PLL Assignment 3
Shubham Kumar 170101064

1. A) Isince we are using list to represent set here so i used length function of list to get if a set is empty or not.
   B) First i subtracted Set A from Set B and then added just taken all the elments from set B and the subtracted setA
   Names of functions that i have created
   is_null setA – To check if setA is empty or not
   set_union set1 set2 -> Give union of set1 and set2
   set_intersection set1 set2 -> intersection of set1 and set2
   set_subtraction set1 set2 ->set1-set2
   set_addition set1 set2 -> set1+set2

2. Generated Random matches based on the permutaions. What i did was taken a permutation of the list of acronyms and then distribued matches among them. Team at odd indices in the list are declared as first team and even one are declared as second team.
3. Algorithm :

```
Initialization -> Initialize all the counts of rooms such as
                      -> KitchenCount = Ceiling(BedrromCount/3)
                      -> Bathroom Count = BedrromCount+1
                      ->GardenCount = 1
                      ->Balcony Count = 1
                      -> Set minimum and maximum dimensions of each type of
                      room/Garden/Balcony
                      -> Initialize with room_type hall and current dimension of hall to mnimum
Recursive Solution -> Now at each step we have two choices
                              -> Either increase dimensions of current room_type/Garden/
                              Balcony.
                              -> Move to next type of room.
                         Base Case of recursion-> When we have dimension at maximum for
                         given room and all type of rooms are fullfilled.
                         -> Choose best design from recursion based on the area covered by
                         two designs. (unused space should be minimum.)
Termination – When we have all the conditions fullfilled
                      -> total_area <= size given initially
                      -> kitchen_area <= hall_area
                      -> kitchen_area<=bedroom_area
                      -> bathroom_area<=kitchen_area
                      -> If these conditions satisfied then print solution
                         else print " No design can be generated".
```

I am assuming that all the components of given type have equal dimensions. i.e. All bedrroms are of same dimension , all bathrooms are of same dimesions. And i am increasing both dimesions of each component by same amount when i am increasing them.
I used 7 function to derive the algorithm. 1 is to initialize, 1 for recursion, 1 for gettng area of given dimensions, 1 for getting total_area of designed room/fgarden types, 1 for choosing best design, 1 for validating design and 1 for printing the solution. All of them are pure since haskell is a purely functional language.

Lazy Feature – Lazy evaluation means nothing is evaluated unitll its needed. In assignmemts we can use this property for better performances. In Solution 2 i have generated random matches before they were already used in the matches. When we are using reciursion in question 3 lazy feature can be used to reduce some extra computancy its like storing solution and Dynamic programming pproach.

In the haskell all functions are mathematical functions means they will give same output for given input. In question3 this was useful as we have created soltions for given input only once since there are no side effects so output for given input will be same in haskell so this make performance better