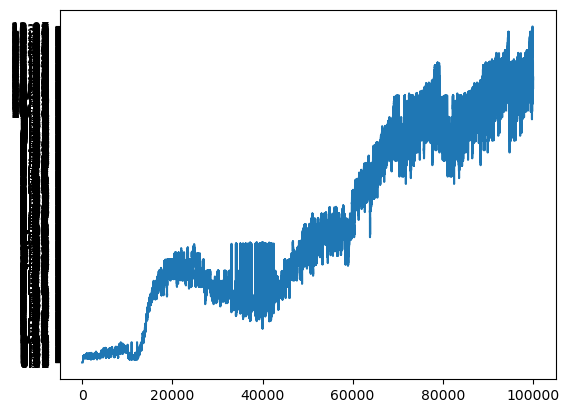
**REPORT**

**Test results:**

The graph shows the plot of episode (x-axis) vs rewards (y-axis)



**Experiments:**

* **DQN architecture**:

Initially my neural architecture looked like:

**2 layers of Convolution layers followed by a ReLU function,** but surprissingly that didn’t help much so I used the architecture from paper only.

* I experimented with **training every 10th frame** while training, this was 1 tip which I found people use (skipping frames while fetching from buffer). But that didn’t make much improvements so I didn’t complicate much.
* Another hyperparameter which I tuned while training was **decaying epsilon**, which I did linearly before but then used exponential decay.

In this too, I experimented with 2 formulaes as follows:

* + EPSILON = EPS\_START + (EPS\_START - EPS\_END) \* math.exp(-(iEpisode+1) \* EPS\_DECAY)
  + EPSILON = EPS\_END + (EPS\_START - EPS\_END) \* math.exp(-1. \* (iEpisode+1) /EPS\_DECAY)

First one gave me better results.

**Mean Reward on test data (100 episodes)**: 48.3