

团副@蒲公英教育 两娃爸 | 蓝桥命题人 | 国际教育实践者 微信号: tuanfu06

由蓝桥青少考试委员会授权，于解密日之后在朋友圈分享STEMA试卷和等考真题。



STEMA 考试中级试卷真题

(2024 年 1 月)



目录

一、 STEMA 考试 Scratch 中级真题	1
二、 STEMA 考试 Python 中级真题	13
三、 STEMA 考试 C++ 中级真题	20
四、 STEMA 考试 Arduino 中级真题	28
五、 STEMA 考试 EV3 中级真题	33



一、STEMA 考试 Scratch 中级真题

一、选择题

第一题

运行下列哪段程序后，蜜蜂会向上移动？（ ）



- A.
- B.
- C.
- D.

第二题

运行以下程序，输入下列哪个数后，角色会说“未通过”？（ ）





A. 90

B. 85

C. 60

D. 58

第三题

运行以下程序后，n 的值为（ ）。



A. 17

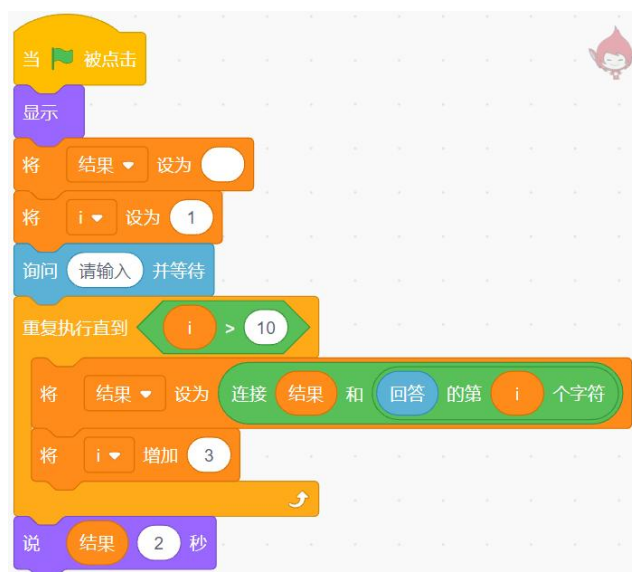
B. 19

C. 21

D. 23

第四题

运行以下程序，输入 unbelievable，角色会说（ ）。





- A. nlv
- B. uee
- C. ueeb
- D. nlve

第五题

已有列表“数据”如下图所示，运行以下程序后，角色会说（ ）。

数据	
1	42
2	35
3	28
4	17
5	89
6	71
7	48
8	7

当 绿旗 被点击

将 i 设置为 1

将 n 设置为 0

重复执行 数据 的项目数 次

如果 数据 的第 i 项 除以 2 的余数 = 0 那么

将 n 增加 数据 的第 i 项 的第 1 个字符

否则

将 n 增加 数据 的第 i 项 的第 数据 的第 i 项 的字符数 个字符

将 i 增加 1

说 n 2 秒

- A. 38
- B. 39
- C. 44
- D. 45

二、编程题

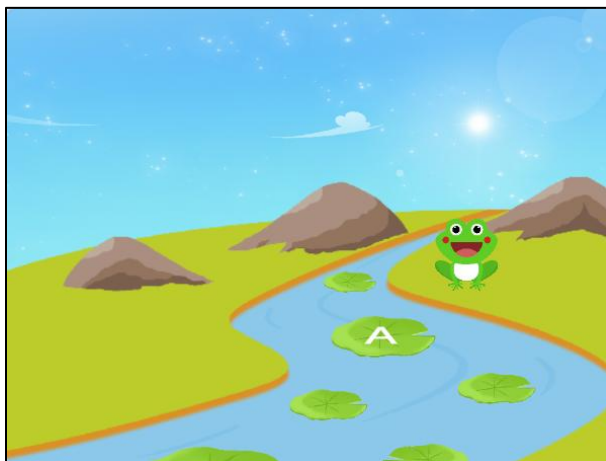
第一题

编程实现：

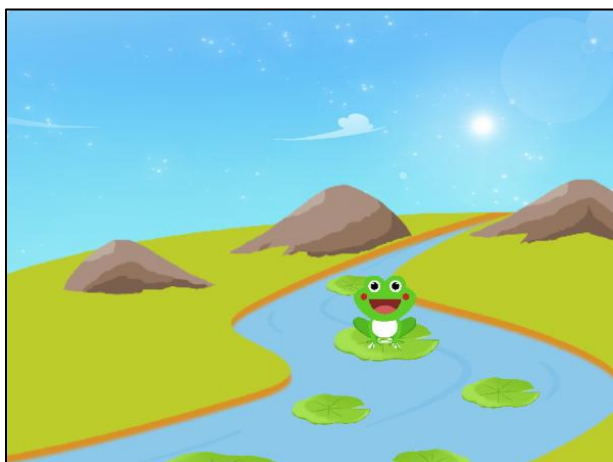
青蛙过河。

具体要求：

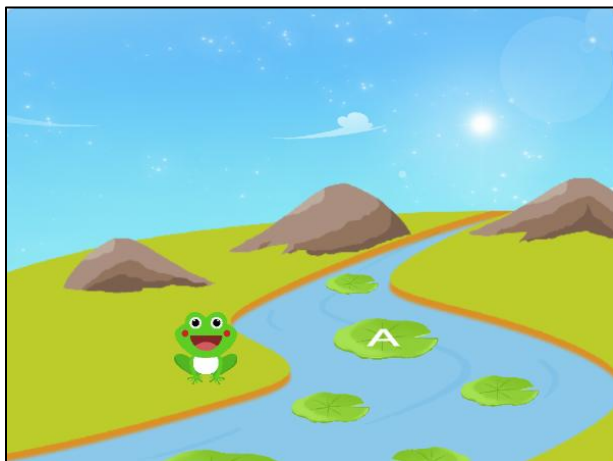
- 1) 点击绿旗，青蛙在右侧的岸上，如图所示；



2) 1 秒后，青蛙到达 A 荷叶上，如图所示；



3) 1 秒后，青蛙到达左侧的岸上，如图所示。



第二题

编程实现：

巡逻的直升机。

具体要求：



1) 点击绿旗，角色、背景如图所示；



2) 1 秒后，直升机从舞台右侧向舞台左侧缓慢移动（能看到移动过程）；

直升机在移动过程中螺旋桨持续转动；

4) 直升机在移动过程中逐渐变小（能看到变小过程），如图所示；



5) 直升机到达舞台左侧边缘后消失。

第三题

编程实现：

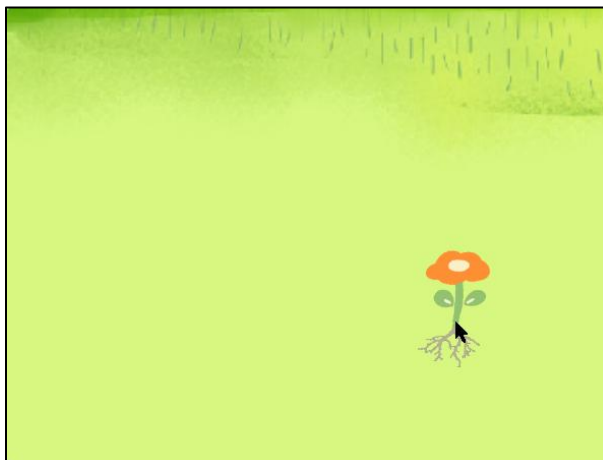
栽花。

具体要求：

1) 点击绿旗，角色、背景如图所示；

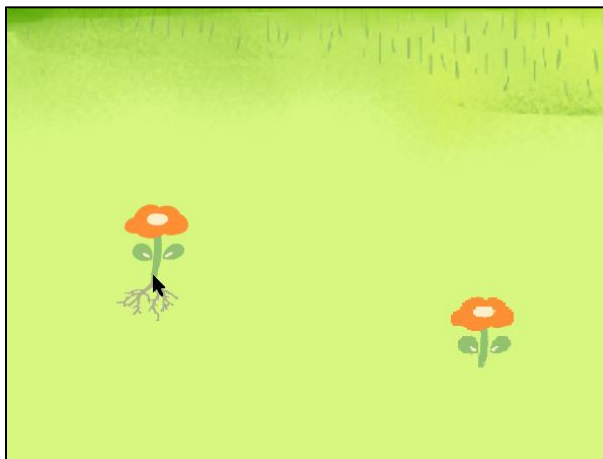


2) 按下空格键后，带根的花朵移动到鼠标所在位置，如图所示；



3) 移动鼠标，带根的花朵会持续随着鼠标移动；

4) 按下鼠标左键，会在鼠标点击处新栽一朵花（新栽的花看不到根）；



5) 每次按下鼠标左键，都会在鼠标点击处新栽一朵花，如图所示；



6) 如果花朵角色碰到舞台边缘或其它花朵，此时按下鼠标左键不栽花。

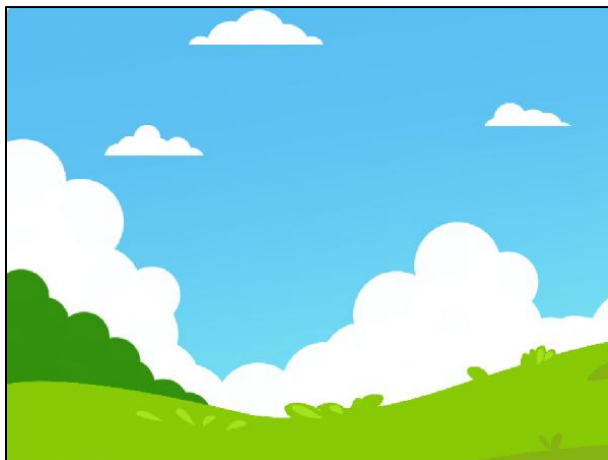
第四题

编程实现：

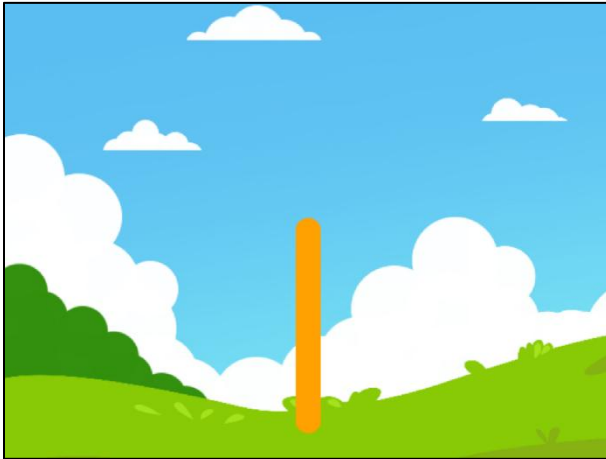
风车。

具体要求：

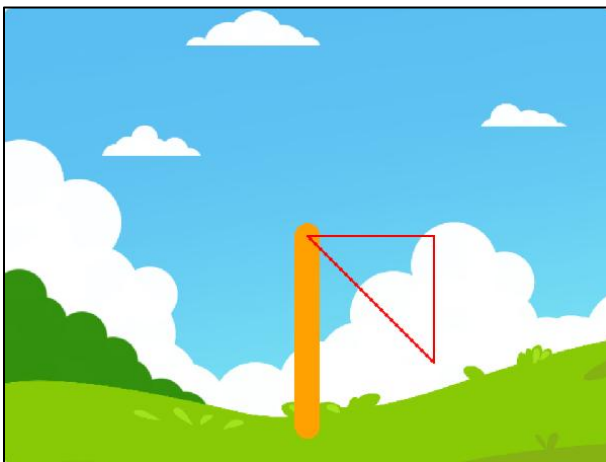
1) 点击绿旗，背景如图所示；



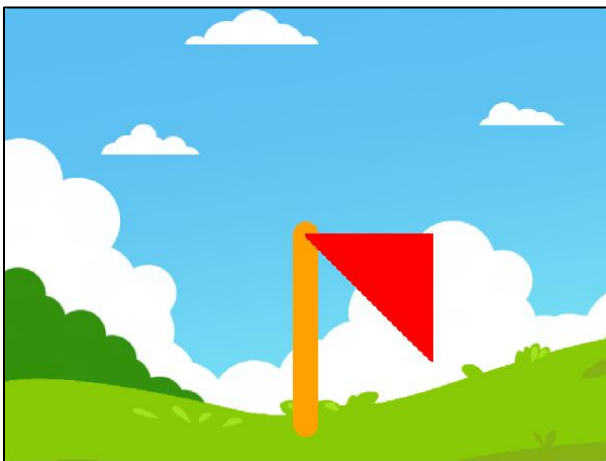
2) 1 秒后，在草地上绘制出一条竖直线段（长短、粗细和颜色不限），如图所示；



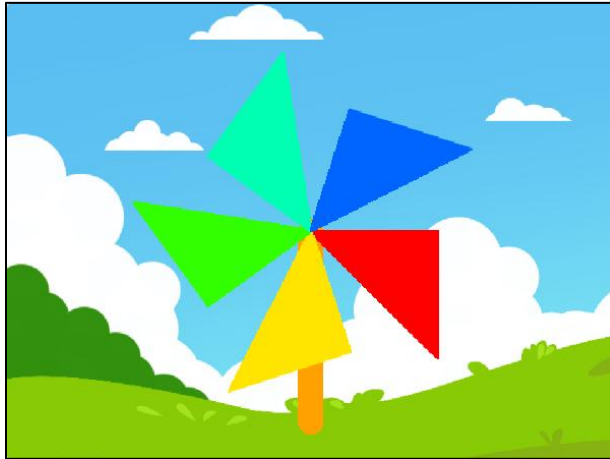
3) 1 秒后，在线段的上端绘制一个空心三角形（颜色、大小不限），如图所示；



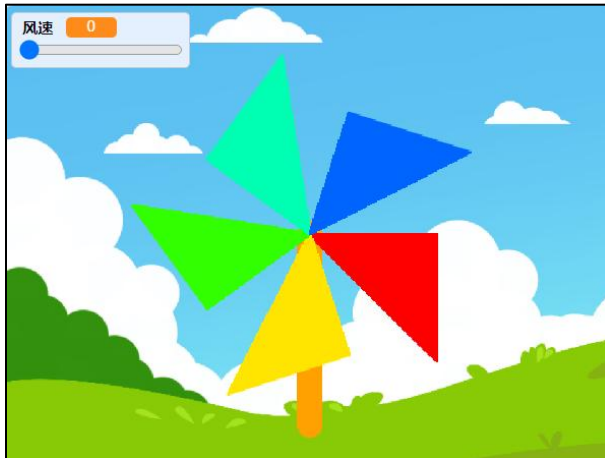
4) 然后，慢慢将空心三角形填充为实心三角形（颜色不限），如图所示；



5) 然后，呈现出 5 个均匀分布的实心三角形（三角形大小相同，颜色均不相同），如图所示；



6) 风车绘制完成后，左上角显示滑杆形式的变量“风速”，且其取值范围为 $0 \sim 2$ (包含 0 和 2，初始值为 0)，如图所示；



7) 拖动滑杆调整“风速”的值，当“风速”值为 0 时风车不转动，当“风速”值为 1 时风车缓慢转动，当“风速”值为 2 时风车快速转动。

第五题

编程实现：

专注力训练游戏。

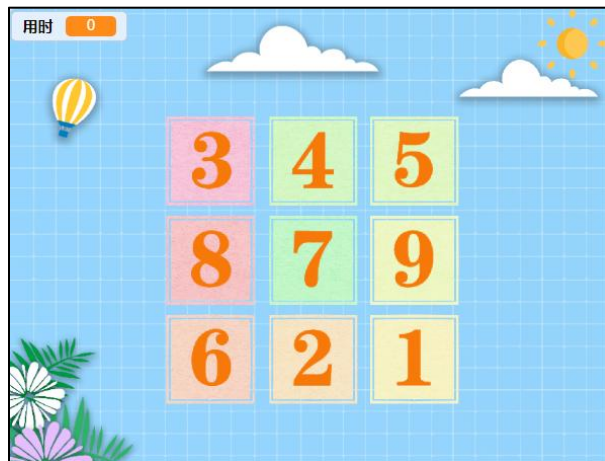
具体要求：

1) 点击绿旗，变量“用时”为 0，背景如图所示；



2) 1 秒后，1、2、3、.....、9 这 9 个数随机出现在九宫格中，如图所示；

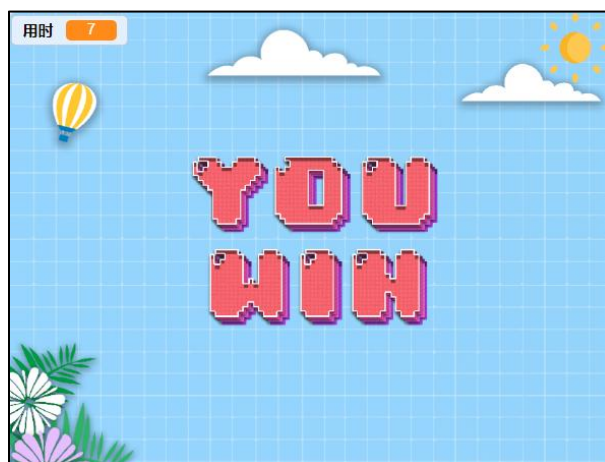
注意：每行的三个数字 y 坐标相同，每列的三个数字 x 坐标相同，行间距和列间距相同。



3) 九宫格里的数字完全呈现后，“用时”开始每隔 1 秒增加 1；

4) 按顺序依次点击 1、2、3、4、.....、9，开始点击 1 时，1 会消失，下一次只有点击 2 时，2 会消失，点击其他数字无任何效果，以此类推；

5) 全部数字都消失后，“用时”不再变化，背景如图所示。





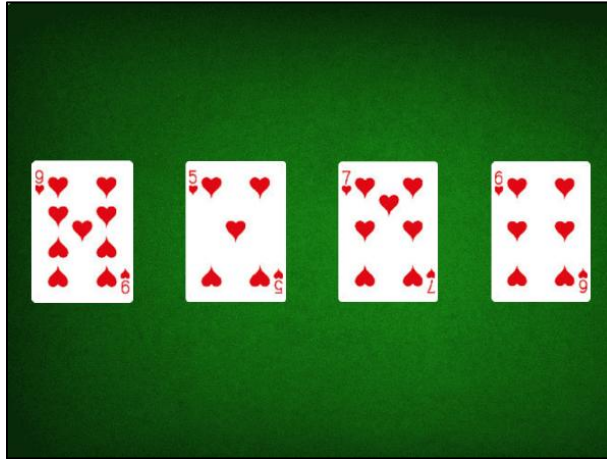
第六题

编程实现：

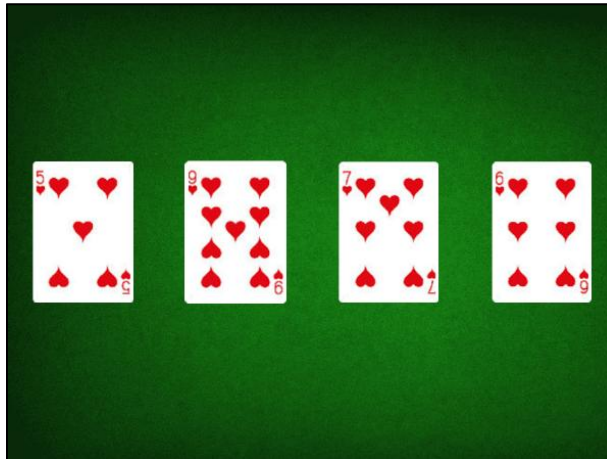
排序。

具体要求：

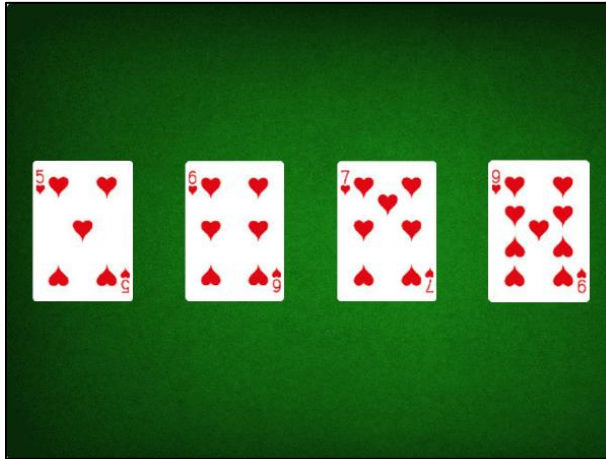
1) 点击绿旗，在舞台上出现 4 张点数不同的扑克牌，牌上的点数是随机的（4~9 点），如图所示；



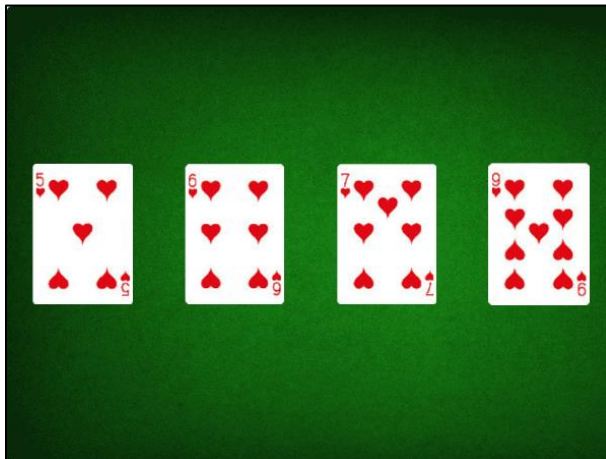
2) 1 秒后，最左侧第 1 张扑克牌与四张牌中最小的牌交换位置（如果第 1 张牌是最小的则不需要交换位置），如图所示；



3) 1 秒后，第 2 张扑克牌与第 2、3、4 三张牌中最小的牌交换位置（如果当前第 2 张牌是三张牌中最小的则不需要交换位置），如图所示；



4) 1 秒后，第 3 张扑克牌与第 3、4 两张牌中较小的牌交换位置（如果当前第 3 张牌是较小的则不需要交换位置），之后排序完成，四张扑克牌从左到右按从小到大的顺序排列，而后程序结束，如图所示。





二、STEMA 考试 Python 中级真题

一、选择题

第一题

下列哪个是赋值运算符？（ ）

- A、= B、!= C、== D、//

第二题

Python 中定义函数的关键字是（ ）。

- A、class B、import C、def D、while

第三题

Python 中可以创建空字典的是（ ）。

- A、dic = () B、dic = {}
C、dic = set() D、dic = []

第四题

已知：s = 'pythonnohtyp'，执行 print(s.index('h') + s.rindex('n')) 语句后，输出的结果是（ ）。

- A、-2 B、-3
C、9 D、11

第五题

已知 a = {1, 2, 3, 4, 5}，b = {2, 3, 4, 5, 6}。下列选项中可以获得集合 {1, 6} 的是（ ）。

- A、(a & b) | (a - b) B、(a | b) | (a - b)
C、(a - b) & (b - a) D、(a | b) - (b & a)

二、编程题

第一题

提示信息：

整数 n 的立方 = $n \times n \times n$ 。



编程实现：

给定一个整数 n ，计算 n 的立方。

例如： $n=2$ ，2 的立方等于 8 ($8=2 \times 2 \times 2$)。

输入描述：输入一个整数 n ($1 \leq n \leq 1000$)

输出描述：输出一个整数，表示 n 的立方

样例输入：2

样例输出：8

第二题

编程实现：

给定三个互不相同的整数，找出其中第二大的数。

例如：三个整数分别为 30、20、25，其中第二大的数是 25。

输入描述：输入共三行，每行一个整数 ($1 \leq \text{整数} \leq 1000$)，三个整数互不相同

输出描述：输出一个整数，表示三个整数中第二大的数

样例输入：

30

20

25

样例输出：

25

第三题

编程实现：

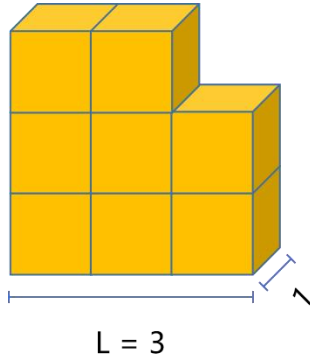
工人需要将 n 个边长为 1 的正方体箱子整齐地摆放在长度为 L ，宽度为 1 的区域。

要求每一层摆满 L 个箱子后，才能往上一层摆放。

请计算 n 个箱子可以放多少层（最上层只要有箱子就算作一层），以及最上层有几个箱子。



例如： $n=8$ ， $L=3$ ，将 8 个箱子摆放在长度为 3，宽度为 1 的区域，可以放 3 层，最上层有 2 个箱子。



输入描述：

第一行输入一个整数 n ($1 \leq n \leq 100$)，表示箱子数量

第二行输入一个整数 L ($1 \leq L \leq 100$)，表示区域的长度

输出描述：

输出两个整数，分别表示 n 个箱子可以放多少层以及最上层有几个箱子，整数之间以一个空格隔开

样例输入：

8

3

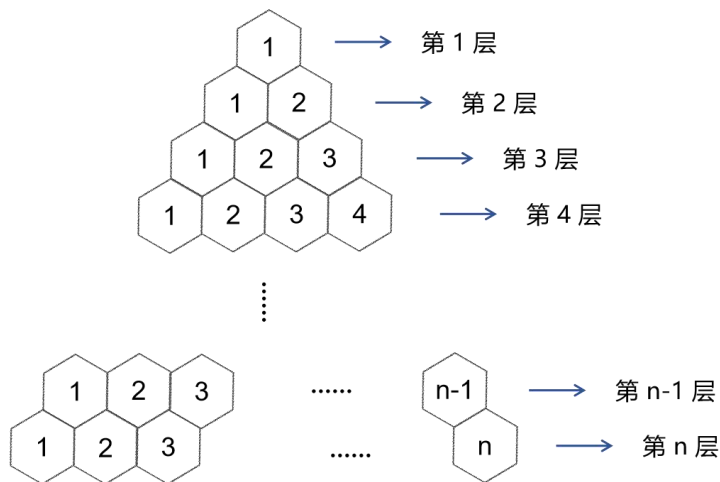
样例输出：

3 2

第四题

提示信息：

数字塔：是由若干个正六边形搭建的一个 n 层的塔，其中每个正六边形中都有一个整数，正六边形和整数的排列规律如下：

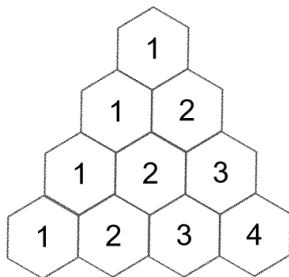


第 n 层的整数从左到右为 1、2、3..... n

编程实现：

给定一个整数 n ，根据提示信息中数字塔的规律，请计算 n 层数字塔中所有整数的和。

例如： $n = 4$ ，4 层的数字塔如下：



所有正六边形中的整数为 1、1、2、1、2、3、1、2、3、4；

所有整数的和为 20。

输入描述： 输入一个整数 n ($1 \leq n \leq 100$)，表示数字塔的层数

输出描述： 输出一个整数，表示 n 层的数字塔中所有整数的和

样例输入： 4

样例输出： 20



第五题

编程实现：

小蓝用仅包含小写字母的字符串表示整数，规则如下：

字符串	表示的整数
a	1
b	2
c	3
⋮	⋮
z	26
aa	27
ab	28
⋮	⋮
az	52
ba	53
bb	54
⋮	⋮
zz	702
aaa	703
aab	704
⋮	⋮

给定一个整数 n ，输出 n 对应的字符串。

例如： $n = 28$ ，28 对应的字符串为 **ab**。

输入描述：输入一个整数 n ($1 \leq n \leq 10^8$)

输出描述：输出一个字符串，表示 n 对应的字符串

样例输入：28

样例输出：ab



第六题

提示信息:

密码锁: 由 n 个从左到右并排的圆环组成, 每个圆环上都有 10 个数字 (0~9), 蓝色框内为密码显示区, 每个圆环在密码显示区只能显示一个数字, 如图所示。可以拨动圆环, 来改变密码显示区显示的数字。

当密码显示区的数字与密码一致时, 密码锁就会被打开。



编程实现:

有一个由 n 个圆环组成的密码锁, 和一个 n 位的密码 S (S 由 1~9 中的数字 (包含 1 和 9) 组成)。每次操作只能选择一个或位置连续的多个圆环拨动。当 S 中的字符从左到右依次显示在密码显示区时, 密码锁会被打开。

已知每个圆环在密码显示区初始数字都为 0, 请计算最少需要操作多少次, 才能打开密码锁。

注意:

1、如果选择了其中一个圆环, 可将该圆环中任意一个数字拨动到密码显示区, 表示 1 次操作;

例如: 将第 3 个圆环拨动到数字 4, 表示 1 次操作:



2、如果选择了位置连续的多个圆环, 只能将这些圆环拨动成同一个数字, 显示在密码显示区, 表示 1 次操作。

例如: 将连续的第 2 个到第 3 个圆环都拨动到数字 5, 表示 1 次操作:

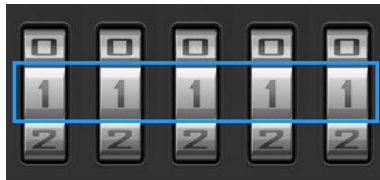




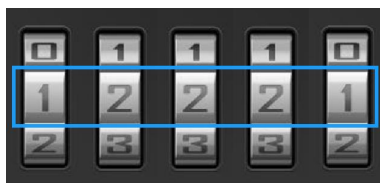
例如：

$n = 5$, $S = "12321"$ ；分别表示 5 个圆环组成的密码锁和密码 12321；将 5 位密码 1、2、3、2、1 从左到右依次显示在密码显示区，以下是操作次数最少的方案：

第一次操作，将 5 个初始状态为 0 的圆环全部拨动到数字 1：



第二次操作，将第 2 个到第 4 个圆环全部拨动到数字 2：



第三次操作，将第 3 个圆环拨动到数字 3：



最少需要操作 3 次，才能打开密码锁。

输入描述：

第一行输入一个整数 n ($1 \leq n \leq 100$)，表示组成的密码锁的圆环数及密码的位数

第二行输入一个长度为 n 的字符串 S ， S 由 1~9 中的数字（包含 1 和 9）组成，表示密码

输出描述：

输出一个整数，表示最少需要操作多少次，才能打开密码锁

样例输入：

5

12321

样例输出：

3



三、STEMA 考试 C++ 中级真题

一、选择题

第一题

表达式 `117 % 16` 的结果是 ()。

- A、0
- B、5
- C、7
- D、10

第二题

下列选项中，字符数组定义正确的是 ()。

- A、`char a[5] = "hello";`
- B、`char a[] = "hello";`
- C、`char a = "hello";`
- D、`char a[6] = 'hello';`

第三题

定义 `int i = 0, a = 10;` 执行表达式 `a = (i <= 0 ? 1 : 2)` 后，`a` 的值是 ()。

- A、1
- B、2
- C、10
- D、随机值

第四题

十进制数 22.5625 转换成十六进制数是 ()。

- A、16.7
- B、16.8
- C、16.9
- D、17.1

第五题

定义 `int a[5] = {1, 2, 3, 4, 5}, *p = a, *q = p++;` 那么 `*p` 和 `*q` 的值分别是 ()。

- A、2、3



B、3、2

C、1、2

D、2、1

二、编程题

第一题

编程实现：第二大的数

题目描述：

给定三个不同的整数，请找出其中第二大的数。

例如：三个整数分别为 30、20、25，其中第二大的数是 25。

输入描述：

一行输入三个不同的整数 a, b, c ($1 \leq a, b, c \leq 1000$)，整数之间以一个空格隔开

输出描述：

输出一个整数，表示三个整数中第二大的数

样例输入：

30 20 25

样例输出：

25

第二题

编程实现：奇妙数

提示信息：

如果一个整数能够被它的各位上的非零数字整除，则称该数为奇妙数。

例如：

整数 102，它的各位上的非零数字为 1 和 2，102 既能被 1 整除，也能被 2 整除，

所以 102 是奇妙数；

整数 456，它的各位上的非零数字为 4、5 和 6，456 能被 4 和 6 整除，但不能被 5



整除，所以 456 不是奇妙数。

题目描述：

给定两个整数 n 和 m ，请输出 n 到 m 之间（包含 n 和 m ）所有的奇妙数；若不存在奇妙数，则输出 -1。

输入描述：

一行输入两个整数 n 和 m ($1 \leq n \leq m \leq 100000$)，整数之间以一个空格隔开

输出描述：

按照从小到大的顺序输出 n 到 m 之间（包含 n 和 m ）所有的奇妙数，整数之间以一个空格隔开；若不存在奇妙数，则输出 -1

样例输入：

10 20

样例输出：

10 11 12 15 20

第三题

编程实现：字符矩阵

题目描述：

给定一个仅包含小写字母的字符串 S ，用这些字符恰好排成一个 n 行 m 列的矩阵 ($m \geq n$)，请找出所有符合要求的矩阵中最接近于正方形的那个矩阵。然后从第一列开始，逐列按照从上到下的顺序输出矩阵中的字符。

例如：

$S = \text{"abcdefgh"}$ 。按要求 $m \geq n$ 有如下两种矩阵：

第一种：

a	b	c	d
e	f	g	h

第二种：

a	b	c	d	e	f	g	h
---	---	---	---	---	---	---	---

最接近正方形的矩阵是第一种。从第一列开始，逐列按照从上到下的顺序输出矩阵中的字符，结果为：aebfcgdh。



输入描述:

输入一个字符串，仅包含小写字母且长度不超过 200

输出描述:

请找出符合题目要求的最接近正方形的字符矩阵，从第一列开始，逐列按照从上到下的顺序输出矩阵中的字符

样例输入:

abcdefgh

样例输出:

aebfcgdh

第四题

编程实现: 走楼梯

题目描述:

一段楼梯共有 n 阶，小明每次最少走 1 阶，最多走 k 阶，请问小明共有多少种不同的走法可以走完这 n 阶楼梯。

例如: $n = 4$, $k = 2$; 楼梯共有 4 阶，小明每次最多走 2 阶;

有如下走法:

第一种: 第一次走 1 阶，第二次走 1 阶，第三次走 1 阶，第四次走 1 阶;

第二种: 第一次走 1 阶，第二次走 1 阶，第三次走 2 阶;

第三种: 第一次走 1 阶，第二次走 2 阶，第三次走 1 阶;

第四种: 第一次走 2 阶，第二次走 1 阶，第三次走 1 阶;

第五种: 第一次走 2 阶，第二次走 2 阶。

所以小明共有 5 种不同的走法可以走完 4 阶楼梯。

输入描述:

一行输入两个整数 n ($1 \leq n \leq 5000$) 和 k ($1 \leq k \leq 10$), 分别表示这段楼梯的阶数及每次最多可以走的楼梯阶数，整数之间以一个空格隔开

输出描述:

输出一个整数，表示小明走完 n 阶楼梯共有多少种不同的走法

样例输入:

4 2



样例输出：

5

第五题

编程实现：均分糖果

题目描述：

有 n 个小朋友站成一排，起初，从左到右每个小朋友分别有 $a_1, a_2, a_3, \dots, a_n$ 颗糖果。

接下来每一轮老师都会选择一个或多个有糖果的小朋友，他们会将自己的一颗糖果，送给一个与他相邻的小朋友（左边或者右边的小朋友）。

请计算最少需要经过多少轮，才能使所有小朋友的糖果数量都相等？如果做不到，则输出 -1。

例 1：

$n = 3$ ，表示有三个小朋友，他们起初的糖果数量分别为 1、0、5；

以下是一种最少轮次的情况：

第一轮，选择第三个小朋友，将 1 颗糖送给第二个小朋友，之后三个小朋友的糖果数量依次为 1、1、4：

1 0 <-- 5 => 1 1 4

第二轮，选择第二个和第三个小朋友，第二个小朋友将 1 颗糖送给第一个小朋友，第三个小朋友将 1 颗糖给第二个小朋友，之后三个小朋友的糖果数量依次为 2、1、3：

1 <-- 1 <-- 4 => 2 1 3

第三轮，选择第三个小朋友，将 1 颗糖送给第二个小朋友，之后三个小朋友的糖果数量依次为 2、2、2：

2 1 <-- 3 => 2 2 2

最少需要经过 3 轮，才能使三个小朋友的糖果数量都相等。

例 2：

$n = 3$ ，表示有三个小朋友，他们起初的糖果数量分别为 0、2、0；

无论如何操作，都不可能使三个小朋友的糖果数量都相等，故输出 -1。

输入描述：

第一行输入一个整数 n ($1 \leq n \leq 10^4$)，表示小朋友的人数

第二行输入 n 个整数 $a_1, a_2, a_3, \dots, a_n$ ($0 \leq a_i \leq 10^5$)，分别表示从左到右每个



小朋友起初拥有的糖果数量，整数之间以一个空格隔开

输出描述:

输出一个整数，表示至少需要经过多少轮，才能使所有小朋友的糖果数量都相等；如果做不到，则输出 -1

样例输入:

3

1 0 5

样例输出:

3

第六题

编程实现: 密码锁

提示信息:

密码锁：由 n 个从左到右并排的圆环组成，每个圆环上都有 10 个数字（0~9），蓝色框内为密码显示区，每个圆环在密码显示区只能显示一个数字，如图所示。可以拨动圆环，来改变密码显示区显示的数字。

当密码显示区的数字与密码一致时，密码锁就会被打开。



题目描述:

有一个由 n 个圆环组成的密码锁，和一个 n 位的密码 S (S 由 1~9 中的数字（包含 1 和 9）组成）。每次操作只能选择一个或位置连续的多个圆环拨动，当 S 中的字符从左到右依次显示在密码显示区时，密码锁会被打开。

已知每个圆环在密码显示区初始数字都为 0，请计算出最少需要操作多少次，才能打开密码锁。

注意:

1、如果选择了其中一个圆环，可将该圆环中任意一个数字拨动到密码显示区，表示 1 次操作；

例如：将第 3 个圆环拨动到数字 4，表示 1 次操作：



2、如果选择了位置连续的多个圆环，只能将这些圆环拨动成同一个数字，显示在密码显示区，表示 1 次操作。

例如：将连续的第 2 个到第 3 个圆环都拨动到数字 5，表示 1 次操作：



例如：

$n = 5$, $S = "12321"$ ；分别表示 5 个圆环组成的密码锁和密码 12321；将 5 位密码 1、2、3、2、1 从左到右依次显示在密码显示区，以下是操作最少次数的方案：

第一次操作，将 5 个初始状态为 0 的圆环全部拨动到数字 1：



第二次操作，将第 2 个到第 4 个圆环全部拨动到数字 2：



第三次操作，将第 3 个圆环拨动到数字 3：



最少需要操作 3 次，才能打开密码锁。

输入描述：

第一行输入一个整数 n ($1 \leq n \leq 100$)，表示组成的密码锁的圆环数及密码的位数

第二行输入一个长度为 n 的字符串 s ， s 由 1~9 中的数字(包含 1 和 9)组成，表示密码



输出描述:

输出一个整数，表示最少需要操作多少次，才能打开密码锁

样例输入:

5

12321

样例输出:

3



四、STEMA 考试 Arduino 中级真题

一、选择题

第一题

测量湿度，应使用（ ）。

- A、声音传感器
- B、光强传感器
- C、温湿度传感器
- D、超声波传感器

第二题

9g 舵机可以从 0 度最多转到（ ）度。

- A、60
- B、90
- C、150
- D、180

第三题

运行以下程序，串口打印（ ）。



- A、4
- B、5
- C、6
- D、7

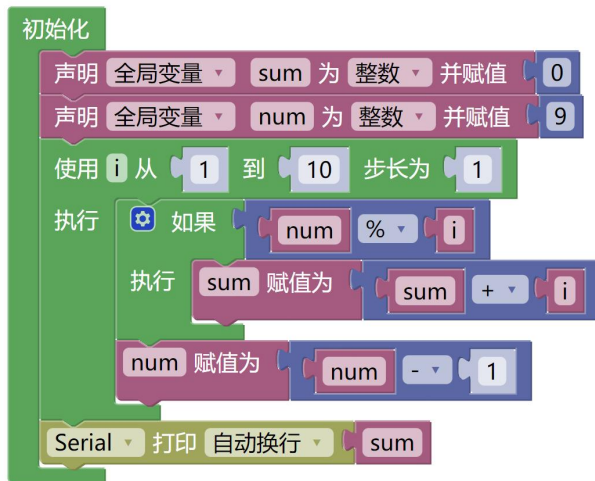
第四题

Arduino Mega 2560 中，声音传感器使用 A3 管脚检测，检测的精度是二进制的（ ）位。

- A、4
- B、6
- C、8
- D、10

第五题

运行以下程序，串口打印（ ）。



A、37

B、40

C、45

D、50

二、编程题

第一题

硬件准备:

温湿度传感器 DHT11 (P42)

编程实现:

温度与湿度。

具体要求:

- 1) 程序开启后，串口无打印；
- 2) 1s 后，串口打印当前温度；
- 3) 接着 1s 后，串口打印当前湿度；
- 4) 重复 2)，3) 过程。

第二题

硬件准备:

全彩 LED (R: P5、G: P6、B: P7)

编程实现:



红、绿、蓝小灯。

具体要求：

- 1) 程序开启后，全彩 LED 呈熄灭状态；
- 2) 1s 后，全彩 LED 中只有红色小灯亮起；
- 3) 红色小灯亮起 1s 后，全彩 LED 中只有绿色小灯亮起；
- 4) 绿色小灯亮起 1s 后，全彩 LED 中只有蓝色小灯亮起；
- 5) 重复 2)，3)，4) 过程。

第三题

硬件准备：

光强传感器 (A4)、全彩 LED (R: P5、G: P6、B: P7)

注：将 P5、P6、P7 均设为低电平时，全彩 LED 会变为白色

编程实现：

光敏小灯。

小灯会根据光线的强弱自动开启或关闭。

具体要求：

- 1) 程序开启后，全彩 LED 呈熄灭状态；
- 2) 当光敏传感器检测数值大于等于 800 时，全彩 LED 点亮并变为白色；
- 3) 当光敏传感器检测数值小于 800 时，全彩 LED 熄灭；
- 4) 重复 2)，3) 过程。

第四题

硬件准备：

四位红色数码管 TM1650 (0x27)、按键 S1 (P22)

编程实现：

倒计时。

使用按键 S1 完成倒计时的开始、暂停功能。

具体要求：

- 1) 程序开启后，四位红色数码管显示 99 并静止；
- 2) 静止状态下，按下一次按键 S1，四位红色数码管以 1 秒的间隔倒计时；



例如：99 在 1s 秒后变为 98；98 在 1s 后变为 97；97 在 1s 后变为 96.....以此类推。

3) 倒计时过程中，按下一次按键 S1，当前计时变为静止状态（即停止倒计时）；

4) 重复 2)，3) 过程。

第五题

硬件准备：

8 路 LED（P31-P38）、超声波传感器（Echo：A0、Trig：A1）

编程实现：

测距灯。

根据超声波传感器测量的距离点亮 P31-P38 的小灯。

具体要求：

- 1) 程序开启后，2s 内 P31-P38 的小灯熄灭，超声波传感器不进行检测；
- 2) 之后每隔 1s，超声波传感器检测一次距离并在串口中打印；
- 3) 根据检测距离所处的范围，点亮以下小灯：

超声波传感器检测距离 x	点亮小灯
$x \leq 10$	P31
$10 < x \leq 20$	P31、P32
$20 < x \leq 30$	P31、P32、P33
$30 < x \leq 40$	P31、P32、P33、P34
$40 < x \leq 50$	P31、P32、P33、P34、P35
$50 < x \leq 60$	P31、P32、P33、P34、P35、P36
$60 < x \leq 70$	P31、P32、P33、P34、P35、P36、P37
$70 < x$	P31、P32、P33、P34、P35、P36、P37、P38

4) 重复 2)，3) 过程。

第六题

硬件准备：

LCD1602（0x20）、按键 S1（P22）

编程实现：

消失的数字。

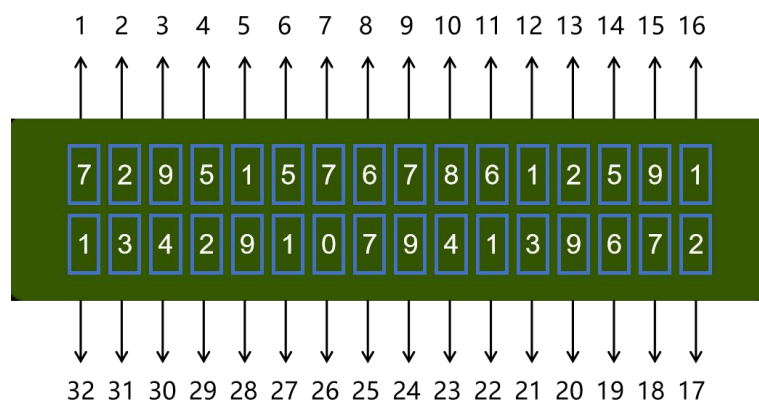
具体要求：



1) 程序开启后，LCD1602 无显示；

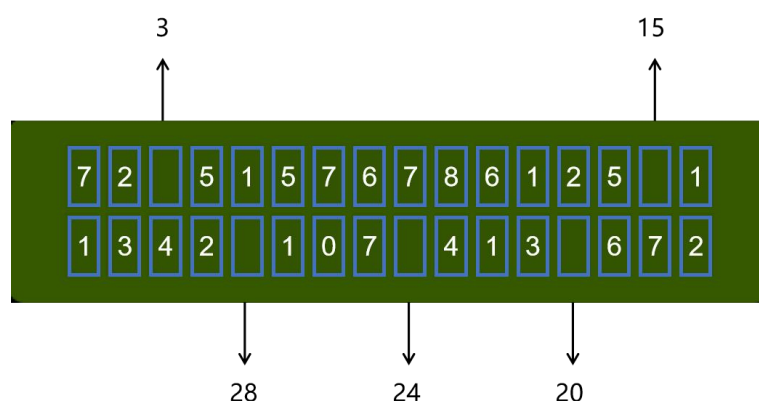
2) 按下按键 S1, 2s 后，LCD1602 的 1~32 号位置，每个位置都出现随机的数字（数字范围 0~9，包含 0 和 9）；

例如：LCD1602 各位置用蓝色矩形框标出并用黑色数字编号，蓝色矩形框内为可能出现的随机数字，如下图所示：



3) 2s 后，1~32 号位置中，当前最大的数字都会消失；

例如：当前最大数字是 9，第 3、15、20、24、28 位置的 9 消失。



4) 之后每隔 2s，剩余的数字中，最大的数字都会消失，重复这个过程直至所有的数字消失；

5) 重复 2)，3)，4) 过程。



五、STEMA 考试 EV3 中级真题

一、选择题

第一题

颜色传感器应该连接 EV3 主机的哪个端口？（ ）

- A. 输出端口
- B. 输入端口
- C. SD 卡端口
- D. PC 端口

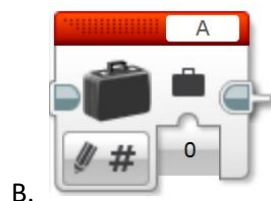
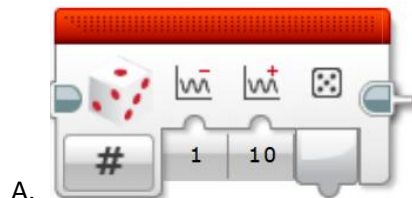
第二题

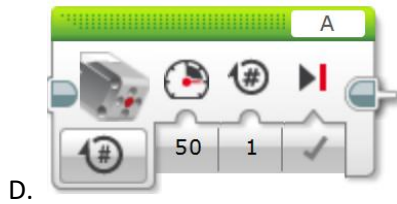
以下哪个传感器可以测量物体间的距离？（ ）

- A. 触动传感器
- B. 颜色传感器
- C. 超声波传感器
- D. 陀螺仪传感器

第三题

以下哪个程序块不在数据操作模块中？（ ）



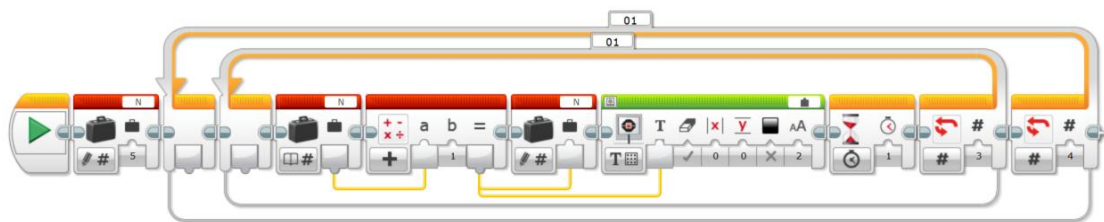


第四题

二进制数 1010 转换成十进制数为 ()。

- A. 3
- B. 6
- C. 10
- D. 32

第五题



运行以上程序后，N-5 的值是 ()。

- A. 8
- B. 9
- C. 6
- D. 17

二、编程题

第一题

硬件准备:

将一个大马达连接到 EV3 主机的 A 口。

编程实现:

大马达转动。

具体要求:

- 1) 运行程序，大马达静止;



- 2) 等待 1 秒，大马达以 10 的功率转动 2 圈；
- 3) 等待 1 秒，大马达以 80 的功率转动 2 圈。

第二题

硬件准备：

EV3 主机。

编程实现：

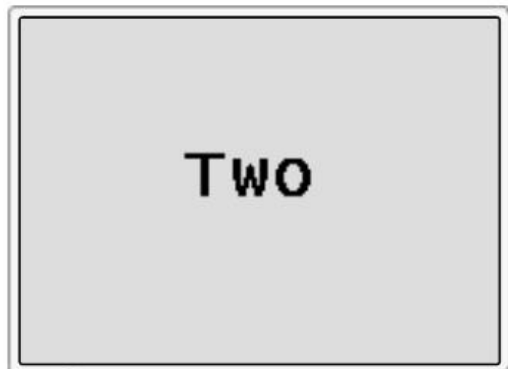
计时数字。

具体要求：

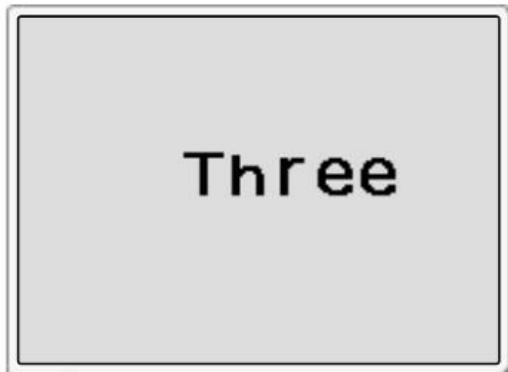
- 1) 运行程序，EV3 主机屏幕中间区域显示 One；



- 2) 等待 1 秒，EV3 主机屏幕中间区域显示 Two；



- 3) 等待 1 秒，EV3 主机屏幕中间区域显示 Three；



4) 等待 1 秒，程序结束。

第三题

硬件准备:

EV3 主机。

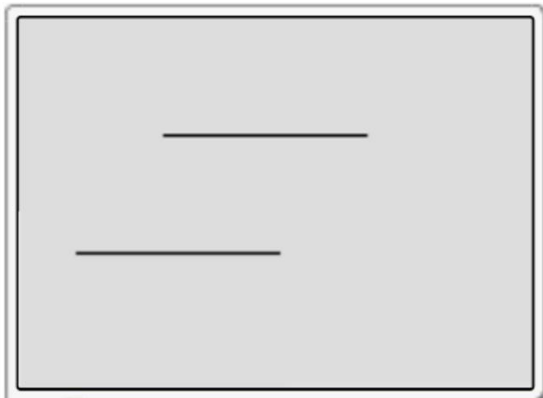
编程实现:

绘制平行四边形

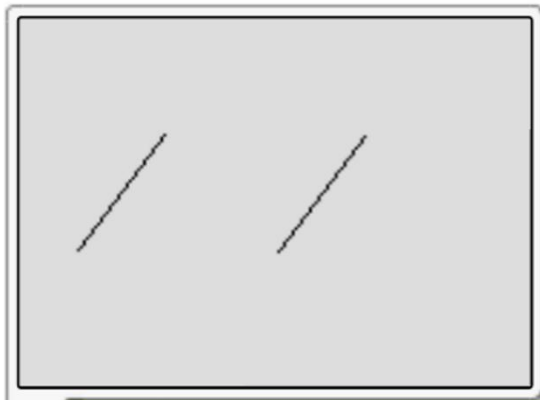
平行四边形的四个顶点坐标分别是 (50, 40)、(120, 40)、(20, 80)、(90, 80)。

具体要求:

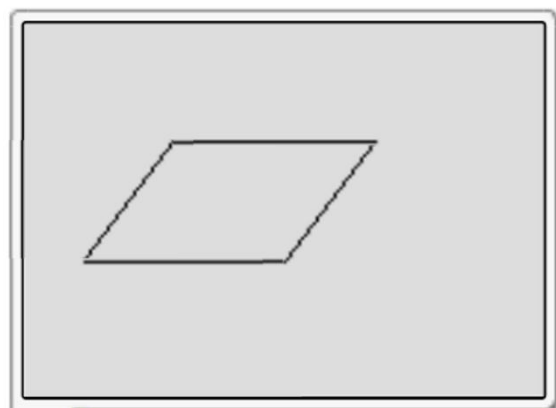
1) 运行程序，屏幕显示平行四边形的上、下两边；



2) 按下程序块左按键显示平行四边形的左、右两边；



3) 按下程序块右按键显示平行四边形;



4) 按程序块中间按键清屏，重复 2)、3) 过程。

第四题

硬件准备:

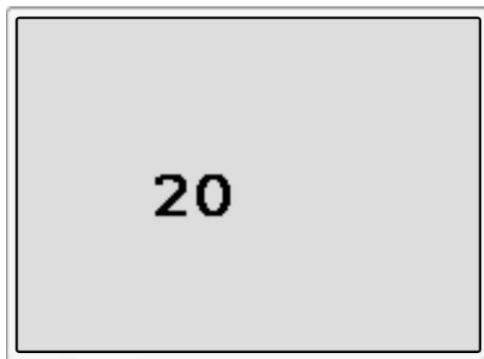
将一个大马达连接到 EV3 主机的 A 口，将两个触动传感器连接到 EV3 主机的 1 口和 2 口。

编程实现:

用触动传感器来控制大马达的转动功率。

具体要求:

- 1) 运行程序，大马达会以 20 的功率持续转动;
- 2) 在屏幕 (50, 60) 的位置实时显示大马达转动功率的数值 (图中 20 代表此时大马达的转动功率);



- 3) 按一次 1 口触动传感器，大马达的转动功率增加 5，屏幕上的数值增加 5；
- 4) 按一次 2 口触动传感器，大马达的转动功率减少 5，屏幕上的数值减少 5；
- 5) 大马达转动功率的范围[20~40]（包含 20 和 40）。

第五题

硬件准备：

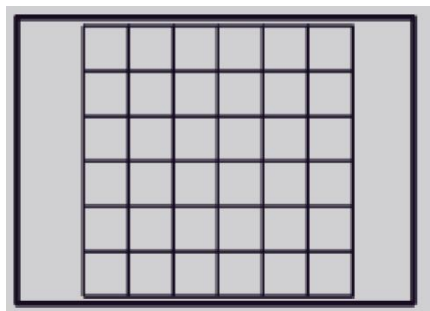
EV3 主机。

编程实现：

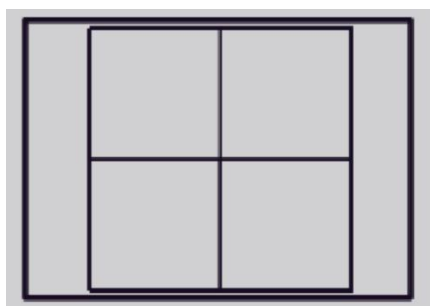
屏幕显示矩阵。

具体要求：

- 1) 运行程序，屏幕清屏；
- 2) 按下中间键，屏幕显示 6*6 的正方形网格；

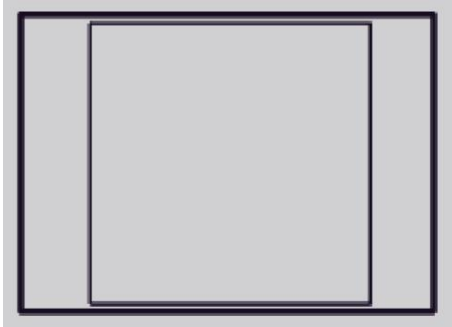


- 3) 再次按下中间键，屏幕显示 2*2 的正方形网格；





4) 最后按下中间键，屏幕显示 1 个正方形，保持 2 秒程序结束。



第六题

硬件准备:

EV3 主机。

编程实现:

生成随机方阵（方阵由填充圆组成），然后由里到外以逆时针漩涡顺序依次变为空心圆。

以下图片示例仅供参考

具体要求:

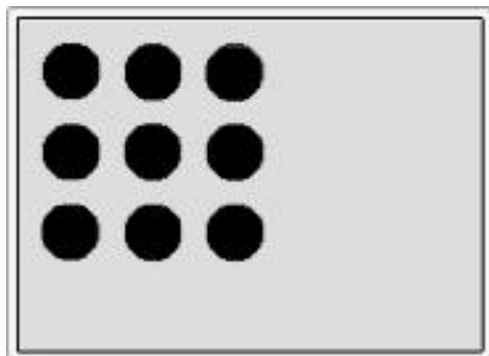
1) 运行程序，EV3 主机屏幕中间位置显示 Matrix;



2) 按一次 EV3 主机中间按键，EV3 主机屏幕清屏，并随机显示 3*3、4*4、5*5 这三种方阵中的一种；

方阵由填充圆组成，每个填充圆的半径为 10，填充圆之间的间距为 10

如图所示，以 3*3 方阵为例



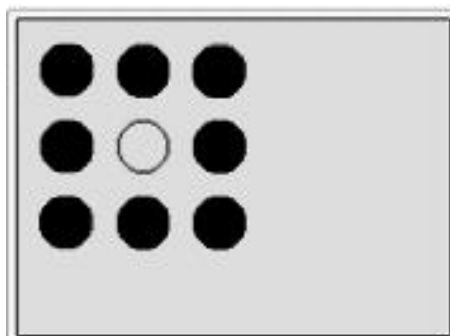
3) 0.2 秒后，第二行第 2 个填充圆变为空心圆；

3*3、4*4、5*5 这三种方阵，初始填充圆变成空心圆的位置如下

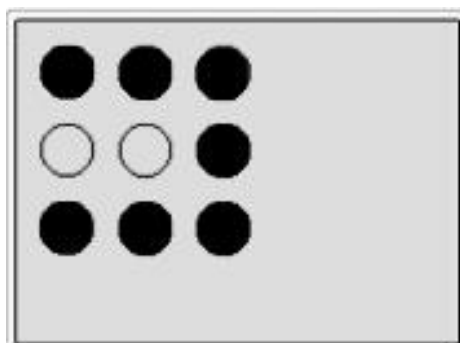
3*3 方阵：从第二行第 2 个填充圆变为空心圆开始

4*4 方阵：从第二行第 3 个填充圆变为空心圆开始

5*5 方阵：从第三行第 3 个填充圆变为空心圆开始

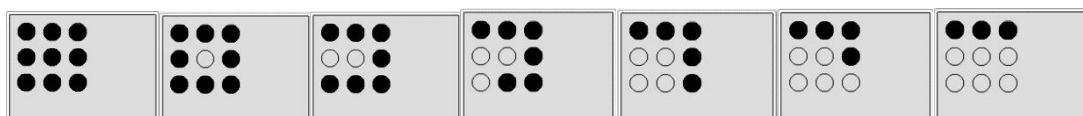


4) 0.2 秒后，第二行第 1 个填充圆变为空心圆；



5) 以此类推，填充圆都以 0.2 秒的间隔，按照逆时针漩涡顺序依次变为空心圆，直到方阵中再无填充圆；

具体路线如图所示：



6) 重复上面的 2), 3), 4), 5) 过程。