**Specify a set of test cases with expected inputs for testing the program.**

Testing is an important part of getting a program to do what it was intended to do. Test cases are input or situations your program may need be able to operate under. For example let’s think about the climb the wall algorithm.

You were asked to: to build a wall about 3 blocks wide and at least 1 block high and make a program for the agent to climb over the wall.

You were given the pseudocode for a mod that will instruct the agent to climb over a wall. One limitation was that the agent needed to be placed 5 or less blocks away from the wall.

After writing the pseudocode the user needs to test whether it works. Here is one test case.

**Tests 1**

**Expected test cases are ones that the program can expect. Situations that are usual for the program.**

|  |  |
| --- | --- |
| **Test Description** | **Expected input/output** |
| When climb command is entered the agent should teleport to players location | Agent should teleport to player and face the wall. |
| When climb command entered agent should climb the wall |  |
|  |  |

**TASK**

Write in the expected input and output.

Write another test case to test.

**Tests 2**

**Boundary test cases are ones that push the limits of the programs operation. Such as placing a hole in front of the agent.**

|  |  |
| --- | --- |
| **Test Description** | **Boundary input/output** |
| Put agent 5 blocks from the wall |  |
| Change wall to a different height |  |
|  |  |

Write in the boundary input and output.

Write another test case to test.

**Tests 3**

**Invalid test cases are ones that test invalid input and output.**

|  |  |
| --- | --- |
| **Test Description** | **Invalid input/output** |
| Put agent 6 blocks from the wall |  |
| Make the wall 2 blocks thick |  |
|  |  |

Write in the boundary input and output.

Write another test case to test.