Assignment 01 – Question C

Firstly I explain how can understand the question for me.

I read the Assignment 01 Question C and I understood the problem is use for business company for Calculate "Best Combination" from each employees for do jobs of the company using Trusted Value. There are several jobs in that company and each employees work in that jobs. An employee can participate many jobs. So that's a point of the question. After that system should input minimum amount of eligibility (M). That's mean if I insert 2 for M value then we selected employee who works job count greater than or equal that M value. Others are rejected (M value less than for that M value). After that we get selected employee and create combination between that employees. For example if employees are (A, B, C) then combinations are AB, AC, BC. After that we check that each combination employees how many time do the same job. And also count that value and compare that value with inputed minimum amount of eligibility (M). If it's equal or greater than Let selected that combination and others are rejected. After that selected employees added to another list and combination that employees with 3 combination. Like that order we generate all combination values that greater than M value and marked their M values. After that Add that Get largest combination value(s) and get second largest value and if second largest values in largest value and it's M value less than second largest values then Add that value(s) to PTS list and if second largest value not in largest value then Add that second largest value to PTS list. Like that order we do the same process for all combination list value and filter that to PTS list. Now PTS list is done. After that get largest PTS combination value to smallest and calculate formula using each PTS value using Trust Value. We should input T value and compare with TV value. We calculate TV using that PTS value. For example PTS value is ABCD then A->BCD formula values TV is [M(ABCD)/M(A)]*100 like that calculate TV of difference combination of each combination value (like B->ACD , C->ABD , D->ABC , AB->CD,.....etc) . If TV value Greater than or equal Trust value(T) then that sub formula will added to Formula list. So these are the steps for solve this problem.

Now I explain how I used data structure to solve that problem.

I used Python programming for solve this problem. Firstly user should input Number of employee of that company. The number of jobs are not input . Firstly asked job1 employee and we can input for that as 1,2,3,4 scenarios. After enter that we ask Add more? If it's 'Y' system will ask job2 employee and like that this steps are repeat. If it's 'N' system will stop the add jobs employees. So number of jobs are depend on this method .After that I use dictionary data structure for store jobs and employees of that jobs. Job number is key and each employee is value of that dictionary and each employee is in Python List.

After that I got the employees and compare that value is in how many jobs, using that dictionary. So count that value and Add it to another dictionary. Now we user input M value and compare that dictionary values with M value. If it's Greater than or equal we get that value's key. Like that we added best employee for another list that name is "sample" in my program . And also I added M value of that employees to another list that name is "selectedcount" in my program.

Now I define recursive combination function for search all combination value from a list. There are several list use in that function. After combination all value of "sample" list lists we filter that combination count of each combination greater than or equal M value. I used several lists for filter that combination list. I use Done() function for do that. Like that scenario I calculate the all combination value that selected to combina_list list and Added their combination count for "combina_count list.

After that I filter that combination values that if second Greater combination values is in Greater combination value get second greater combination value count and compare it with Greater combination value count. If it's Greater than Greater combination value count. We add that second greater combination value to that PTS list. And other condition is if second greater combination value not in greater combination value it will Add to PTS list. Like that scenario next we get second greater combination value and third greater combination value and do the same process. Like that scenario program will get all combination value and do same process.

After calculating PTS list we get the Greater combination value and get all combination formula from that PTS value. Firstly we user input Trust value(T). And after that get that combination formula and calculate the TV of each sub formula. If PTS value is ABCD sub formulas are A->BCD , B->ACD ,C->ABD , D->ABC ,AB->CD , AC->BD,..... And we generate TV as follow. If sub formula is A->BCD TV is M value of ABCD devide by M value of A multiplied by 100. If that TV value is greater than or equal that inputed T value that sub formula is add to a list and print. Like that same process will happened. So that is the solution for that problem.

So I used only List , Dictionary data structure for solve that problem.

Complexity of my data structures.

I used only List and Dictionary. So time complexity of search from list is O(n) and search from dictionary is O(n). So I think I can satisfy that.

Special cases I encountered and how I solve that .

In combination function I encountered that the selection of a number of things, taking some or all of them at a time is called combination. The number of ways of selecting r things out of n dissimilar things is denoted by C(n, r) or ${}^{n}C_{r}$. In combination order does not matter. The number of combinations of n objects taken r, (r < n) is denoted by ${}^{n}C_{r}$

And also I used the library method copy. I import copy from library for success my combination easily.

Sometimes I think the complexity of all program is not best. So I fully tried to solve that. But It's not happened.

So that's the report of My program.