PART A Conceptual Study – What is AI, how to use it in Game building?   [10 points]
Explain the difference between Strong AI and Weak AI.   In the context of <u>any computer game</u>, how are Weak AI and Strong AI applicable for the implementation and playing of a game? Explain identifying at least 2 algorithms each (no need to explain them).

Weak AIs are AIs which are focussed on one narrow task. For example, AlphaGo is an example of a weak AI. It has one specific task which it does very well. Strong AI on the other hand is AI system that can think and has cross domain capabilities like humans.In Strong Artificial Intelligence, the machine can actually think and perform tasks on its own just like a human being. In Weak Artificial Intelligence the devices cannot follow these tasks on their own but are made to look intelligent.

In the context of computer games, Artificial Intelligence plays a huge role to make the game more appealing. A computer game which is not artificially intelligent may not be fun to play since they might exhibit patterns which might be easy to spot and hence can be easily beaten. An artificially intelligent game, ideally, will not exhibit any pattern and hence is more difficult to beat which makes to more interesting to play. For example, games like chess or FIFA use a weak AI to imitate human moves. A strong AI hasn't been built yet. However if we ever create an AI which thinks and acts like a human and is good at multiple tasks, it would be termed as a strong AI.

PART B

I have submitted a Jupiter notebook named 'Assignment1_partB.ipynb' which contains the solution to part B of the homework. The output window will contain two rectangles one of which is stable and the other one can be controlled using the arrow keys. When the two rectangles collide, a text shows up saying that the rectangles have collided. Also, theres a text at the bottom right which indicates the distance between the centers of the two rectangles at any time. This is to demonstrate the distance function.

PART C

The file 'game_of_life.py' contains the code to part c. You can run the code using the command

> python3 game_of_life.py

This will start the simulation with a random board. You can also pass a command line argument to start the board with a given pattern as mentioned in point 4 of the question. To start the simulation with the glider pattern, the command is :

>python3 game_of_life.py —pattern glider