0.1 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`.

0.1.1 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.

```
#include "RobotMovingFunctions.h"
const int BUMPER_PIN = A0;
void setup()
{
  setIOpins();
  pinMode(BUMPER_PIN, INPUT);

  bool bumperIsPressed = digitalRead(BUMPER_PIN);
  if ( bumperIsPressed )
  {
    stopTheRobot();
  }
  else
  {
    moveForward();
  }
}
void loop()
{
}
```

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.

0.1.2 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?

0.1.3 Summary:

0.1.3.1 <++>   <++>

0.1.4 Issues:

0.1.4.1 <++>   <++>