

GESP 图形化四级试卷

(满分：100 分 考试时间：90 分钟)

学校：_____

姓名：_____

题目	一	二	三	总分
得分				

一、单选题 (每题 2 分, 共 30 分)

题号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
答案	D	C	D	B	D	A	C	B	C	A	C	D	C	B	A

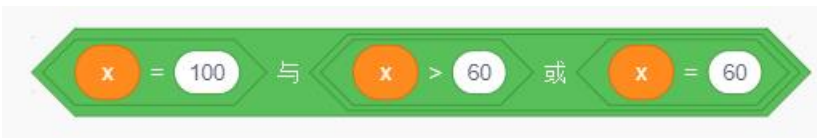
1、高级语言编写的程序需要经过以下（ ）操作，可以生成在计算机上运行的可执行代码。

- A. 编辑
- B. 保存
- C. 调试
- D. 编译

2、排序算法是稳定的 (Stable Sorting)，就是指排序算法可以保证，在待排序数据中有两个相等记录的关键字 R 和 S (R 出现在 S 之前)，在排序后的列表中 R 也一定在 S 前。下面关于排序稳定性的描述，正确的是（ ）。

- A. 冒泡排序是不稳定的
- B. 插入排序是不稳定的
- C. 选择排序是不稳定的
- D. 以上都不正确

3、下列哪个选项可以判断 x 不等于 100 并且大于等于 60？（ ）

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

4、默认小猫角色，输入一个三位数，希望程序输出这个三位数各个数位上的数字之和，下列代码中的“①”是？（ ）



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

5、全班同学的成绩已经存储在列表“lst”中，小于 60 分为不及格，默认小猫角色，执行下列哪个程序可以让小猫说出不及格的人数？（ ）

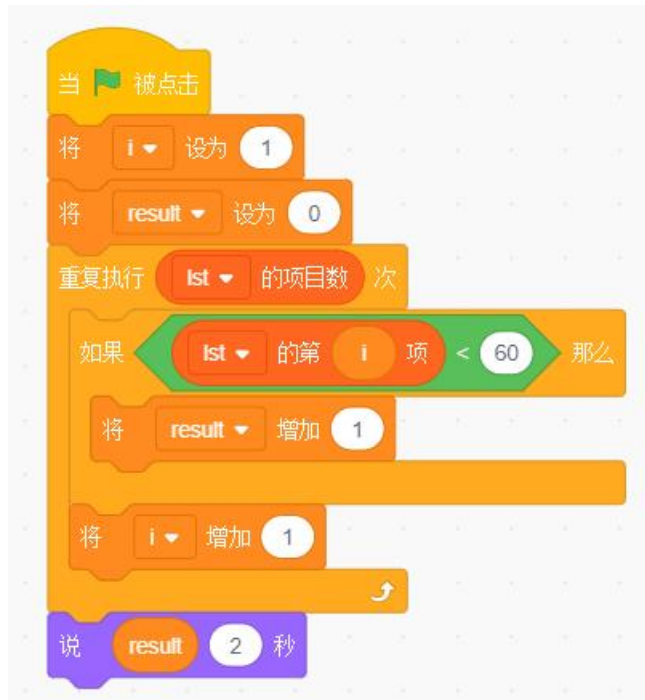
- A. 



B.



C.

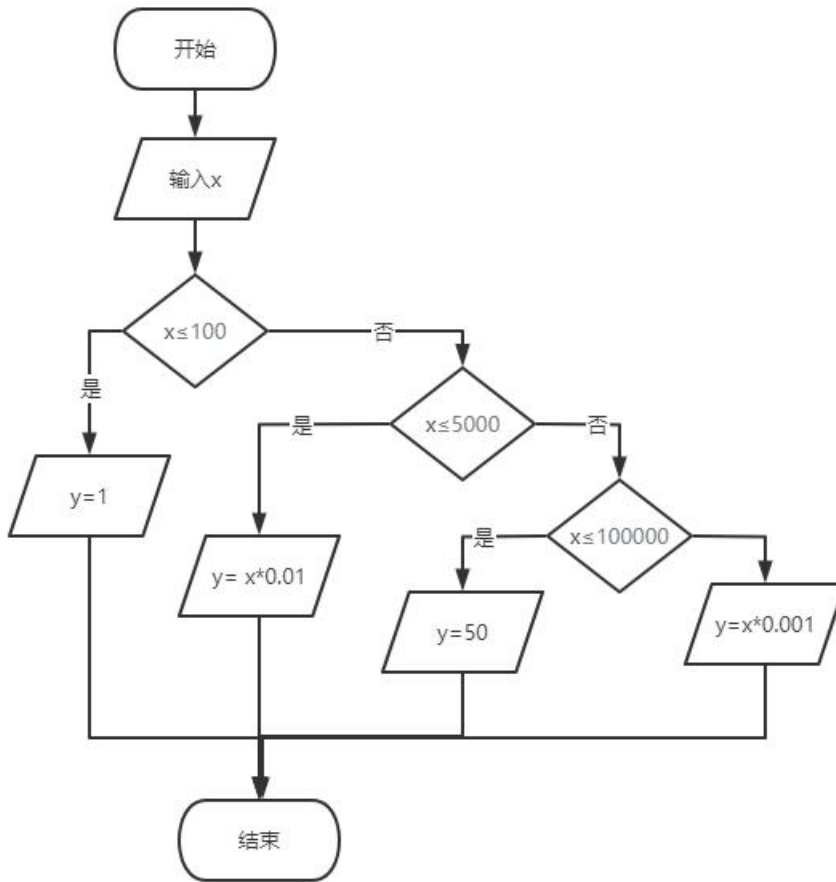


D.

6、对存有 8 个元素的列表 [49, 38, 65, 97, 76, 13, 27, 50] 按从小到大的顺序进行排序，选择排序第一趟的排序结果是？（ ）

- A. [13, 38, 65, 97, 76, 49, 27, 50]
- B. [13, 27, 38, 49, 50, 65, 76, 97]
- C. [97, 76, 65, 50, 49, 38, 27, 13]
- D. [13, 38, 65, 50, 76, 49, 27, 97]

7、当输入 x 为 5500 时，根据以下流程图 y 的值为？（ ）



- A. 1
- B. 55
- C. 50
- D. 5.5

8、默认小猫角色，自定义一个判断质数的积木，其中“①”处应填入的条件是？（ ）



A.



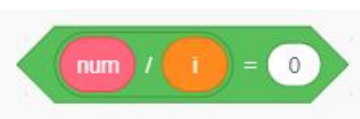
B.



C.



D.

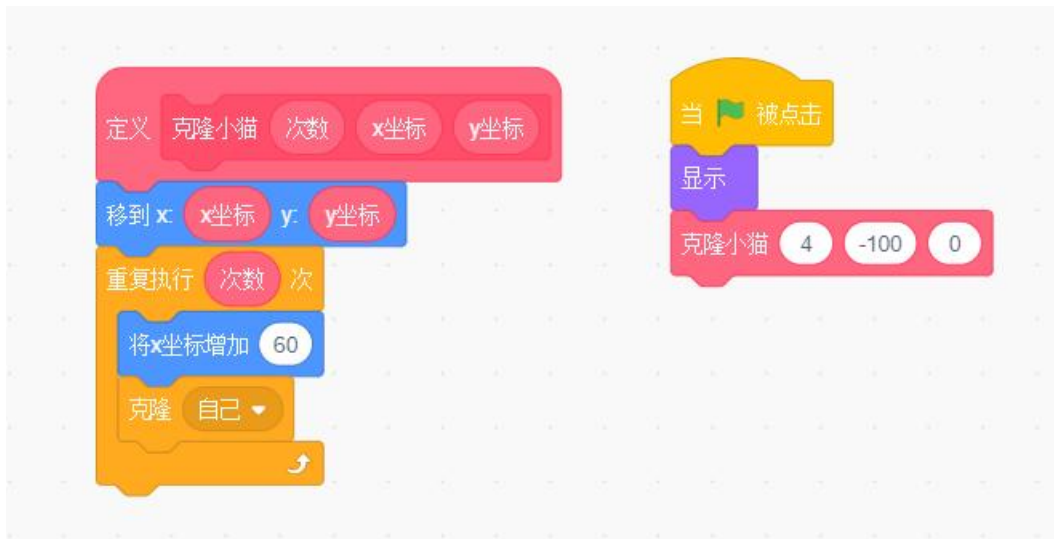


9、默认小猫角色，执行下列程序，变量 result 的值是？（ ）



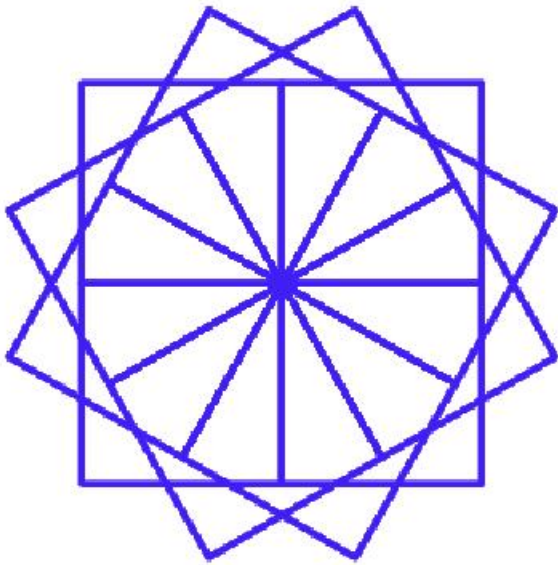
- A. 4
- B. 6
- C. 7
- D. 9

10、默认小猫角色，执行下列程序，舞台上会出现几只小猫？（ ）



- A. 4
- B. 5
- C. 3
- D. 6

11、默认小猫角色，若想绘制如下图形，下列代码中的“①、②”是？（ ）



- A. 9、40
- B. 10、36
- C. 12、30
- D. 13、32

12、默认小猫角色，若想实现颠倒字符串，比如输入“apple”，小猫会说“elppa”，下列代码中的“①”是？（ ）



A.

连接 str 和 回答

B.

连接 回答 和 str

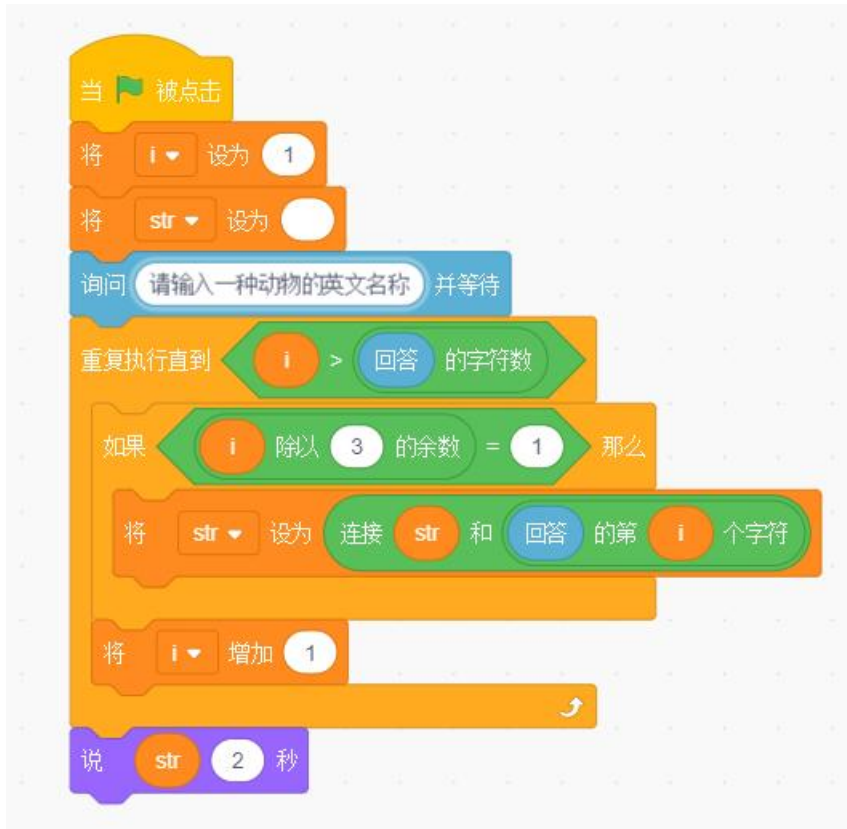
C.

连接 str 和 回答 的第 i 个字符

D.

连接 回答 的第 i 个字符 和 str

13、默认小猫角色，执行下列程序，输入蝴蝶的英文单词“butterfly”，输出结果是（ ）？



- A. uel
- B. try
- C. btf
- D. tf

14、默认小猫角色，执行下列程序，输出的结果是？（ ）

水果清单	
1	苹果
2	香蕉
3	橙子
4	葡萄
5	草莓
6	芒果
7	柚子

+ 长度7 =



```

当 被点击
  将 猕猴桃 加入 水果清单
  在 水果清单 的第 4 项前插入 火龙果
  如果 水果清单 包含 草莓 ? 那么
    删除 水果清单 的第 5 项
  否则
    将 梨 加入 水果清单
  说 水果清单 的项目数 2 秒
        
```

- A. 7
- B. 8
- C. 9
- D. 10

15、默认小猫角色，执行下列程序，输出的结果是？（ ）



- A. 笔记本
- B. 笔袋
- C. 铅笔
- D. 彩笔

二、判断题（每题 2 分，共 20 分）

题号	1	2	3	4	5	6	7	8	9	10
答案	×	√	√	×	√	×	√	√	×	√

1、域名是由一串用点分隔的名字来标识互联网上一个计算机或计算机组的名称,CCF 编程能力等级认证官方网站的域名是 `gesp.ccf.org.cn`，其中顶级域名是 `gesp`。

（ ）

2、枚举算法是一种基于穷举搜索的算法，可以通过遍历所有可能的解来找到最终结果。（ ）

3、默认小猫角色，执行“摇号”程序，小猫说“选中你啦”的概率为 20%。（ ）



4、广播与变量都能起到在角色间传递信息的作用，广播与变量一样也可以设定作用范围为“适用于所有角色”或“仅适用于当前角色”。（ ）

5、默认小猫角色，执行下列代码，输入“王大鹏”后，角色会说“欢迎光临，王先生”。（ ）



6、默认小猫角色，执行下列代码，列表 lst 中从上到下分别为 1, 3, 5, 7, 9, 11。

()



7、默认小猫角色，下列两段代码功能相同。()



脚本一



脚本二

8、默认小猫角色，下面两段代码都能实现按下 a 键或者按下 b 键使角色旋转。（ ）



程序1



程序2

9、利用枚举算法可以从 1 到 100 中筛选出即能被 5 整除又能被 7 整除的整数，筛选出的整数个数为 3。（ ）

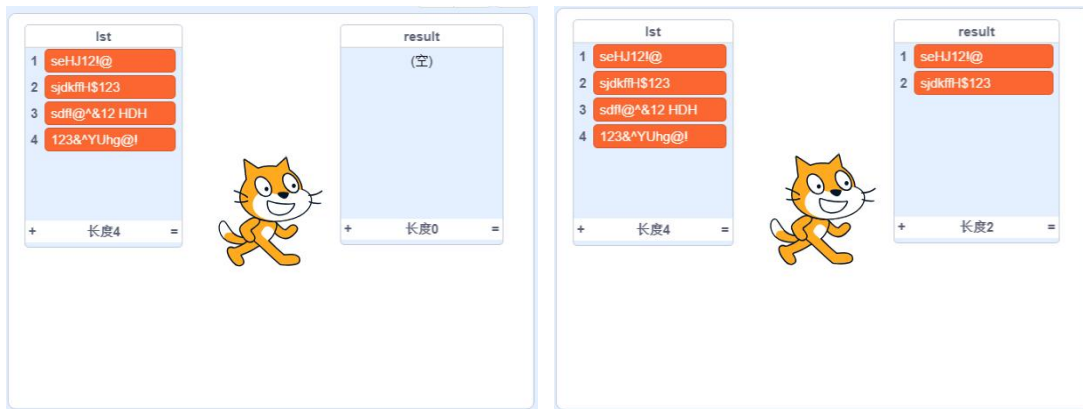
10、斐波那契数列指的是这样的—个数列 1, 1, 2, 3, 5, 8, 13, 21, 34, ... 这个数列从第 3 项开始，每一项都等于前两项之和。默认小猫角色，执行下列代码，可使小猫每隔 1 秒依次说出斐波那契数列的每一项。（ ）



三、编程题（每题 25 分，共 50 分）

题号	1	2
答案		

1、密码合规检测



【题目描述】

网站注册需要有用户名和密码，默认小猫角色和白色背景，编写程序以检查用户输入密码的有效性。

- （1）合法的密码只能由 a-z 之间 26 个字母（字母不区分大小写）、0-9 之间 10 个数字以及!@#\$四个特殊字母构成。
- （2）密码最短长度：6 个字符，密码最大长度：12 个字符。
- （3）密码必须包含字母、数字和特殊字符，缺一不可。

【输入描述】

新建变量“lst”，用于存储待检测的密码。

如下图所示：

lst	
1	seHJ12!@
2	sjdkffH\$123
3	sdf!@^&12 HDH
4	123&^YUhg@!

+

长度4

=

【输出描述】

新建列表“result”，用于存储合规的密码，存入顺序以“lst”列表中字符串的先后为序。

如下图所示：

result	
1	seHJ12!@
2	sjdkffH\$123

+

长度2

=

【输入样例】

```
lst = [seHJ12!@, sjdkffH$123, sdf!@^&12 HDH, 123&^YUhg@!]
```

【输出样例】

```
result = [seHJ12!@, sjdkffH$123]
```

【输入样例】

```
lst = [123!@#%^ZAqaz, abc123ABC!@#, qaz123WSX!@#, 12QAqa!@, 12QAqa!@^&]
```

【输出样例】

```
result = [abc123ABC!@#, qaz123WSX!@#, 12QAqa!@]
```

【评分标准】

一共 5 组测试用例，通过一组测试用例得 5 分。

测试用例 1:

```
lst = [abcq313QAX!@, qaz123!@#QAZ, 123!!XYZxyz@@, 123!@#&^%ZAqaz, 12QAqa!@^&];  
result = [abcq313QAX!@, qaz123!@#QAZ]
```

测试用例 2:

```
lst = [abcq313QAX!@, helpCCF9!@#$, try!#!#TRY147];  
result = [abcq313QAX!@, helpCCF9!@#$]
```

测试用例 3:

```
lst = [147WRwr!@#, qaz123!@#QAZ, 123XYZxyz@@, !@#$$12345qQ];  
result = [147WRwr!@#, qaz123!@#QAZ, 123XYZxyz@@]
```

测试用例 4:

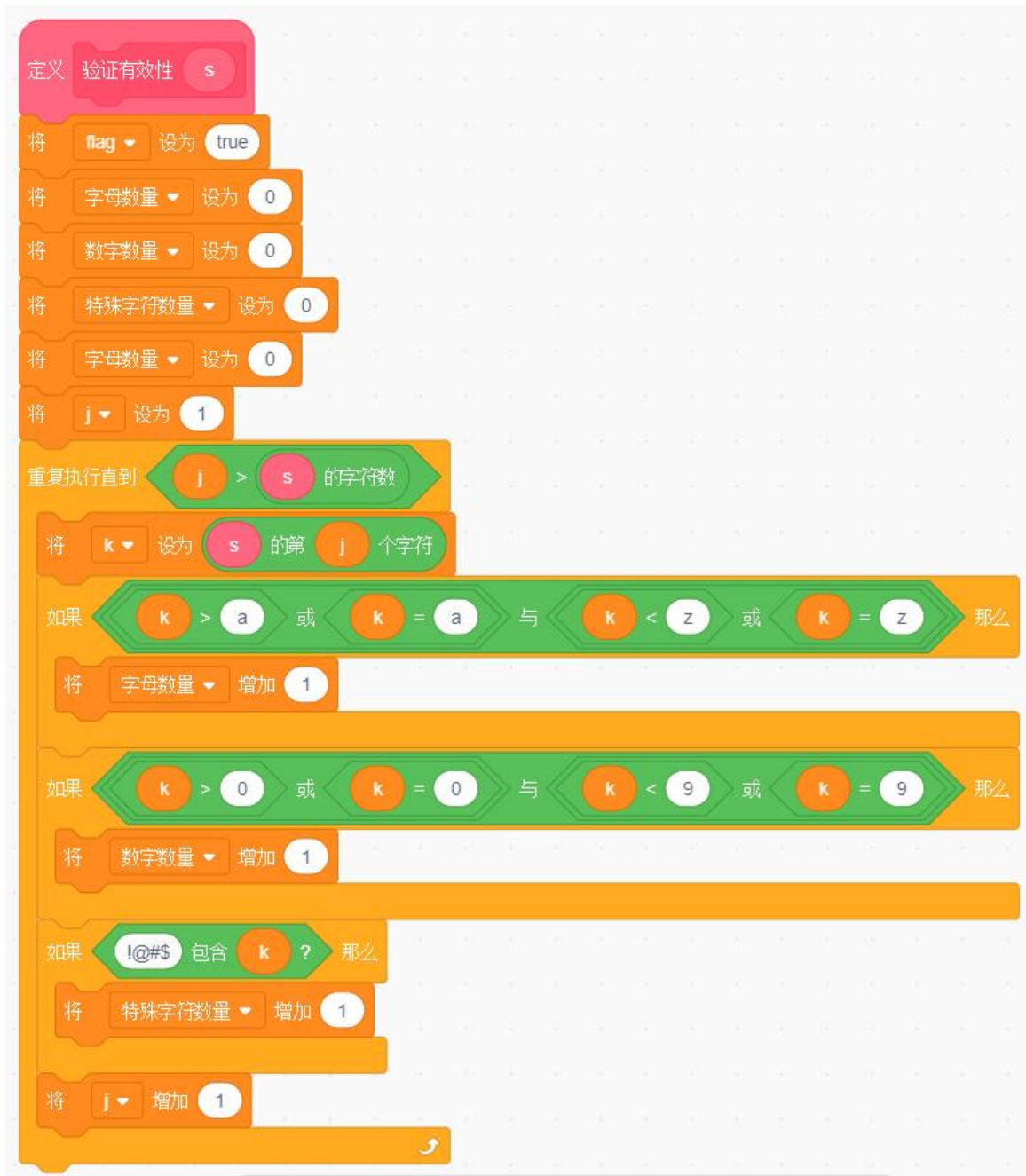
```
lst = [258DHZdhz#@!, qaz&^123!@#QAZ, 123!!XYZxyz@@, !@#$$12345qQ];  
result = [258DHZdhz#@!]
```

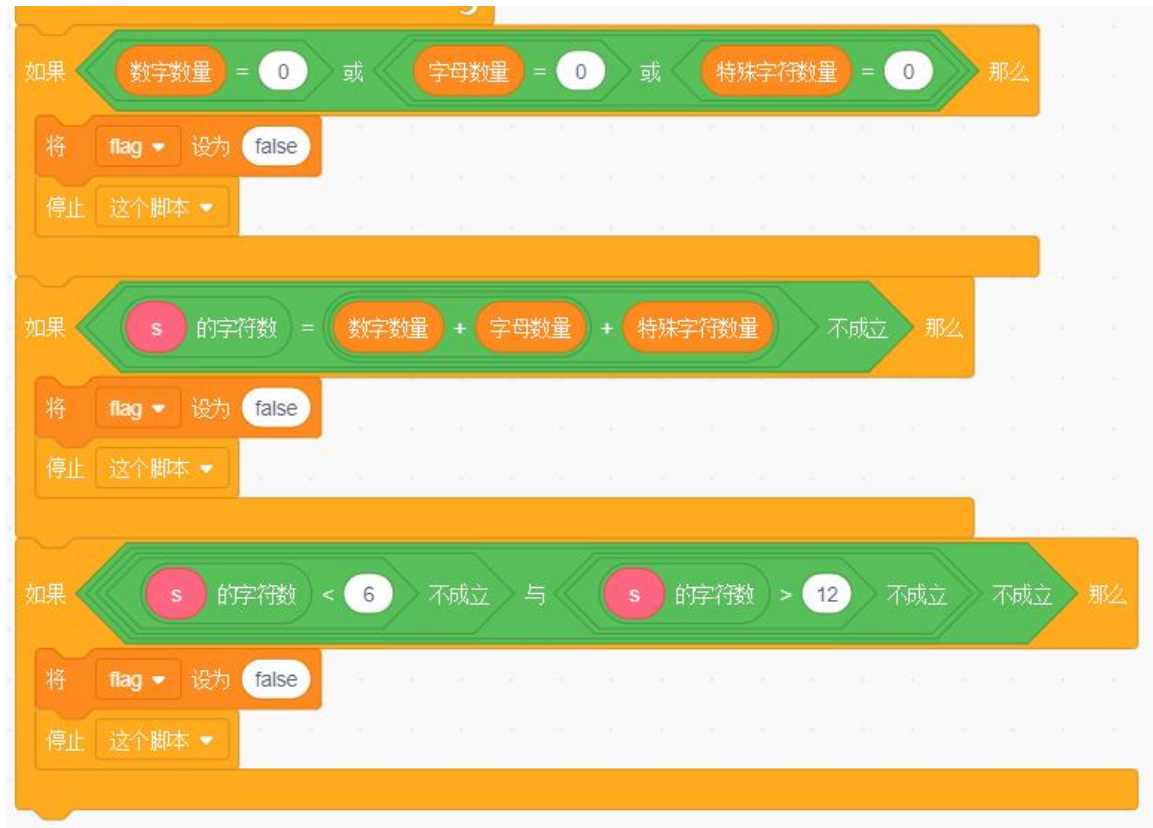
测试用例 5:

```
lst = [!@#$$12345qQ, 123XYZxyz@@, helpCCF9!@#$, 123!@#&^%ZAqaz];  
result = [123XYZxyz@@, helpCCF9!@#$]
```

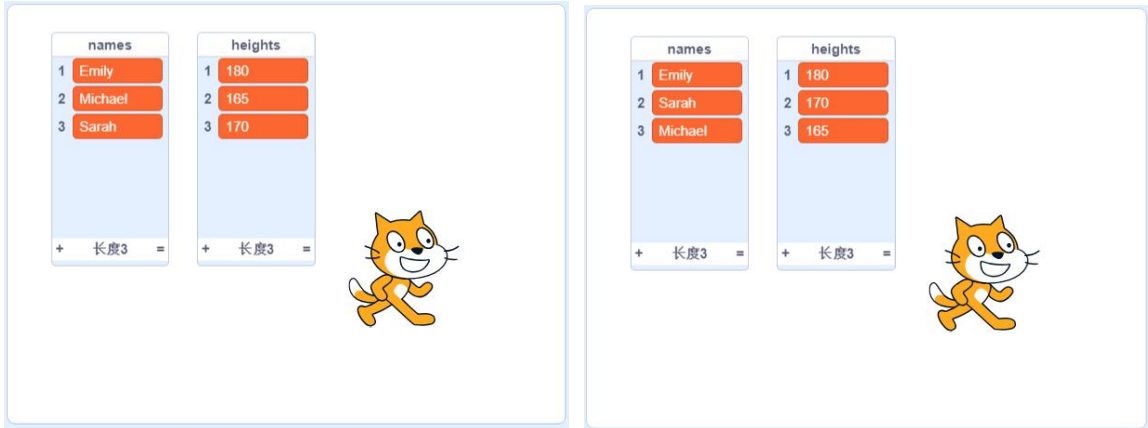
【参考程序】







2、按身高排序



【题目描述】

默认小猫角色和白色背景。有两个列表，第一个列表“names”存储名字，第二个列表“heights”存储这组名字对应的身高，这些身高由互不相同的正整数组成。

请按身高由大到小排序，同时能够得到对应名字的列表“names”。

例如：

名字列表：names = ['Emily' , ' Michael' , ' Sarah']

身高列表：heights = [180,165,170]

按照身高由大到小排序后，Emily 最高，接着是 Sarah 和 Michael

最后得到的名字列 heights = ['Emily' , ' Sarah' , ' Michael']

【输入描述】

新建两个列表，“names”存储一组姓名，“heights”存储这组姓名对应的身高，这些身高是互不相同的正整数。

如下图所示：

names	
1	Emily
2	Michael
3	Sarah
+ 长度3 =	

heights	
1	180
2	165
3	170
+ 长度3 =	

【输出描述】

无需新建，仍用以前的列表“names”，存储按照身高由大到小排序后，对应的姓名。
如下图所示：

names	
1	Emily
2	Sarah
3	Michael
+ 长度3 =	

【输入样例】

```
names = [ 'Emily' , ' Michael' , ' Sarah' ]
```

```
heights = [180,165,170]
```

【输出样例】

```
names = [ 'Emily' , ' Sarah' , ' Michael' ]
```

【输入样例】

```
names = [ 'Mary' , ' John' , ' Emma' , ' Tom' ]
```

```
heights = [160, 180, 170, 157]
```

【输出样例】

```
names = [ ' John' , ' Emma' , ' Mary' , ' Tom' ]
```

【评分标准】

一共 5 组测试用例，通过一组测试用例得 5 分。

测试用例 1:

```
names = [ 'Alice' , ' Bob' , ' Tom' ], heights = [155, 185, 150];
```

```
names = [ ' Bob' , 'Alice' , ' Tom' ]
```

测试用例 2:

```
names = [ 'Nicholas' , 'Abigail' , 'Benjamin' , 'Elizabeth' ],
```

```
heights = [160, 175, 158, 180];
```

```
names = [ 'Elizabeth' , 'Abigail' , 'Nicholas' , 'Benjamin' ]
```

测试用例 3:

```
names = [ 'William' , 'Olivia' , 'James' , ' Sarah' , ' Nicholas' ],
```

```
heights = [174, 160, 178, 158, 180];
```

```
names = [ ' Nicholas' , 'James' , 'William' , 'Olivia' , ' Sarah' ]
```

测试用例 4:

```
names = [ 'Olivia' , 'William' ], heights = [178, 180];
```

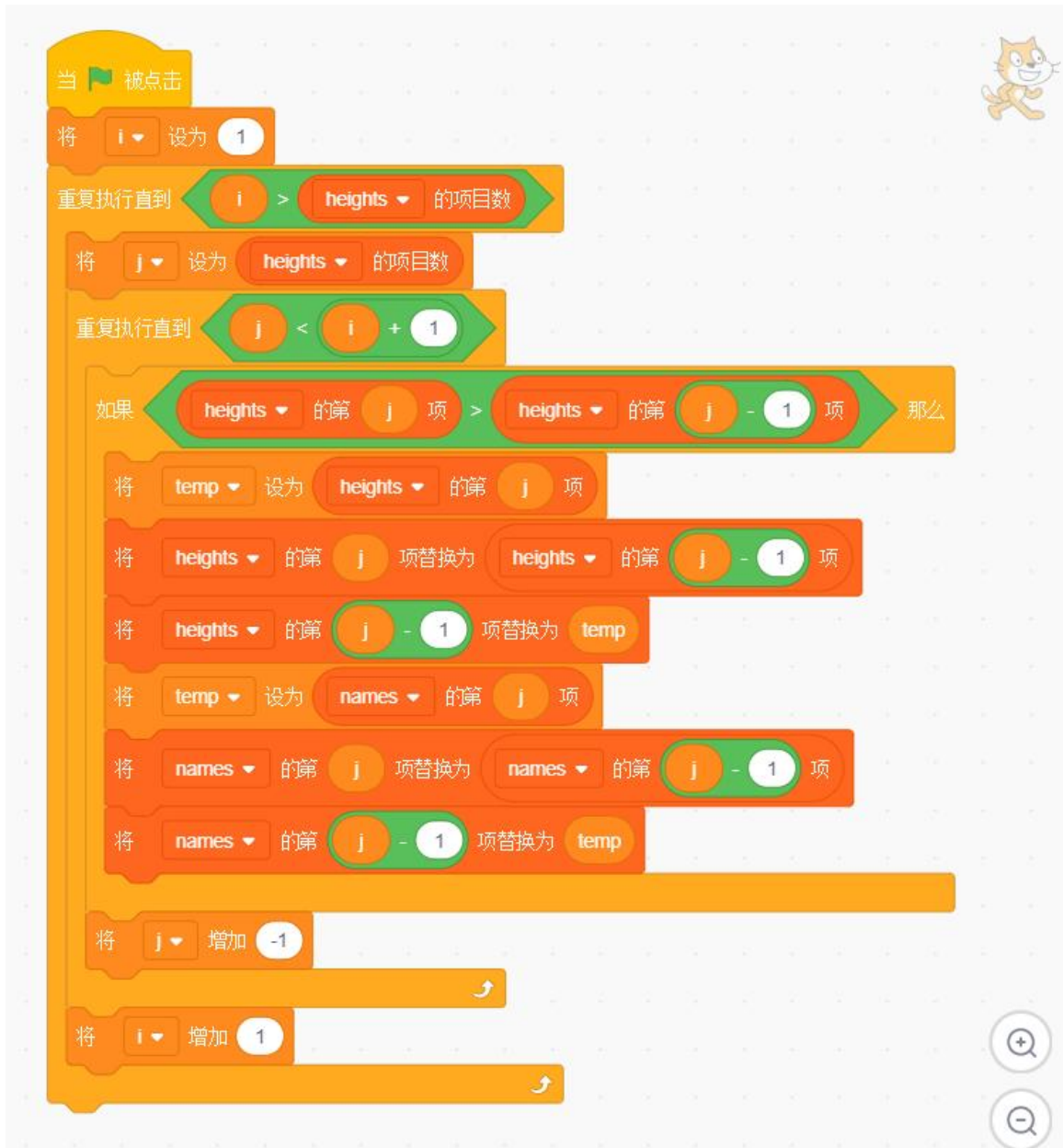
```
names = [ 'William' , 'Olivia' ]
```

测试用例 5:

```
names = [ 'Emily' ], heights = [175];
```

```
names = [ 'Emily' ]
```

【参考程序】



当 被点击

将 i 设为 1

重复执行直到 $i > \text{heights}$ 的项目数

将 j 设为 heights 的项目数

重复执行直到 $j < i + 1$

如果 heights 的第 j 项 $>$ heights 的第 $j - 1$ 项 那么

将 temp 设为 heights 的第 j 项

将 heights 的第 j 项替换为 heights 的第 $j - 1$ 项

将 heights 的第 $j - 1$ 项替换为 temp

将 temp 设为 names 的第 j 项

将 names 的第 j 项替换为 names 的第 $j - 1$ 项

将 names 的第 $j - 1$ 项替换为 temp

将 j 增加 -1

将 i 增加 1

The script implements a bubble sort algorithm. It starts with a 'when clicked' event, initializes i to 1, and enters a loop that continues until i is greater than the number of items in the 'heights' list. Inside this loop, j is initialized to the length of the 'heights' list, and another loop continues until j is less than $i + 1$. Within this inner loop, an 'if' statement checks if the item at index j in the 'heights' list is greater than the item at index $j - 1$. If true, it swaps the items at these indices using a temporary variable 'temp'. The same logic is applied to the 'names' list. After each inner loop iteration, j is decremented by 1, and after the outer loop iteration, i is incremented by 1. A Scratch cat icon is in the top right, and zoom controls are in the bottom right.