Name: ____________________

Assessment

Walker Detect

1) What is an Ultrasonic Range Finder used for in Walker's construction?
    a) To control Walker's speed
    b) To sense the distance to nearby objects
    c) To measure the weight of Walker
    d) To power Walker's motors

2) What does the term "GND" refer to in the context of connecting the hardware?
    a) Ground
    b) Signal
    c) Voltage
    d) No Connection

3) Which pin on the Raspberry Pi Pico should the SIG wire of the Ultrasonic Range Finder connect to?
    a) GP5
    b) GP22
    c) 3V3
    d) GP0

4) What is the purpose of the variable is_not_blocked in the code?
    a) To check if the motors are functioning
    b) To measure the speed of Walker
    c) To set the angle of the servos
    d) To determine if Walker should stop or continue walking

5) Why is it important to loop over the testing and walking steps in the code?
    a) To ensure the motors keep running
    b) To change the colors of Walker's lights repeatedly
    c) To prevent Walker from walking onto the judges
    d) To increase Walker's walking speed

6) Imagine Walker is too close to the judges and needs to back away. Describe how you could modify the code to make Walker walk backward. What specific changes would you make to the angles and the sequence of movements?

Name: ____________________

7) How could you apply sensors to detect obstacles to improve safety features in self-driving cars? Provide a detailed explanation.

8) The Python code provided makes Walker move forward as long as the distance sensor detects that it is more than 8 centimeters away from an obstacle. Can you analyze this code and change it so that Walker moves backward instead of forward when it detects an obstacle closer than 8 centimeters? Write the new function for moving backward and update the code to use this function.

Name: ____________________

Answer Key
Walker Detect

1) B - To sense the distance to nearby objects

2) A - Ground

3) B - GP22

4) D - To determine if Walker should stop or continue walking

5) C - To prevent Walker from walking into the judges

6) *Example:* To make Walker walk backward, I would reverse the sequence of either the front legs or the back legs. I would also adjust the initial angles of the motors to 90° to ensure they start from a neutral position. This sequence would need to be coded into a loop, similar to the forward walking loop, to continually check the distance and move Walker backward if it gets too close to the judges.

7) *Example:* Using sensors to detect obstacles can greatly improve safety features in self-driving cars. In self-driving cars, sensors like ultrasonic range finders, LIDAR, radar, and cameras can detect obstacles, other vehicles, pedestrians, and road conditions. The data from these sensors can be processed in real-time to make decisions about accelerating, braking, and steering. For example, if an obstacle is detected, the car's system can automatically slow down or stop to avoid a collision. Additionally, the car can be programmed to follow specific routes and avoid hazards by continuously monitoring the environment and adjusting its path. This technology ensures the car can navigate safely and efficiently without human intervention.

8) Example:

```
## ---- Imports ---- ##
import time
import board
from piper_blockly import *

## ---- Definitions ---- ##
is_not_blocked = None
GP22 = piperDistanceSensorPin(board.GP22, "GP22")

try:
  set_digital_view(True)
except:
  pass
```

PIPER

Name: ___________________

```
def distance_range(_value):
  if (_value == None):
    return 520
  else:
    return _value

GP1 = piperServoPin(board.GP1, "GP1")
GP0 = piperServoPin(board.GP0, "GP0")

# Describe this function...
def step_forward():
  global is_not_blocked
  GP1.setServoAngle(60)
  time.sleep(0.2)
  GP0.setServoAngle(120)
  time.sleep(0.2)
  GP1.setServoAngle(120)
  time.sleep(0.2)
  GP0.setServoAngle(60)
  time.sleep(0.2)

# Describe this function...
def step_backward():
  global is_not_blocked
  GP0.setServoAngle(60)
  time.sleep(0.2)
  GP1.setServoAngle(120)
  time.sleep(0.2)
  GP1.setServoAngle(120)
  time.sleep(0.2)
  GP0.setServoAngle(60)
  time.sleep(0.2)

# Describe this function...
def set_up_motors():
  global is_not_blocked
  GP0.setServoAngle(90)
  GP1.setServoAngle(90)

## ---- Code ---- ##
set_up_motors()
while True:
  is_not_blocked = (distance_range(GP22.readDistanceSensor())) > 8
  if is_not_blocked:
    step_forward()
  else:
    step_backward()

  time.sleep(0.2)
```

Name: ____________________

Explanation:
I added the `step_backward` function to make Walker move backward. I also updated the main loop to call `step_backward` when the obstacle is detected within 8 centimeters.