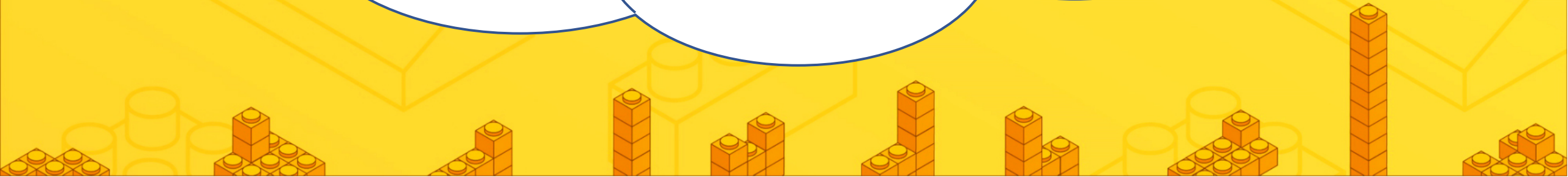
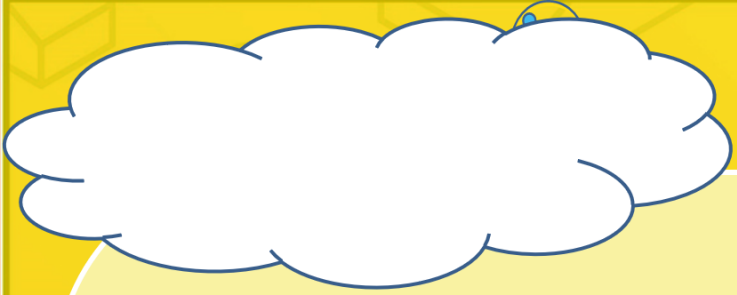


**Let me remember**



# Target

- **Learn the application of variables.**
- **Complete an LED lighting scheme using multiple variable storage and applications.**
- **Understand how the robot performs memory storage and retrieval.**



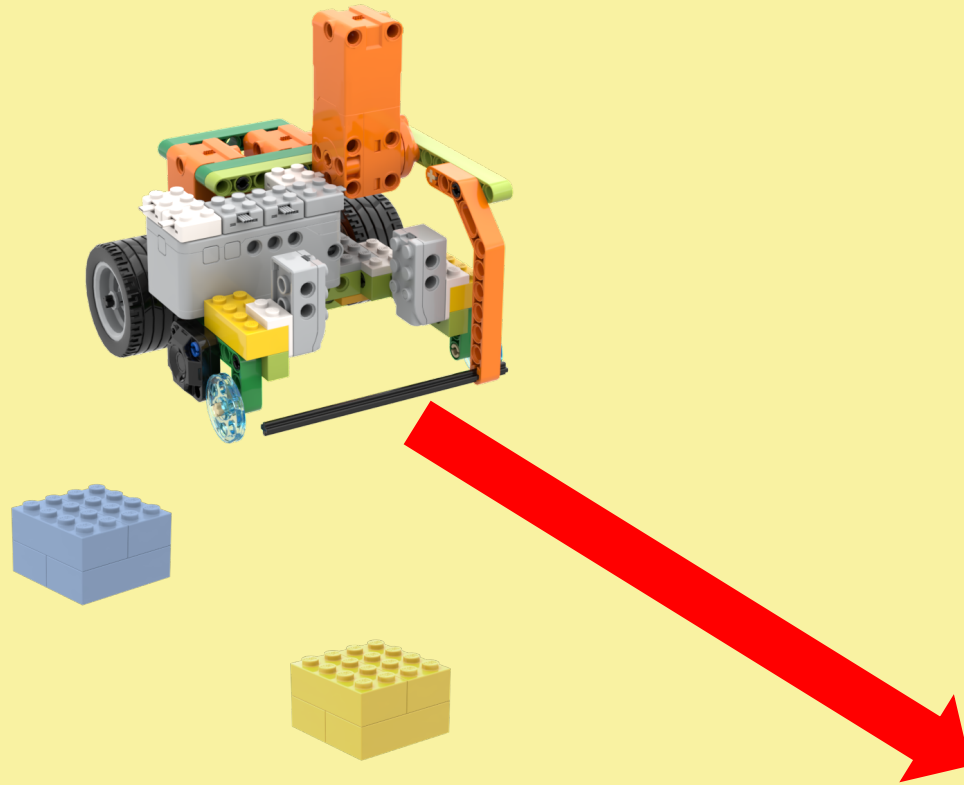
# 01 Task





# Task

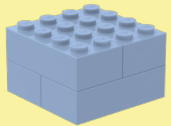
- **Task 1: Detect two colors, and then light up the color LEDs in sequence.**



# Task Segmentation

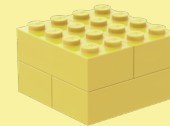
- First, complete the color detection.
- You can refer to the line-following program for guidance.

Detect the color



Not detect  
the color

Detect the color



Not detect  
the color



# Task Segmentation

- Referrable program



```
when clicked
  set 1# ext servo to keep running at 30 (-100~100)% power on anticlockwise
  set 2# ext servo to keep running at 30 (-100~100)% power on clockwise
  wait until 1# color sensor's color code < 50
  wait 0.5 seconds
  wait until 1# color sensor's color code > 50
  wait 0.5 seconds
  wait until 1# color sensor's color code < 50
  wait 0.5 seconds
  wait until 1# color sensor's color code > 50
  stop all ext motor(s)
```

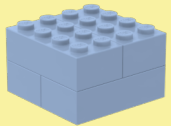
The image shows a Scratch script on a grid background. The script starts with a 'when clicked' event block. It then contains two 'set' blocks for external servos: '1#' is set to 30% power, anticlockwise; '2#' is set to 30% power, clockwise. This is followed by a sequence of 'wait until' and 'wait' blocks. The sequence is: 'wait until 1# color sensor's color code < 50', 'wait 0.5 seconds', 'wait until 1# color sensor's color code > 50', 'wait 0.5 seconds', 'wait until 1# color sensor's color code < 50', 'wait 0.5 seconds', 'wait until 1# color sensor's color code > 50'. The script ends with a 'stop all ext motor(s)' block.



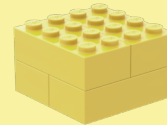
# Task Segmentation

- Use variables to store the two colors.
- After the robot stops, read, evaluate, and display the colors.

variable a



variable b



Read variable a  
Read variable b



# Task Segmentation

## Referrable program

### Color Storage

```
when clicked
  turn the all lights off
  set 1# ext servo to keep running at 30 (-100~100)% power on anticlockwise
  set 2# ext servo to keep running at 30 (-100~100)% power on clockwise
  wait until 1# color sensor's color code < 50
  set a to 1# color sensor's color code
  wait 0.3 seconds
  wait until 1# color sensor's color code > 50
  wait 0.3 seconds
  wait until 1# color sensor's color code < 50
  set b to 1# color sensor's color code
  wait 0.3 seconds
  wait until 1# color sensor's color code > 50
  stop all ext motor(s)
```

### Color reading and judgment

```
if a = 5 then
  set all lights color to yellow
  wait 2 seconds
if a = 4 then
  set all lights color to blue
  wait 2 seconds
wait 0.3 seconds
if b = 5 then
  set all lights color to yellow
  wait 2 seconds
if b = 4 then
  set all lights color to blue
  wait 2 seconds
wait 0.3 seconds
```

# Precautions

```
if a = 5 then
  set all lights color to yellow
  wait 2 seconds
if a = 4 then
  set all lights color to blue
  wait 2 seconds
wait 0.3 seconds
if b = 5 then
  set all lights color to yellow
  wait 2 seconds
if b = 4 then
  set all lights color to blue
  wait 2 seconds
wait 0.3 seconds
```

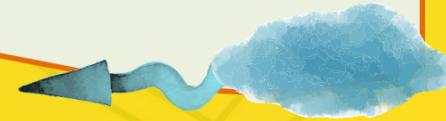
**Note: Color code 5 is actually green. When the color sensor detects a yellow block, color code 5 appears when the distance is 1-3 cm, and the correct color code 7 only appears when the distance is less than 1 cm.**

**To accurately distinguish between yellow and green, let's make a slight structural adjustment.**

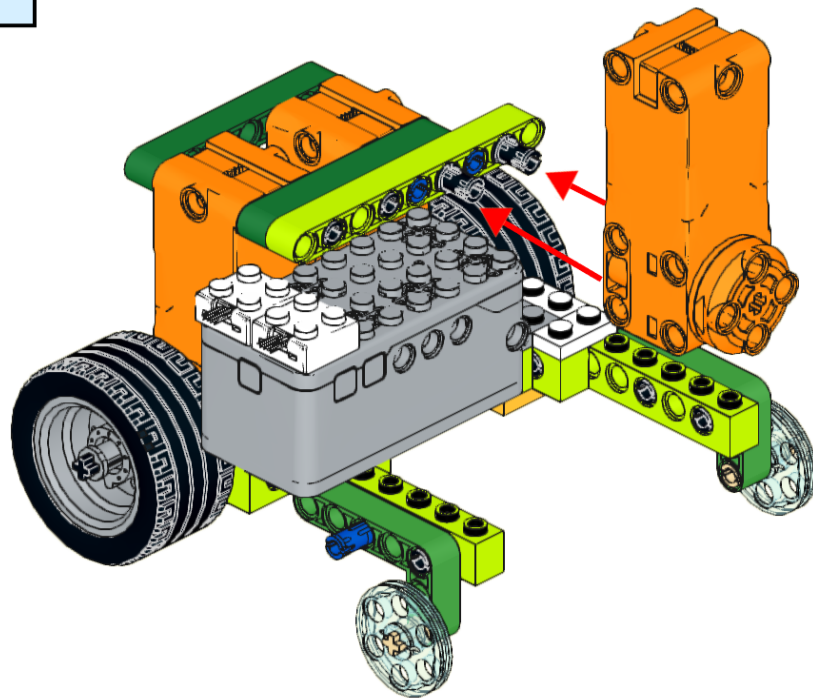
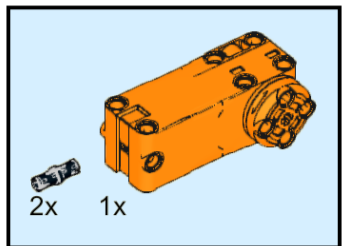




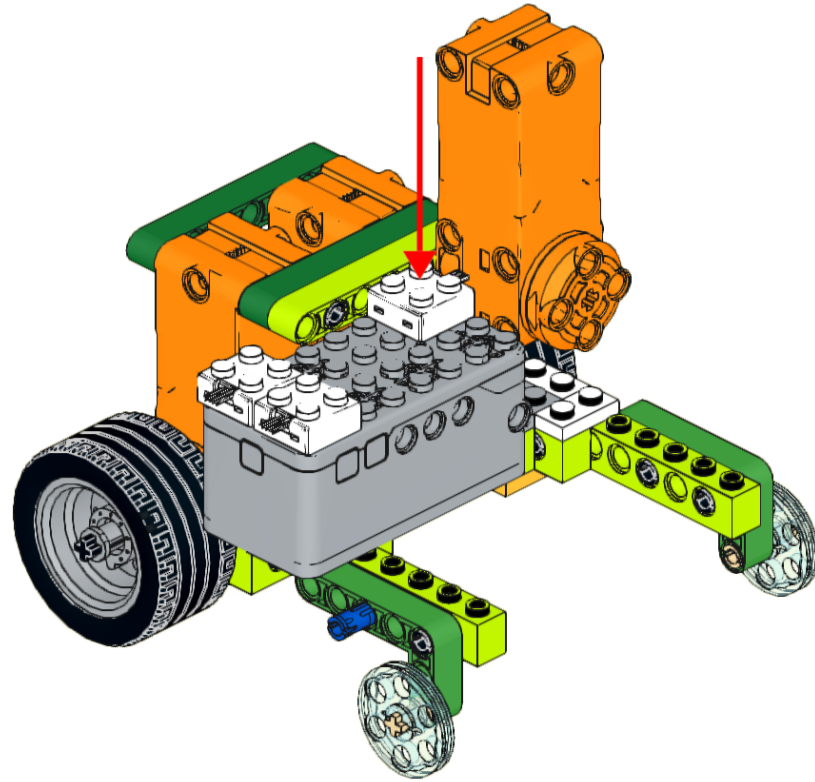
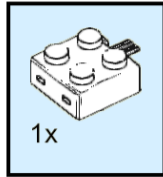
# 02 Assembly



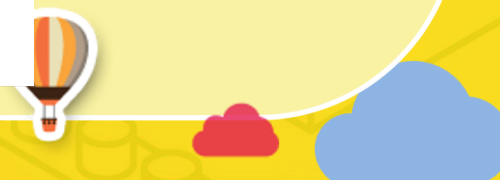
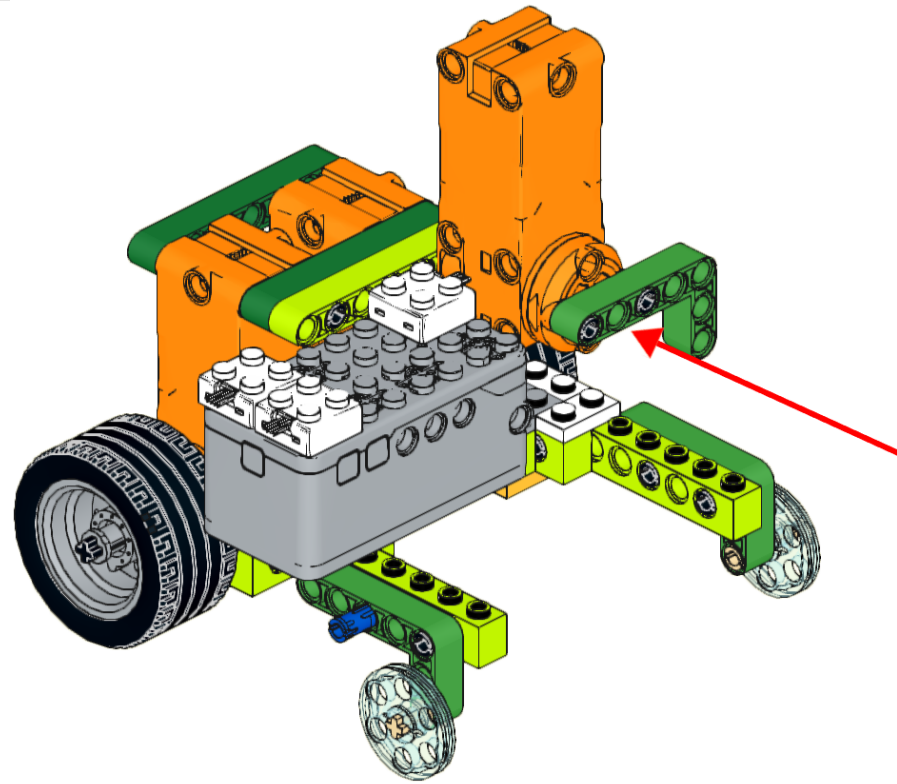
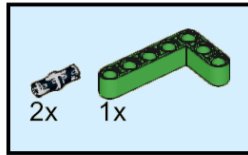
# Assembly



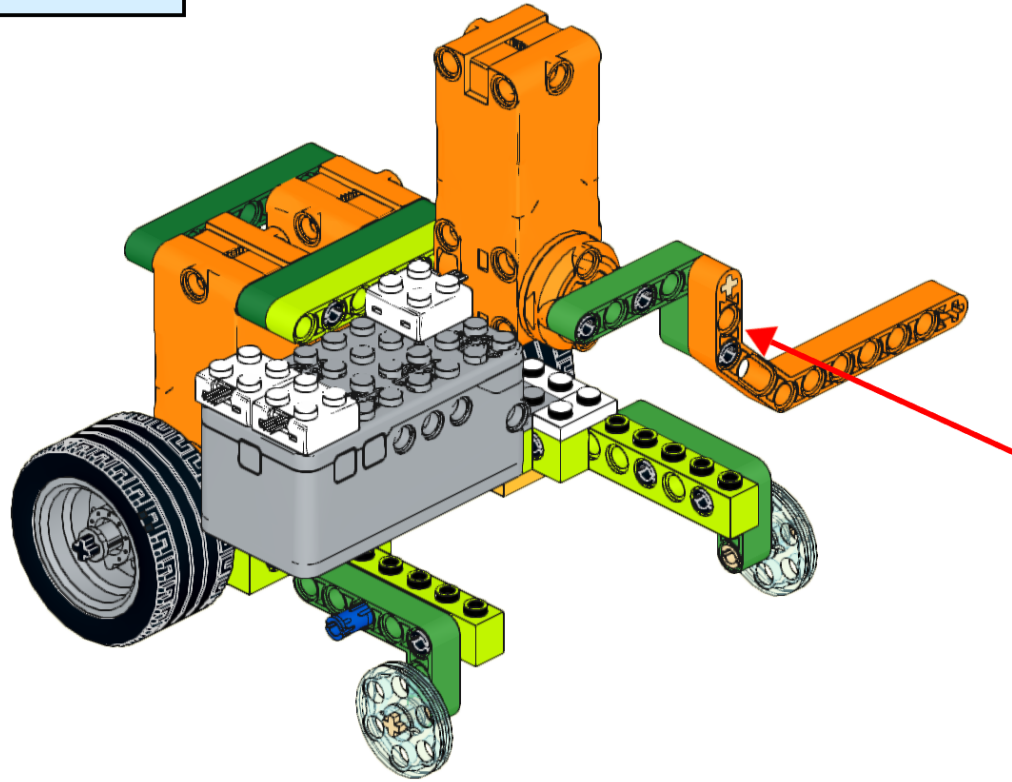
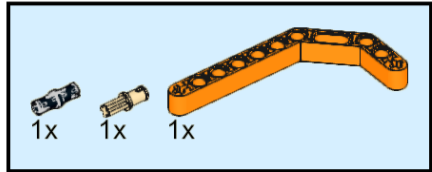
# Assembly



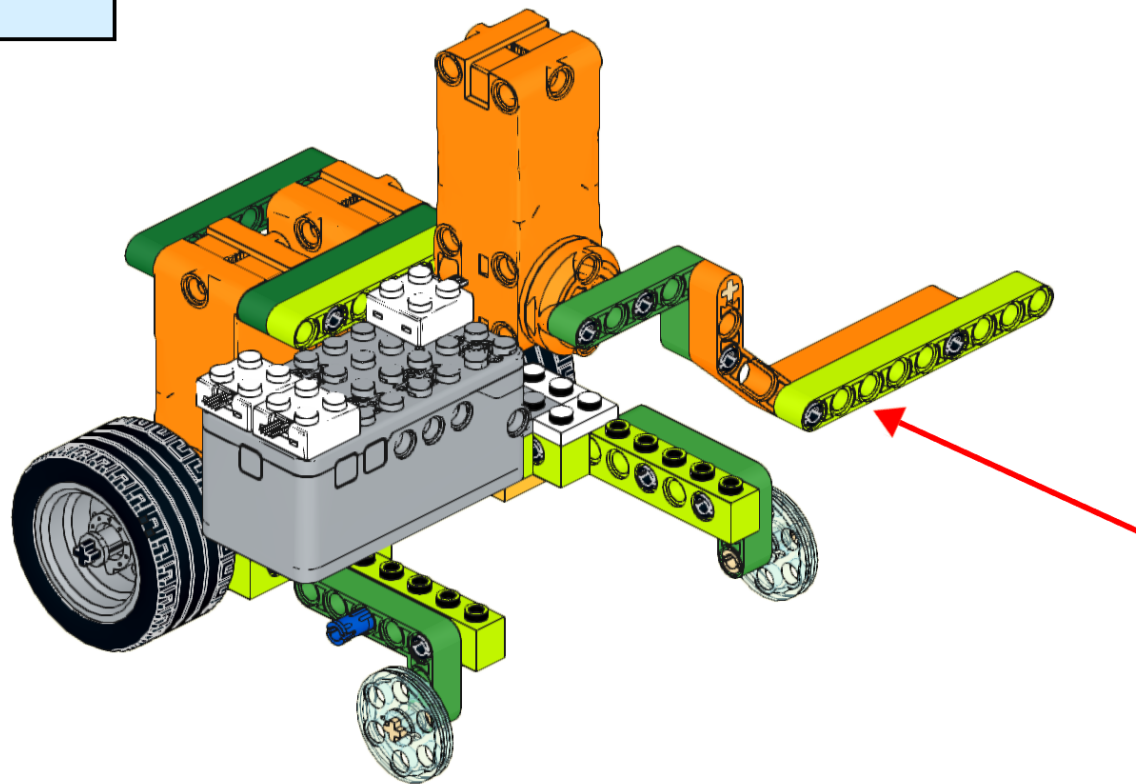
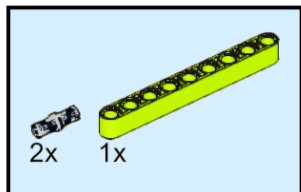
# Assembly



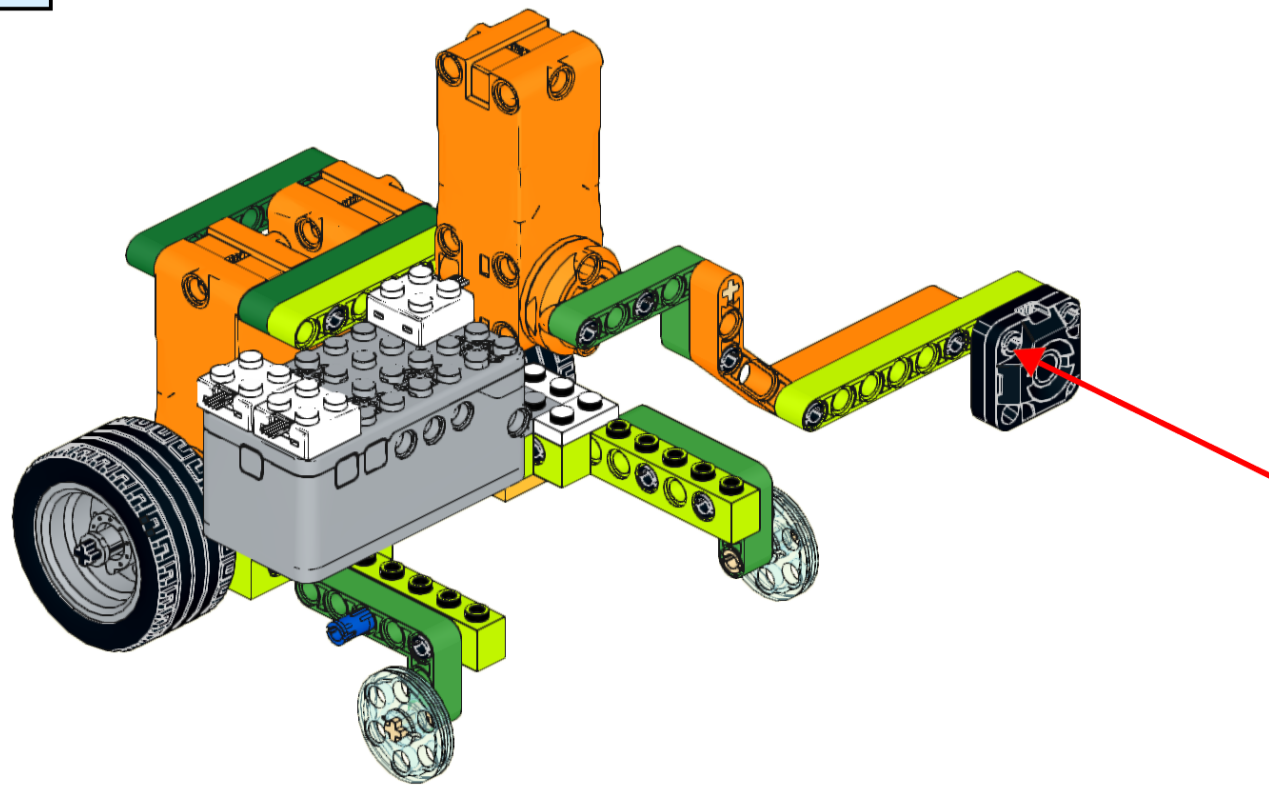
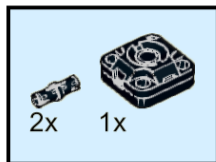
# Assembly



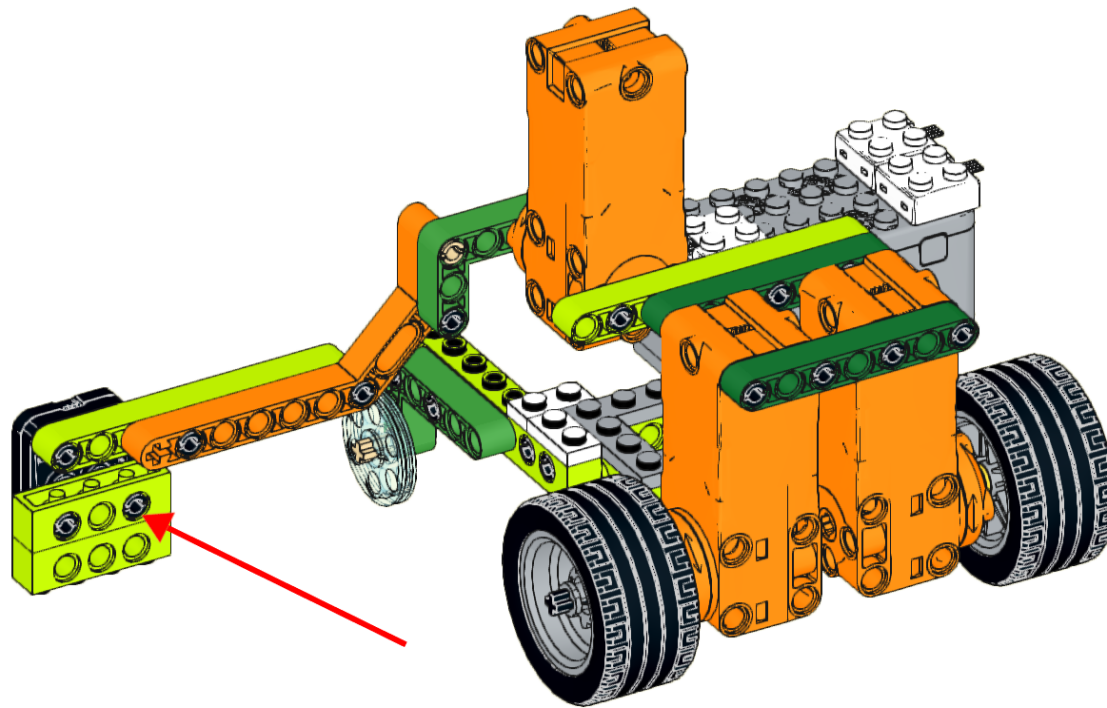
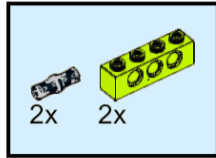
# Assembly



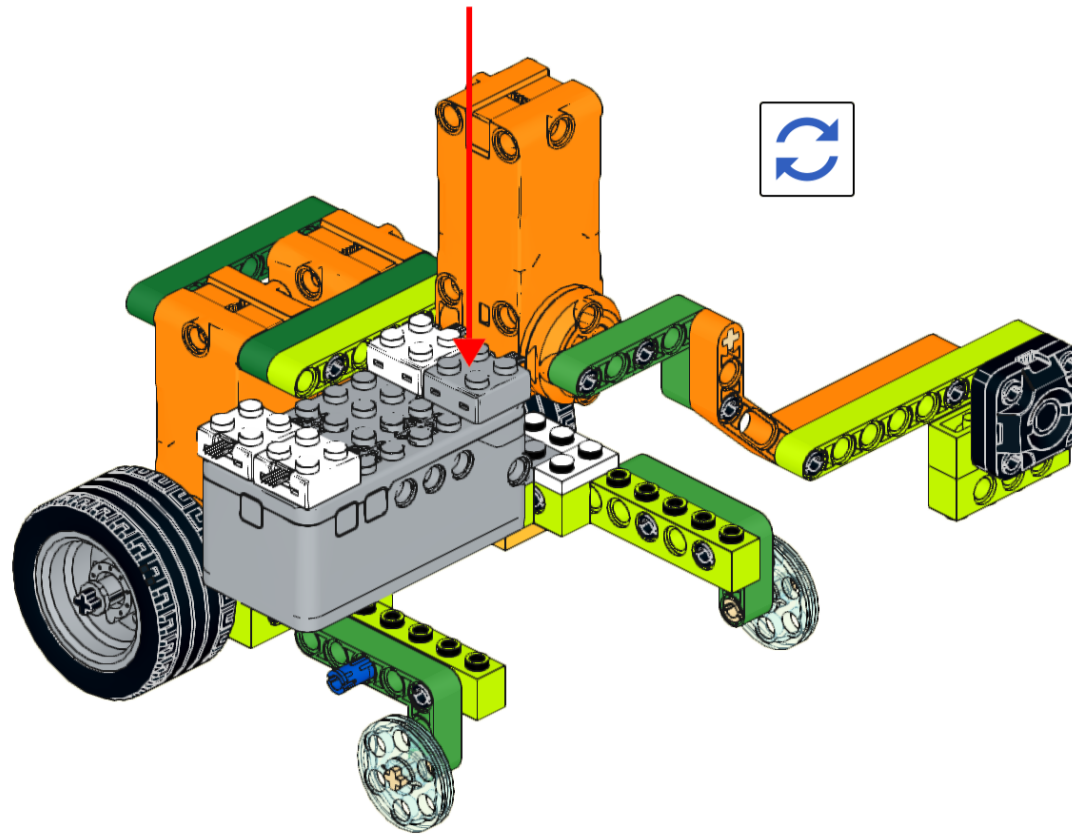
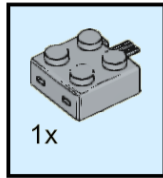
# Assembly



# Assembly

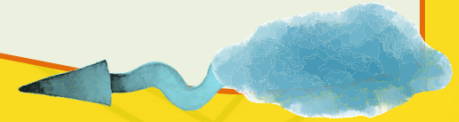


# Assembly





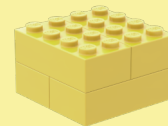
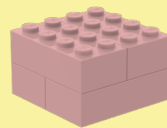
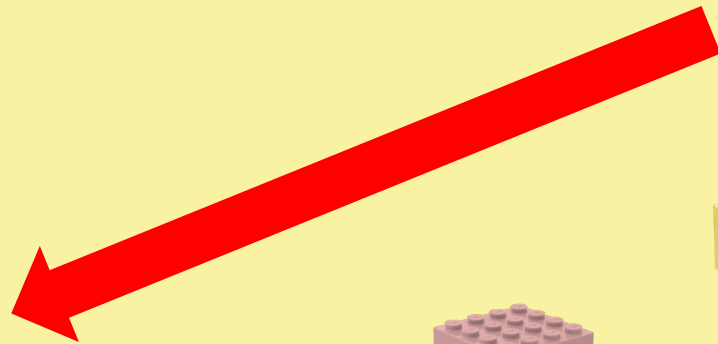
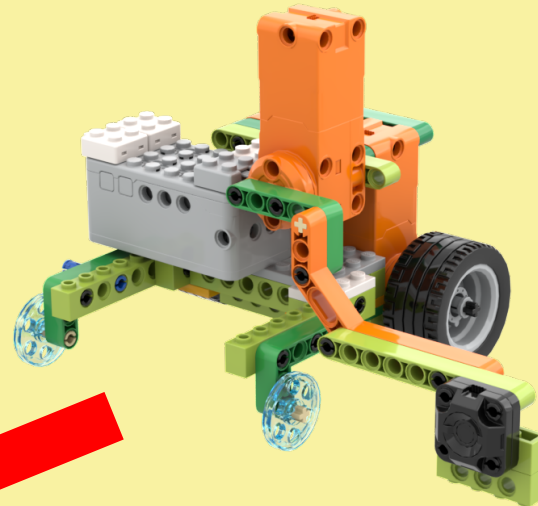
# 03 Task





# Task

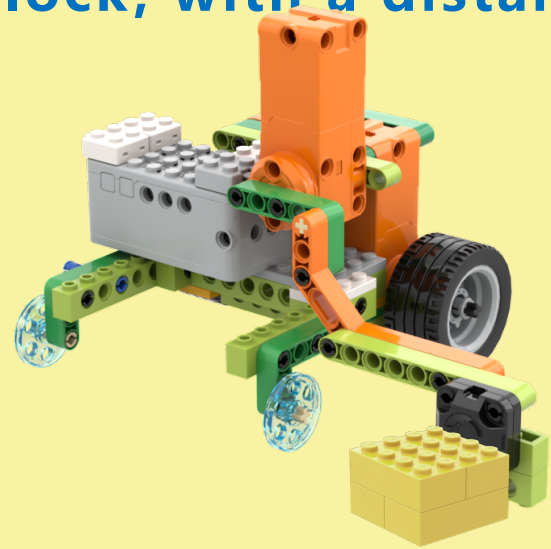
- **Task 2: Detect red and yellow. After detection, light up the color LEDs in the order specified by the teacher.**





## Task Segmentation

- Before starting, raise the robotic arm to ensure the robot meets the size requirements. After starting, lower the robotic arm and align the color sensor with the block.
- When the color sensor detects the color, for more accurate detection, the robot should move forward a small distance, bringing the sensor closer to the block, with a distance of less than 1 cm.



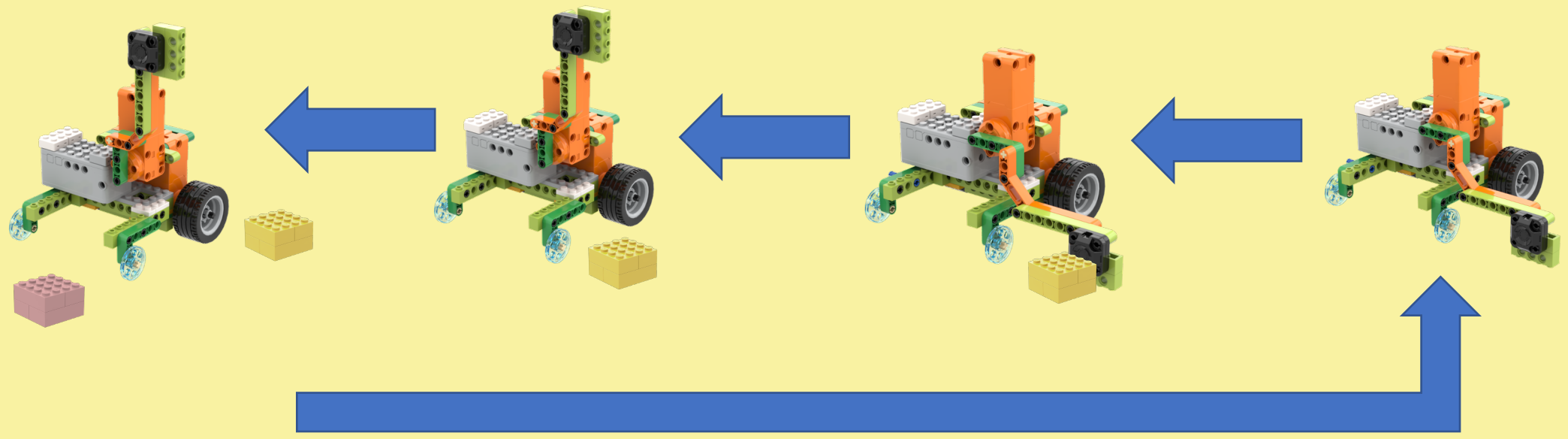


# Task Segmentation

## • Task Flow

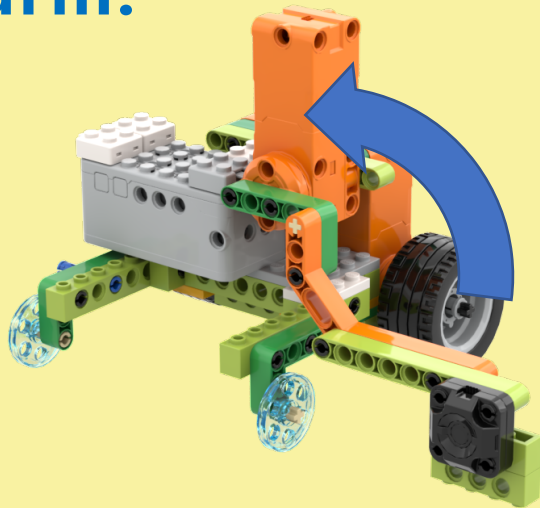
### Color detection process

4. Go a little further    3. Raise the robot arm    2. Detecte the color    1. Put down the robotic arm

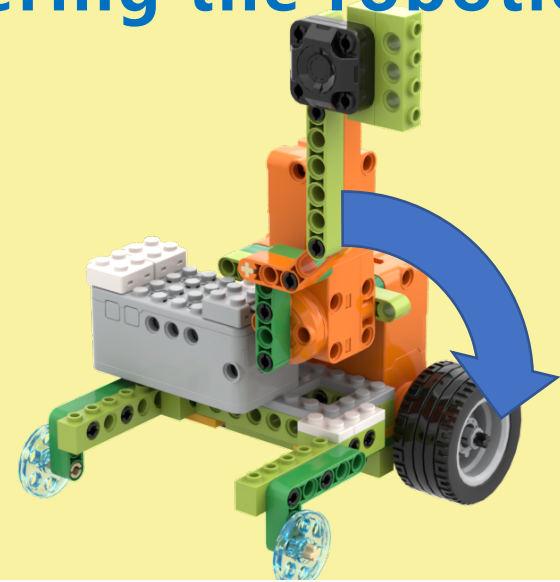


# Task Segmentation

- Complete my module for raising and lowering the robotic arm.



It is recommended to use the absolute angle module for controlling the robotic arm. The absolute angle module can be used for operations up to 360 degrees.



```
set 3# ext servo to rotate absolute angle 0 (0~359) degree anticlockwise at 30 (0~100)% power  
wait 0.1 seconds  
wait until [servo] is 3# ext servo done  
wait 0.3 seconds
```

```
set 3# ext servo to rotate absolute angle 94 (0~359) degrees clockwise at 30 (0~100)% power  
wait 0.1 seconds  
wait until [servo] is 3# ext servo done  
wait 0.3 seconds
```



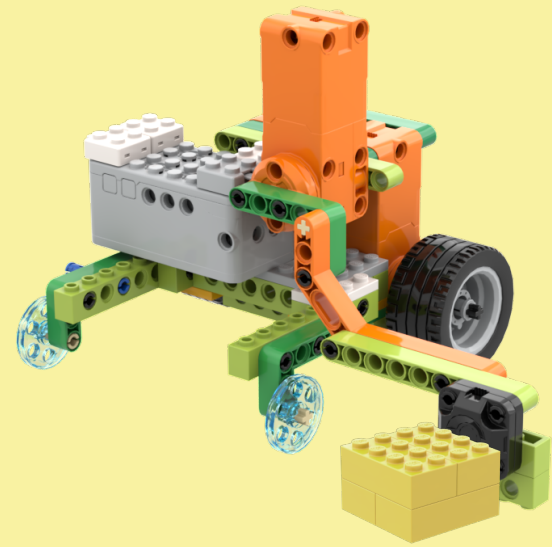


# Task Segmentation

- To accurately detect colors, after detecting a color, move forward a short distance to get close to the color block, and then make the record.

```
set 1# ext servo to keep running at 30 (-100~100)% speed on anticlockwise
set 2# ext servo to keep running at 30 (-100~100)% speed on clockwise
wait until 1# color sensor's color code < 50
wait 0.3 seconds
stop all ext servo(s)
wait 0.3 seconds
```

```
set a to 1# color sensor's color code
wait 0.5 seconds
```

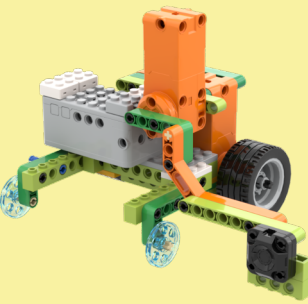




# Task Segmentation 1

## • Task Flow

1. Put down the robotic arm



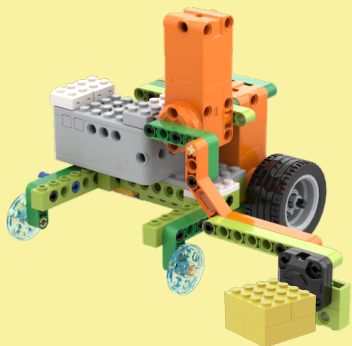
```
when clicked
  set a to 0
  set b to 0
  set c to 0
  set 3# ext servo to rotate absolute angle 95 (0~359) degrees clockwise at 30 (0~100)% power
  wait 0.1 seconds
  wait until 3# ext servo done
  turn the all lights off
  wait 0.1 seconds
```



# Task Segmentation 2

- Task Flow

## 2. Detecte the color



```
define forward
  set 1# ext servo to keep running at 30 (-100~100)% speed on anticlockwise
  set 2# ext servo to keep running at 30 (-100~100)% speed on clockwise
  wait until 1# color sensor's color code < 50
  wait 0.3 seconds
  stop all ext servo(s)
  wait 0.3 seconds
```

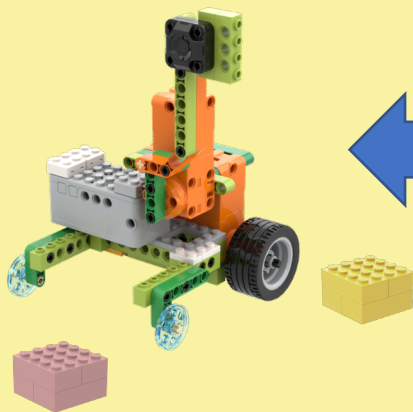
```
forward
  set a to 1# color sensor's color code
  wait 0.5 seconds
```



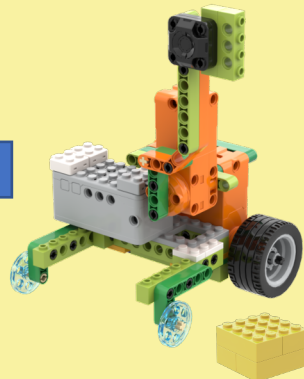
# Task Segmentation 3

## • Task Flow

4. Go a little further



3. Raise the robot arm



```
define arm
  set 3# ext servo to rotate absolute angle 0 (0~359) degrees anticlockwise at 30 (0~100)% power
  wait 0.1 seconds
  wait until [3#] is [3#] ext servo done
  set 1# ext servo to keep running at 30 (-100~100)% speed on anticlockwise
  set 2# ext servo to keep running at 30 (-100~100)% speed on clockwise
  wait 0.5 seconds
  stop all ext servo(s)
  set 3# ext servo to rotate absolute angle 94 (0~359) degrees clockwise at 30 (0~100)% power
  wait 0.1 seconds
  wait until [3#] is [3#] ext servo done
  wait 0.3 seconds
```



# Task Segmentation 4

## • Task Flow

```
define colors color
if color = 9 then
  set all lights color to red
if color = 7 then
  set all lights color to yellow
if color = 5 then
  set all lights color to green
if color = 4 then
  set all lights color to blue
```

- After completing the detection and saving, finally display the color.

```
colors a
wait 2 seconds
```



# Task Integration

- My module uses the modules from the first three pages. Connect the overall program to complete the detection of multiple color blocks.

```
when green flag clicked
  set a to 0
  set b to 0
  set c to 0
  set 3# ext servo to rotate absolute angle 95 (0~359) degrees clockwise at 30 (0~100)% power
  wait 0.1 seconds
  wait until 3# ext servo is done
  turn the all lights off
  wait 0.1 seconds
```

Connect the remaining program content with the right side.

```
forward
set a to 1# color sensor's color code
wait 0.5 seconds
arm
forward
set b to 1# color sensor's color code
wait 0.5 seconds
arm
forward
set c to 1# color sensor's color code
wait 0.5 seconds
colors a
wait 2 seconds
colors b
wait 2 seconds
colors c
```