```
# This program written for ROBOMASTER S1
# Watch the video that happend

import time

def do_maze():
    chassis_ctrl.move_with_distance(0,5)# move ahead 5 meters
    chassis_ctrl.move_with_distance(0,0.8)# move ahead 0.8 meters
    chassis_ctrl.move_with_distance(-90,.8)# move left 1 meter ###############
    chassis_ctrl.move_with_distance(0,0.50)# move ahead 0.35 meters
    chassis_ctrl.move_with_distance(90,1.58)# move right 1.64 meter
    chassis_ctrl.move_with_distance(0,0.5)# move ahead 0.42 meters
    chassis_ctrl.move_with_distance(-90,0.55)# move left 0.51 meter
    chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,57.6)############
    chassis_ctrl.move_with_distance(0,1.58)# move right 1.64 meter
    chassis_ctrl.rotate_with_degree(rm_define.clockwise,57.5)
    chassis_ctrl.move_with_distance(0,0.65)# move ahead 0.52 meters
    chassis_ctrl.move_with_distance(90,0.84)# move right 0.835 meter
    chassis_ctrl.move_with_distance(0,0.20)# move ahead 0.385 meters
    time.sleep(8)# First reset point B

def execute_room_C():
    Room1Type = 3

if Room1Type == 1:# 1 - There is a fire, find marker and shoot it!
        chassis_ctrl.move_with_distance(0,2.11)# Move forward 2.11 meters
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)# Rotate the chassis 90 degrees to the right:
        chassis_ctrl.move_with_distance(0,2.12)# Move forward 2.12 meters
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)# Rotate the chassis 90 degrees to the right:

        scan_for_marker()
        gimbal_ctrl.yaw_ctrl(0)
        gimbal_ctrl.pitch_ctrl(0)

# After firing, robot should leave
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)
        chassis_ctrl.move_with_distance(0,2.12)
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        chassis_ctrl.move_with_distance(0,2.11)
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)
        chassis_ctrl.move_with_distance(0,5)
        chassis_ctrl.move_with_distance(0,0.48)
# At reset point D

elif Room1Type == 2:# 2 Poisonous Room - Skip
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        gimbal_ctrl.yaw_ctrl(0)
        gimbal_ctrl.pitch_ctrl(0)
```

```
        chassis_ctrl.move_with_distance(0,5)
        chassis_ctrl.move_with_distance(0,0.48)

# At reset point D

else:# 3 - Room has a person - Locate the person and take them to safety
        chassis_ctrl.move_with_distance(0,2.11)# Move forward 2.11 meters
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)# Rotate the chassis 90 degrees to the right:
        chassis_ctrl.move_with_distance(0,2.12)# Move forward 2.12 meters
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)# Rotate the chassis 90 degrees to the right:

        scan_for_person()

# After scanning, the robot should leave.
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)
        chassis_ctrl.move_with_distance(0,2.12)
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        chassis_ctrl.move_with_distance(0,2.11)

# Should be at point outside first room (Room 227)
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        chassis_ctrl.move_with_distance(0,5)
        chassis_ctrl.move_with_distance(0,1.46)
# Should be at first reset point B
        time.sleep(8)

        chassis_ctrl.move_with_distance(0,5)
        chassis_ctrl.move_with_distance(0,3.62)

# Drop off person at start then turns around
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,180)
        gimbal_ctrl.yaw_ctrl(0)
        gimbal_ctrl.pitch_ctrl(0)
        time.sleep(8)

        chassis_ctrl.move_with_distance(0,5)
        chassis_ctrl.move_with_distance(0,3.62)

        time.sleep(8)# At first reset point B
        chassis_ctrl.move_with_distance(0,5)
        chassis_ctrl.move_with_distance(0,1.56)

# Should be outside first room again

        chassis_ctrl.move_with_distance(0,5)
        chassis_ctrl.move_with_distance(0,0.97)

# At rest point D
```

```
def execute_room_E():

    Room2Type = 1

if Room2Type == 1:# 1 - There is a fire, find marker and shoot it!
        chassis_ctrl.move_with_distance(0,2.4)
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)
        chassis_ctrl.move_with_distance(0,1.68)
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        chassis_ctrl.move_with_distance(0,4.70)
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)
        chassis_ctrl.move_with_distance(0,1.75)
        gimbal_ctrl.yaw_ctrl(0)
        gimbal_ctrl.pitch_ctrl(10)
        scan_for_marker()

# After firing it should exit the room again
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,180)
        chassis_ctrl.move_with_distance(0,1.70)
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        chassis_ctrl.move_with_distance(0,4.70)
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)
        chassis_ctrl.move_with_distance(0,1.60)
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        chassis_ctrl.move_with_distance(0,2.4)
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)
        chassis_ctrl.move_with_distance(0,4.0)########## Check Measurement

# Should be at rest point F

elif Room2Type == 2:# 2 Poisonous Room - Skip
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        gimbal_ctrl.yaw_ctrl(0)
        gimbal_ctrl.pitch_ctrl(0)
        chassis_ctrl.move_with_distance(0,4.20)
# Should be at reset point F

else:# 3 - Room has a person - Locate the person and take them to safety
# Goes into 2nd room, scans person and escorts back to start
        chassis_ctrl.move_with_distance(0,2.4)
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)
        chassis_ctrl.move_with_distance(0,1.68)########
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        chassis_ctrl.move_with_distance(0,4.70)
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)
        chassis_ctrl.move_with_distance(0,1.30)######### Check Measurement - 1.30
        gimbal_ctrl.yaw_ctrl(+180)# Rotate gimbal 180 degrees
```

```
        scan_for_person()

        chassis_ctrl.rotate_with_degree(rm_define.clockwise,180)
        chassis_ctrl.move_with_distance(0,1.75)
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        chassis_ctrl.move_with_distance(0,4.70)
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)
        chassis_ctrl.move_with_distance(0,1.20)############# check 1.48
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        chassis_ctrl.move_with_distance(0,2.4)
# Should be outside 2nd room now
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        chassis_ctrl.move_with_distance(0,4.80)
# Should be at Reset point D
        time.sleep(8)

        chassis_ctrl.move_with_distance(0,5.0)
        chassis_ctrl.move_with_distance(0,5.0)
        chassis_ctrl.move_with_distance(0,1.90)
# Should be at rest point B
        time.sleep(8)

        chassis_ctrl.move_with_distance(0,5.0)
        chassis_ctrl.move_with_distance(0,3.62)
# Should be at start point A - turns around and resets
        chassis_ctrl.rotate_with_degree(rm_define.clockwise,180)
        gimbal_ctrl.yaw_ctrl(0)
        gimbal_ctrl.pitch_ctrl(0)
        time.sleep(8)

        chassis_ctrl.move_with_distance(0,5.0)
        chassis_ctrl.move_with_distance(0,3.62)

# maze reset point B
        time.sleep(8)

        chassis_ctrl.move_with_distance(0,5.0)
        chassis_ctrl.move_with_distance(0,5.0)
        chassis_ctrl.move_with_distance(0,2.04)
# Reset point D
        time.sleep(8)

        chassis_ctrl.move_with_distance(0,4.8)
# Should be outside second room again

        chassis_ctrl.move_with_distance(0,4.30)
```

```
# At reset point F

def execute_room_G():
    Room3Type = 2

if Room3Type == 2:# 2 Poisonous Room - Skip
        chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)
        gimbal_ctrl.yaw_ctrl(0)
        gimbal_ctrl.pitch_ctrl(0)
        chassis_ctrl.move_with_distance(0,5)
        chassis_ctrl.move_with_distance(0,0.53)
# Should be at reset point H

def scan_for_marker():
# Move robot within 1 meter of marker before trying to scan
# Enable detection
    vision_ctrl.enable_detection(rm_define.vision_detection_marker)

    gimbal_ctrl.yaw_ctrl(-90)
    gimbal_ctrl.yaw_ctrl(+180)

def vision_recognized_marker_letter_F(msg):# Shoot the marker.
# Since you found the marker, turn detection off.
    vision_ctrl.disable_detection(rm_define.vision_detection_marker)
# Detect marker and aim.
    vision_ctrl.detect_marker_and_aim(rm_define.marker_letter_F)
# Shoot
    gun_ctrl.fire_once()

def vision_recognized_marker_number_one(msg):# Do something with the chasis and gimbal.
# Since you found the marker, turn detection off.
    vision_ctrl.disable_detection(rm_define.vision_detection_marker)

# Move chassis 180 degrees and back 180.
    chassis_ctrl.rotate_with_degree(rm_define.clockwise,180)
    chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,180)
# Make gimbal go up and down here.
    gimbal_ctrl.pitch_ctrl(30)

def vision_recognized_marker_number_two(msg):####### Do something with the LED lights. #########
####
# Since you found the marker, turn detection off.
    vision_ctrl.disable_detection(rm_define.vision_detection_marker)

# Change LED lights to Red
    led_ctrl.set_bottom_led(rm_define.armor_bottom_all,255,0,0, rm_define.effect_always_on)
    led_ctrl.set_top_led(rm_define.armor_top_all,255,0,0, rm_define.effect_always_on)
    time.sleep(3)
```

```
# Change LED lights back to blue
    led_ctrl.set_bottom_led(rm_define.armor_bottom_all,69,215,255, rm_define.effect_always_on)
    led_ctrl.set_top_led(rm_define.armor_top_all,69,215,255, rm_define.effect_always_on)

def vision_recognized_marker_number_three(msg):#######  Do both the chasis and gimbal and the LED l
ights. ############
# Since you found the marker, turn detection off.
    vision_ctrl.disable_detection(rm_define.vision_detection_marker)

# Change LED lights to Red
    led_ctrl.set_bottom_led(rm_define.armor_bottom_all,255,0,0, rm_define.effect_always_on)
    led_ctrl.set_top_led(rm_define.armor_top_all,255,0,0, rm_define.effect_always_on)

# Move chassis 180 degrees and back 180.
    chassis_ctrl.rotate_with_degree(rm_define.clockwise,180)
    chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,180)
# Make gimbal go up and down here.
    gimbal_ctrl.pitch_ctrl(35)
    gimbal_ctrl.pitch_ctrl(-10)
    gimbal_ctrl.pitch_ctrl(35)
    gimbal_ctrl.pitch_ctrl(-10)
    gimbal_ctrl.yaw_ctrl(0)

# Change LED lights back to blue
    led_ctrl.set_bottom_led(rm_define.armor_bottom_all,69,215,255, rm_define.effect_always_on)
    led_ctrl.set_top_led(rm_define.armor_top_all,69,215,255, rm_define.effect_always_on)

def scan_for_person():
    vision_ctrl.enable_detection(rm_define.vision_detection_people)# Enable person detection
    gimbal_ctrl.yaw_ctrl(-90)
    gimbal_ctrl.yaw_ctrl(+180)

def vision_recognized_people(msg):
# Since you found the person, turn detection off.
    vision_ctrl.disable_detection(rm_define.vision_detection_people)
# Make a sound
    media_ctrl.play_sound(rm_define.media_sound_recognize_success)

def do_dance():
# Robot Twist and Nod
    gimbal_ctrl.pitch_ctrl(-20)# down Nod
    led_ctrl.set_bottom_led(rm_define.armor_bottom_all,255,0,150, rm_define.effect_always_on)# Change ch
assis to Pink
    led_ctrl.set_top_led(rm_define.armor_top_all,255,0,150, rm_define.effect_always_on)# Change gimbal to
Pink
    gimbal_ctrl.pitch_ctrl(20)# up Nod

    chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)# Rotate clockwise
```

```
    gimbal_ctrl.pitch_ctrl(-20)# down Nod
    led_ctrl.set_bottom_led(rm_define.armor_bottom_all,161,255,69, rm_define.effect_always_on)# Change c
hassis to Green
    led_ctrl.set_top_led(rm_define.armor_top_all,161,255,69, rm_define.effect_always_on)# Change gimbal to
 Green
    gimbal_ctrl.pitch_ctrl(20)# up Nod
    chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)# Rotate Anti-clockwise
    gimbal_ctrl.pitch_ctrl(-20)# down Nod
    led_ctrl.set_bottom_led(rm_define.armor_bottom_all,36,103,255, rm_define.effect_always_on)# Change c
hassis to Blue
    led_ctrl.set_top_led(rm_define.armor_top_all,36,103,255, rm_define.effect_always_on)# Change gimbal to
 blue
    gimbal_ctrl.pitch_ctrl(20)# up Nod
    time.sleep(0.5)

# # Change to a variety of colours
# led_ctrl.set_bottom_led(rm_define.armor_bottom_all, 255, 0, 150, rm_define.effect_always_on) # Pink
# led_ctrl.set_top_led(rm_define.armor_top_all, 255, 0, 150, rm_define.effect_always_on)
# led_ctrl.set_bottom_led(rm_define.armor_bottom_all, 161, 255, 69, rm_define.effect_always_on) # Green
# led_ctrl.set_top_led(rm_define.armor_top_all, 161, 255, 69, rm_define.effect_always_on)
# led_ctrl.set_bottom_led(rm_define.armor_bottom_all, 36, 103, 255, rm_define.effect_always_on) # Blue
# led_ctrl.set_top_led(rm_define.armor_top_all, 36, 103, 255, rm_define.effect_always_on)
# led_ctrl.set_bottom_led(rm_define.armor_bottom_all, 255, 193, 0, rm_define.effect_always_on) # Yellow
# led_ctrl.set_top_led(rm_define.armor_top_all, 255, 193, 0, rm_define.effect_always_on)
# led_ctrl.set_bottom_led(rm_define.armor_bottom_all, 255, 0, 0, rm_define.effect_always_on) # Red
# led_ctrl.set_top_led(rm_define.armor_top_all, 255, 0, 0, rm_define.effect_always_on)
# led_ctrl.turn_off(rm_define.armor_all) # Turn off

    zig_zag()

# Repeat nodding & twist with different colours
    gimbal_ctrl.pitch_ctrl(-20)# down Nod
    led_ctrl.set_bottom_led(rm_define.armor_bottom_all,255,193,0, rm_define.effect_always_on)# Chassis Yel
low
    led_ctrl.set_top_led(rm_define.armor_top_all,255,193,0, rm_define.effect_always_on)# Gimbal Yellow
    gimbal_ctrl.pitch_ctrl(20)# up Nod

    chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)# Rotate clockwise

    gimbal_ctrl.pitch_ctrl(-20)# down Nod
    led_ctrl.set_bottom_led(rm_define.armor_bottom_all,128,0,128, rm_define.effect_always_on)# Chassis Pu
rple
    led_ctrl.set_top_led(rm_define.armor_top_all,128,0,128, rm_define.effect_always_on)# Gimbal Purple
    gimbal_ctrl.pitch_ctrl(20)# up Nod
    chassis_ctrl.rotate_with_degree(rm_define.anticlockwise,90)# Rotate Anti-clockwise
    gimbal_ctrl.pitch_ctrl(-20)# down Nod
    led_ctrl.set_bottom_led(rm_define.armor_bottom_all,255,0,0, rm_define.effect_always_on)# Chasis Red
```

```
    led_ctrl.set_top_led(rm_define.armor_top_all,255,0,0, rm_define.effect_always_on)# Gimbal Red
    gimbal_ctrl.pitch_ctrl(20)# up Nod
    time.sleep(0.5)

    zig_zag()

    led_ctrl.turn_off(rm_define.armor_all)

def zig_zag():
# Zig Zag forwards and then backwards with lights.

    led_ctrl.set_top_led(rm_define.armor_top_all,128,0,128, rm_define.effect_marquee)# Gimbal Purple
    led_ctrl.set_bottom_led(rm_define.armor_bottom_all,128,0,128, rm_define.effect_always_on)# Chassis Pu
rple
    chassis_ctrl.move_with_distance(-45,0.5)# move diagonally left 0.5 meters
    chassis_ctrl.move_with_distance(45,0.5)# move diagonally right 0.5 meters
    chassis_ctrl.move_with_distance(-135,0.5)# move back left diagonally 0.5m
    chassis_ctrl.move_with_distance(135,0.5)# move back right diagonally 0.5m
    time.sleep(0.5)

    led_ctrl.gun_led_on()
# Spin gimbal 90 degrees right & shine light.
    gimbal_ctrl.yaw_ctrl(90)
    gimbal_ctrl.yaw_ctrl(0)
    time.sleep(1)
    led_ctrl.gun_led_off()

    led_ctrl.set_top_led(rm_define.armor_top_all,128,0,128, rm_define.effect_marquee)# Gimbal Purple
    led_ctrl.set_bottom_led(rm_define.armor_bottom_all,128,0,128, rm_define.effect_always_on)# Chassis Pu
rple
    chassis_ctrl.move_with_distance(45,0.5)# move diagonally right 0.5 meters
    chassis_ctrl.move_with_distance(-45,0.5)# move diagonally left 0.5 meters
    chassis_ctrl.move_with_distance(135,0.5)# move back right diagonally 0.5m
    chassis_ctrl.move_with_distance(-135,0.5)# move back left diagonally 0.5m
    time.sleep(0.5)

    led_ctrl.gun_led_on()
# Spin gimbal 90 degrees left and shine light.
    gimbal_ctrl.yaw_ctrl(-90)
    gimbal_ctrl.yaw_ctrl(0)
    time.sleep(1)
    led_ctrl.gun_led_off()

# Rotate chassis 360 degrees
    chassis_ctrl.rotate_with_degree(rm_define.clockwise,180)
    chassis_ctrl.rotate_with_degree(rm_define.clockwise,180)

# Reset gimbal
```

```
    gimbal_ctrl.yaw_ctrl(0)

################################################################################
######################
def start():
    robot_ctrl.set_mode(rm_define.robot_mode_free)# set mode
    gimbal_ctrl.set_rotate_speed(80)# set rotation speed of the gimbal
    chassis_ctrl.set_trans_speed(0.8)# set movement speed of the robot in m per sec
    chassis_ctrl.set_rotate_speed(80)# set rotation speed of the chassis

# First section - maze
    do_maze()

# At reset point B - already slept

# Goes to front of room one - moves 6.70m
    chassis_ctrl.move_with_distance(0,5)# Move forward 5 meters
    chassis_ctrl.move_with_distance(0,1.70)# Move forward 1.70
# Is in front of Room C
    chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)# Rotate the chassis 90 degrees to the right
    gimbal_ctrl.yaw_ctrl(0)
    gimbal_ctrl.pitch_ctrl(20)
    gimbal_ctrl.pitch_ctrl(-20)
    gimbal_ctrl.pitch_ctrl(0)
    execute_room_C()

# At reset point D
    gimbal_ctrl.yaw_ctrl(0)
    gimbal_ctrl.pitch_ctrl(0)
    time.sleep(8)

    chassis_ctrl.move_with_distance(0,4.86)# Goes to front of room 225 - point E

    chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)# Rotate the chassis 90 degrees to the right:
    gimbal_ctrl.yaw_ctrl(0)
    gimbal_ctrl.pitch_ctrl(20)
    gimbal_ctrl.pitch_ctrl(-20)
    gimbal_ctrl.pitch_ctrl(0)
    execute_room_E()# ends at reset point F
    gimbal_ctrl.yaw_ctrl(0)
    gimbal_ctrl.pitch_ctrl(0)
    time.sleep(8)

    scan_for_marker()

    gimbal_ctrl.yaw_ctrl(0)
    gimbal_ctrl.pitch_ctrl(0)
```

```
    chassis_ctrl.move_with_distance(0,4.78)# Moves to front of room 224B - ### CHECK MEASUREMENT, sh
ould be 4.70
    chassis_ctrl.rotate_with_degree(rm_define.clockwise,90)# Rotates 90 degrees
    gimbal_ctrl.yaw_ctrl(0)
    gimbal_ctrl.pitch_ctrl(20)
    gimbal_ctrl.pitch_ctrl(-20)
    gimbal_ctrl.pitch_ctrl(0)
# Now facing point G
    execute_room_G()

# Is at reset point H
    chassis_ctrl.rotate_with_degree(rm_define.clockwise,180)
    gimbal_ctrl.yaw_ctrl(0)
    gimbal_ctrl.pitch_ctrl(0)
    chassis_ctrl.set_trans_speed(1.5)# set movement speed of the robot in m per sec
    time.sleep(8)
    media_ctrl.play_sound(rm_define.media_sound_count_down,wait_for_complete_flag=True)
## Sound clip here ###

# Moves 1030cm
    chassis_ctrl.move_with_distance(0,5)
    chassis_ctrl.move_with_distance(0,5)
    chassis_ctrl.move_with_distance(0,0.30)
# Reset point F
    time.sleep(8)
    media_ctrl.play_sound(rm_define.media_sound_count_down,wait_for_complete_flag=True)

# Moves 879cm
    chassis_ctrl.move_with_distance(0,5)
    chassis_ctrl.move_with_distance(0,3.79)

# Reset point D #### Do dance before reset.

######## DO DANCE HERE ############
    media_ctrl.play_sound(rm_define.media_custom_audio_0,wait_for_complete_flag=False)
    do_dance()
    gimbal_ctrl.yaw_ctrl(0)
    gimbal_ctrl.pitch_ctrl(0)
# After dance, reset.
    time.sleep(8)
    media_ctrl.play_sound(rm_define.media_sound_count_down,wait_for_complete_flag=True)
# Moves 1190cm ######## EDIT
    chassis_ctrl.move_with_distance(0,5)
    chassis_ctrl.move_with_distance(0,5)
    chassis_ctrl.move_with_distance(0,1.80)##### Check this measurement, should be 1.55 #########

# Should be at reset point B
    time.sleep(8)
```

```
    media_ctrl.play_sound(rm_define.media_sound_count_down,wait_for_complete_flag=True)

# Moves 863cm
    chassis_ctrl.move_with_distance(0,5)
    chassis_ctrl.move_with_distance(0,3.53)

# Should be back at the start!! WOO HOO!!
```