 中转

将 当前序号 增加 1

将 轮次 增加 1

定义 输出偶数项的和

将 i 设为 1

将 输出 设为 0

重复执行 数字 的项目数 次

如果 i 除以 2 的余数 = 0 那么

将 输出 增加 数字 的第 i 项

将 i 增加 1



