PART 3(Linear Programming)
Team 52 (Kartik[2019101060],Ayush[2019111026])

Q.1) How A matrix was formed?
=> A matrix'x dimension => NUM_STATES*NUM_COLUMNS
NUM_COLUMNS = sigma(possible actions in each state)

NUM_STATES = 5(Five positions for IJ)*3(Three values for Materials)*4(Four values for Arrows)*2(Two states for Monster)*5(Five values for health MM)

So now iterating for each state, i will iterate over the possible actions in that state
Now, i am with a state and an action that is to be performed in that state.
Now just as the analogy was made in class that it is like the flow problem.
Flow out is considered as positive and flow in is considered as negative.
So what we now do is we assign the value of current state and current action as +1.0 (with probability 1, everything moves out). Now we perform the action that was selected. On performing the action there are certain probabilities of each action to success and fail along with an inbuilt probability for MM to be in dormant state or go to ready state(if in dormant initially) or to attack or not attack(if initially in ready state).
Now the different probabilities are basically the probability with which it goes in that state. Which means out of 1, there is this much chance that i end up in that state. That is out of 1 products this many products go in this state. That is this much amount flows into that state. Now since it flows in therefore is taken as negative. Since now i have the state in which i will land up and i have the action so i add a negative amount of probability to that state,action pair.
In this way the A matrix was formed. This matrix basically stores the transition probabilities, a kind of. Like how much change will occur to a state by a following action that is captured in this matrix.

Q.2) Explain the procedure of finding the policy and explain the results?
=> To find the policy that is to suggest an action for each posible state.
Since we have solved out LP which will result in the formation of X vector which basically maximizes the avg reward that we will get till the end of the game. In this X we have the probabilities by which each action is

selected in each state. Now lets say we have 4 actions possible for a state. Now what LP solving would have done is, it would have given me some probabilities for each action which is basically the probability by which each action is picked. Now since i have the probabilities and i am considering the sigma of R^iX^i and maximising it, now i should check the action which makes the most contribution that action will be selected. Therefore i should select the max(X^i) for all i for a particular state. And the action for which we have found the max value will be selected action for that state.

The current policy that we have got by assuming the start state to be [“C”,2,3,”R”,100] says to move to NORTH by choosing UP action and then opting to craft if i can and i have number of arrows such that 25*num_arrows < MM_health then this is preferred. If my arrows are not according to the condition mentioned then the policy says to go down to center and attack MM. In case MM is in ready state then the policy says to stay on north until MM becomes DORMANT.

Q.3) Can there be multiple policies?

=> Yes, off course there can be multiple policies. One of the obvious things that one can do to extract a different policy is to change my initial state. As changing initial state will lead to change of alpha vector that is constraints are changed now that results in a new policy. Moreover if we wanna visualise it then one can say from the reasoning that this whole thing is given an analogy to the flow thing that the amount that flows in is same as the amount that flows out. And the different probabilities can be viewed as different pipes of different crossection. Now since my initial point therefore the environment that is the surrounding pipes are changed therefore the results will change now. One can also change the policies by changing the A matrix that is change the transition probabilities. One can again relate it to that now i am changing the size of the different pipes attached to my starting position. On the formulation of LP from MDP we define a set of constraints from the Bellman equation that tell us that all V_i’s are greator than equal to some constant and at the same time we should find the minimum such value that satisfies the constraints. Now since we are maximizing the sigma(R_iV_i) , so as far as we think that on changing R my result will not change as since all of V_i’s were greator than equal to zero and have to be greator than some value so to maximize the product they have to attain their maxima without fearing about the coefficient they have. But on observing the results by applying the change R thing in the code we found that this is not the case. The actual reason might because of the fact that V is not exactly linear in its variation. Other is that the combination of V and R does make an effect in the calculation. So on changing any of the following R,A,alpha one can change the policy suggested by LP.