Q2

According to the question, we already know that we know the exactly R_a, P_a and S_a number and $R_a + P_a + S_a = N$, same as R_b, P_b and S_b also $R_b + P_b + S_b = N$, however, we don't need to throw in the consecutive order.

The question is how should we play to maximise the number of points, which can be convert to that to minimise the loss round, as the total round is N, so win round plus loss round plus tie round is equal to N. Minimise loss round is equal to maximise the win round plus tie round.

Assume we have two arrays, array A with $[R_a,\ P_a,\ S_a]$ and array B with $[R_b,\ P_b,\ S_b]$, so the maximum win round is $win\_round = min(R_a, P_b) + min(P_a, S_b) + min(S_a, R_b)$, Denote $w_0 = min(R_a, P_b)$, $w_1 = min(P_a, S_b)$ and $w_2 = min(S_a, R_b)$, we throw Paper, Scissor and Rock in w_0, w_1 and w_2 rounds respectively, after that we can derive new array A_1 with $[R_a - w_0,\ P_a - w_1,\ S_a - w_2] = [R_a^{'},\ P_a^{'},\ S_a^{'}]$ and new array B_1 with $[R_b - w_2,\ P_b - w_0,\ S_b - w_1] = [R_b^{'},\ P_b^{'},\ S_b^{'}]$, then we can compute the maximum tie rounds is $tie\_round = min(R_a^{'}, R_b^{'}) + min(P_a^{'}, P_b^{'}) + min(S_a^{'}, S_b^{'})$ and same as above, for number of array A_1 and B_1, minus the minimum value at the same index, we finally get new array A_2 $[R_a^{''},\ P_a^{''},\ S_a^{''}]$ and new array B_2 $[R_b^{''},\ P_b^{''},\ S_b^{''}]$, it is clearly that some of those values are 0. Finally the rest of number of array A_2 and B_2, no matter how to throw are all loss so the loss round which is $loss\_round = sum(A_2) = sum(B_2)$ so the total score that by doing this method is $win\_round - loss\_round$ which is also the maximum score.

Optimality: assume we have another optimal solution with the same scores but not the same maximum rounds of win_round plus tie_round which is less than our algorithm. If so, this solution must have more loss rounds as the $win\_round + tie\_round + loss\_round = N$, in this situation, the scores must less than our's method, which is a contradictory. So my algorithm shows above can give the optimal solution.