6.3 S-R-A loop

S-R-A loop is repeating process where:

1. Seasoning,
2. Reasoning and
3. Acting

is involved during the procedure of controlling the robot. This is the most important part of software in robotics. Remember the `autonomous` control is `ability to perform intended tasks based on current state` **and** `sensing, without human intervention`.

The S-R-A loop is a common design pattern in robotics. It refers to the process of using sensors to gather information about the environment, processing the information to determine an appropriate response, and then executing the response using actuators.

Here is an pseudo example of how the S-R-A loop could be implemented in C++:

```cpp
while (true) {
  // 1. Sense the environment using sensors
  sensor_data = gatherSensorData();

  // 2. Process the sensor data to determine an appropriate response
  response = processSensorData(sensor_data);

  // 3. Execute the response using actuators
  executeResponse(response);
}
```

In this example, the `gatherSensorData` function is used to gather data from the robot's sensors, the `processSensorData` function is used to determine an appropriate response based on the sensor data, and the `executeResponse` function is used to execute the response using the robot's actuators. The loop is executed continuously, allowing the robot to constantly sense and respond to its environment.

6.3.1 Tasks:

1. Using the S-R-A loop technique you should write the program in particular order:

   1. Check the sensor. IF the bumper ...
   2. ... Is pressed the robot has to stop/go back/turn.
   3. ... Is not pressed the robot can drive forward.

   Test the prog. 1 and **find out why the robot does not stop.** (Such mistake is quite often - can you fix it):

Program 1: SRA Loop.

```
1   #include "RobotMovingFunctions.h"
2   const int BUMPER_PIN = A0;
3   void setup()
4   {
5     setIOpins();
6     pinMode(BUMPER_PIN, INPUT);
7
8     bool bumperIsPressed = digitalRead(BUMPER_PIN);
9     if ( bumperIsPressed )
10    {
11      stopTheRobot();
12    }
13    else
14    {
15      moveForward();
16    }
17  }
18  void loop()
19  {
20  }
```

2. Hint for fixing the prog. 1: *S-R-A must be a loop function!*

3. Write a program to drive the robot around the class and avoid the obstacles.

6.3.2 Questions:

1. What for `S-A-R loop` stands for?
2. Mark all three basic S-A-R processes in previous code example.
3. Can the `line` 7 of the prog. 1 be written outside of `loop()` function? What would happened if so?

6.3.3 Summary:

6.3.3.1 **<++>** <++>

6.3.4 Issues:

6.3.4.1 **<++>** <++>