Algorithm Question

A program has been written to obtain 3 random virtual dice rolls:

```
DiceRoll(1) = Random Number 1-6
DiceRoll(2) = Random Number 1-6
DiceRoll(3) = Random Number 1-6
DISPLAY DiceRoll(1)
DISPLAY DiceRoll(2)
DISPLAY DiceRoll(3)
```

Rewrite this algorithm in pseudocode so that it makes use of iteration.

[4 marks]
A program has been written to obtain 3 random virtual dice rolls:

\[
\begin{align*}
\text{DiceRoll}(1) &= \text{Random Number 1-6} \\
\text{DiceRoll}(2) &= \text{Random Number 1-6} \\
\text{DiceRoll}(3) &= \text{Random Number 1-6} \\
\text{DISPLAY DiceRoll}(1) \\
\text{DISPLAY DiceRoll}(2) \\
\text{DISPLAY DiceRoll}(3)
\end{align*}
\]

Rewrite this algorithm in pseudocode so that it makes use of iteration

[4 marks]

***There are always different ways to solve a problem. This algorithm is just an example. What is important is that the logic is correct!***

**LOGIC:**

- Use of a FOR or WHILE loop
  - If a FOR loop used: must loop for 3 iterations
  - If a WHILE loop used: condition must allow for only 3 iterations (may require a counter being incremented / decremented)
- Within each loop:
  - Creation of random number
  - OUTPUT of dice generated

**EXAMPLE ALGORITHM:**

<table>
<thead>
<tr>
<th>FOR LOOP EXAMPLE:</th>
<th>WHILE LOOP EXAMPLE:</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOR counter 1 – 3:</td>
<td>counter = 3</td>
</tr>
<tr>
<td>\hspace{1cm} dice_number = Random Number 1-6</td>
<td>WHILE counter !=0:</td>
</tr>
<tr>
<td>\hspace{1cm} OUTPUT dice_number</td>
<td>\hspace{1cm} dice_number = Random Number 1-6</td>
</tr>
<tr>
<td></td>
<td>\hspace{1cm} OUTPUT dice_number</td>
</tr>
<tr>
<td></td>
<td>\hspace{1cm} counter = counter - 1</td>
</tr>
</tbody>
</table>