Yuan Gao z5239220 Q2

Rock. Paper. Scissors. The rules are simple. The game is contested by two people over N rounds. During each round, you and your opponent simultaneously throw either Rock, Paper or Scissors. Rock beats Scissors, Scissors beats Paper, and Paper beats Rock. If your throw beats your opponent's, you gain one point. Conversely, if their throw beats yours, you lose one point. Your opponent is very predictable. You know that they will throw Rock in the first R_a rounds, throw Paper in the next P_a rounds, then finally throw Scissors in the last S_a rounds, where $R_a + P_a + S_a = N$. You have to throw Rock in R_b rounds, Paper in P_b rounds, and Scissors in S_b rounds, where $R_b + P_b + S_b = N$. However, as you are an experienced player, you may throw these in any order you like. At the beginning of the game, you start with 0 points. How should you play to maximise the number of points you can finish with?

According to the game rules, there are N rounds and $N = win + draw + lose$.

Win one round, player gain one point

Draw one round, player gain zero point

Lose one round, player lose one point

Hence, when the number of win rounds increase and the number of $lose$ rounds decrease, the player can get the increasing number of points.

Therefore, there are three step strategy to get the maximise the number of points:

1. Win as much as possible
2. Try not to lose (draw)
3. Deduct points(lose)

We create two array (opponent[3], player[3]), one stores the number of R_a, P_a, S_a, others stores the number of R_b, P_b, S_b

Code:

```cpp
/*offset the number of RPS*/
void after_terget(int & opponent, int & player){
    if(opponent > player){
        opponent = opponent - player;
        player = 0;
    }else{
        player = player - opponent;
        opponent = 0;
    }
}

/*conut the max get point*/
int RPS_game(vector<int> opponent, vector<int> player){
    int res;
    //opponent[0] = Ra, opponent[1] = Pa, opponent[2] = Sa
    //player[0] = Rb, player[1] = Pb, player[2] = Sb

    //win
    //win = min(Ra, Pb) + min(Pa, Sb) + min(Sa, Rb)
    int win = min(opponent[0], player[1]) + min(opponent[1], player[2]) + min(opponent[2], player[0]);

    after_terget(opponent[0], player[1]);
    after_terget(opponent[1], player[2]);
    after_terget(opponent[2], player[0]);

    //draw
    int draw = 0;
    after_terget(opponent[0], player[0]);
    after_terget(opponent[1], player[1]);
    after_terget(opponent[2], player[2]);

    //lose: remaind opponent is player lose point
    int lose = opponent[0] + opponent[1] + opponent[2];

    //get result
    res = win + draw - lose;

    return res;
}
```