

Name: _____

NQC

Basic output control commands:

- `On(outputs);` — Turns **ON** the specified outputs
- `Off(outputs);` — Turns **OFF** the specified outputs
- `Fwd(outputs);` — Sets the direction to **FORWARD** for the specified outputs
- `Rev(outputs);` — Sets the direction to **REVERSE** for the specified outputs

`outputs` parameter can be `OUT_A`, `OUT_B`, `OUT_C`, or a combination of these outputs added together with the '+' sign, such as `OUT_A+OUT_C`. For example:

```
On(OUT_A+OUT_C);
```

Will turn **on** outputs A and C.

We will also need a Wait command, it looks like this:

- `Wait(time);` — Tells the RCX to not continue with the next command until the `time` has passed.

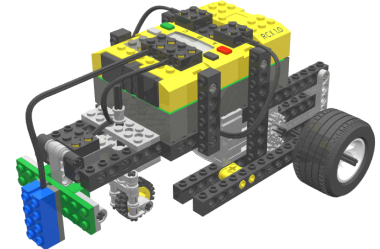
The `time` parameter is in hundredths of a second. For example, if you want to wait for 1 second you would use the command:

```
Wait(100);
```

Note that all these commands ended with a semi-colon. The semi-colon marks the end of the command for the NQC compiler.

All NQC programs must have a `task main`. The `task main` is where you put the commands.

```
task main ()  
{  
    commands  
}
```



Assuming you have a robot like the GT Scooter with the left motor connected to output A and the right motor connected to output C.

Program sample	Description
<pre>task main () { On(OUT_A+OUT_C); Wait(200); Off(OUT_A+OUT_C); }</pre>	Drive forward for 2 seconds and then stop.
<pre>task main () { On(OUT_A+OUT_C); Wait(100); Rev(OUT_A+OUT_C); Wait(100); Off(OUT_A+OUT_C); }</pre>	

<pre> task main() { On(OUT_A+OUT_C); Wait(200); Off(OUT_A); Wait(100); Off(OUT_A+OUT_C); } </pre>	
<pre> task main() { repeat(4) { On(OUT_A+OUT_C); Wait(200); Off(OUT_A); Wait(100); } Off(OUT_A+OUT_C); } </pre>	<p>Now we introduce the <code>repeat(n)</code> statement. The commands inside the curly-brackets after the <code>repeat</code> are repeated n times. What will this sample do?</p>
<pre> task main() { while(true) { On(OUT_A+OUT_C); Wait(200); Rev(OUT_A); Wait(100); Fwd(OUT_A); } Off(OUT_A+OUT_C); } </pre>	<p><code>while(true)</code> is similar to <code>repeat(n)</code> except the commands inside the curly-brackets are repeated forever (as long as <code>true</code> is true).</p>

The examples below have one or more **errors**; can you see what is wrong?

<pre> task main() On(OUT_A+OUT_C); Wait(200); Off(OUT_A); Wait(100); Off(OUT_A+OUT_C); </pre>	<p>Missing the curly brackets around the commands.</p>
<pre> task main() { On(OUT_A+OUT_C) Off(OUT_A+OUT_C) } </pre>	
<pre> task main() { on(OUT_A+OUT_D); wait(250); off(OUT_A+OUT_D); } </pre>	

Homework: Write a program that will drive the robot in some interesting way and describe the expected behavior.