REPORT

Main predicates:

1. **straight\_chain\_alkane: it generates a straight chain alkane without any branching carb by doing a case that return a carb of hs if the number of alkane contains only 1 carbon and if it contains more than 1 it will do it recursively to generate the straight chain.**
2. **branched\_alkane: it takes N which is the number of carbons and generate all possibilities of branched alkane that can be given and handle cases that no branches at the most left carb and the most right carb ,and also handle the case that no chain is longer than the straight one.**
3. **isomers: it takes N which is the number of carbons and call branched\_alkanes to generate all possibilities of branches and use the predefined predicate setof to collect all results in a list and remove duplicates and call mirror2 which remove mirrors lists with mirror in carb it self and mirror3 which remove any list that have a mirror.**

**Helper predicates:**

1. **target\_name: it takes the number of carbons and generate a name of branched , example: N=2 Target=c2h5.**
2. **add\_branch: it takes N which is the number of branched alkane and carb , first it checks if top is h , it call target\_name to generate the name for the given N and and return the new carb ,and if top is not h it will use the second predicate to generate the same carb as the first one after checking that top is not h ,then return carb.**
3. **helper:- it takes a number and generate a list of ones with sum of given number.**
4. **split:- it takes a list and return each element from the list recursively .**
5. **last2:- it takes a list and return the last two elements in it.**
6. **Add:- it adds two numbers and return the sum.**
7. **Delete1:-it deletes the last element from a list.**
8. **Remove2:- it takes a list and two elements and remove the both elements from the list and return the updated list.**
9. **Insert: it takes a list and generate a list of the same elements except the last two elements are added and return a number which is appended to the generated list ,and it sort the generate list and recall insert recursively to backtrack and get all results that for repeation The second insert do the same thing but it reverse the generated list from append and recall insert recursively to check that there is no repeation,**
10. **Break\_down: it takes a number and call insert to get all possibilities of lists that gives the sum of this number , and call it into setof to collect results in list and remove duplicates then split to give all possibilities of sum of this element. Break\_down is used to generate all possible numbers of branches from the given number of carbon that need to be branched .**
11. **Track : it takes two numbers and generate all differences between this two numbers then recall track again with incrementing the smallest number by 1 to generate other differences till reaching one, and we used it to generate all possible combinations from a total number of carbons and generate the number of carbons to be in straight chain and branched carbons .**
12. **Counter: it takes a number and increment it by 1.**
13. **Split2: it takes a list and gives elements in it as a single element recursively same like split but it also gives the index of each element .**
14. **Replace: it takes a list and element and index and it replace the element in the list with index equal to given index by given element and return the new list.**
15. **Hlp2: it is called in main predicate branched\_alkane, and it takes the straight chane and number of the first answer from breaking down the number of carbon that needs to be branched that is given from track, it split the straight chain by split2 to get split and get index, and it calls hlp3 that returns the the list of branched alkane, this call hlp2 recursively to backtrack and get all answers of trying the remaining elements in list that is generated from break\_down.**
16. **Hlp3:it takes straight chain and the splited element from straight chain and it calls add branch to add branch to carbon with the number splited from list taken from break down ,then it replace it with element in the given index and then it check the output list from replacing predicate that there is no chain longer than straight chain by checking lengths .**
17. **Get\_number: it a name like c2h5 and return 2 by using atomic concat predicate and atom number predicate which changes string to number.**
18. **Mirror: it takes a carb and it reverse it internally.**
19. **Mirror2: it takes a list that contains all branched\_alkanes, and call mirror that is used to revesre the carb intrenaly , then it deletes duplicates from bramched alkans , which means that we remove all mirrors of carbons then we call it again recursively to make the same process to all lists in branched alkane.**
20. **Mirror3: it takes the list from mirror2 and remove mirrors of lists only and recursively do the same for all lists in given list.**