MONTE HALL - 3 DOORS AND A TWIST 02

So first of all let me give you a brief introduction of what Monte hall problem is all about, in Monte hall problem we have three doors two of them comprises of goods and one of them comprises of a price say bmw in this whole programme screen cast we will be using bmw as the price. So in Monte hall problem participant is asked to choose any one of this doors, after he or she had made his or her choice the host ask the participant to change his choice what the host basically does here is he opens a particular door out of these three doors ok now we are left with only two doors one that has been chosen by the participant and one that has been opened yet so now the host ask the participant to change his choice, it depends on the participant to swap or not swap his or her choice now what the participant do here is should he swap or not swap this is the question that we should answer ok so this seems question will be exploring will be answering in the programming screen cast will have a kind of setup here in which we have three doors two of them comprise of goods and one of them comprise of bmw after the choice whether the participant swapped or not swapped we will we will try to explore this fact that what is the optimal strategy here, whether the participant should swap or not swap? Let us try to explore this fact so let us start with the programming screen cast of Monte hall. As i said we need three doors here so i will be taking a list here namely doors ok this is the list doors, i will be taking another list here called goat door that will keep a track of the door that comprise of goats. After that i will be taking two variables swap and don’t swap this will take care of the fact of number of swap wins and number of don’t swap wins, swap will basically keep a track of number of swap wins and just i will write here number of swap wins and don’t swap will keep a track of number of don’t swap wins don’t swap wins ok now that we are done with it we will generate a random number here that will keep a track of what particular door will comprise of bmw? So since we are given only three doors here we will call them zeroth door, first door and second door so we have to generate a random number here random dot randint from zero to two ok so either any of these three doors will comprise of bmw so i will just write here ‘x’ will comprise ‘xth’ door will comprise of bmw ok now that we are done with it after that i will just write doors x will comprise of bmw yes we are done with it now, now the rest of the doors should comprise of goats now i will just do that here i in range just write zero to three since this will take into count that this is zero one and two so if i here is equal to is equal to ‘x’ then to continue ok, continue what does continue what does continue do here, what purpose does it serve here, if i write continue then it will go to the start of the loop if i write continue it will go to the start of the loop since the ‘xth’ door is already comprise of the bmw we only need to take into the count the doors except ‘x’ so if i equal to is equal to x here it will again go to the start of loop and implement i, because we are not considering x here we are only considering the doors that comprise of goat else what we are going to do here else will just see doors of i it will comprise of goat so you just write goat here, one thing we have to do here we have to append goat door also here goat door dot append and particular index that is i here so we are done with the doors two of them comprises of goat and one of them comprises of bmw after that we have to make the user input its choice so let us do that choice is equal to int enter your choice after that int input you just write int input that will make the user input his or her choice now we have to we also now that user has made his choice now the host will ask the user about swap or not to swap but before that he will open a door, so door open will take a variable here door open i will again generate it randomly here so i will write random dot choice goat door now please note this fact that we can only open the door that comprise of a goat we can only open the door that comprise of a goat so we are only considering goat door here, i hope that you are familiar here with random dot choice we have already explained it so goat door is a list here from which from which it choice from which a door will be will be taken randomly so here we are we have to open the door only comprise of goat so i will write here open a door that comprises of goat. Ok now that we have opened the door after that what we have to do here is we also need to take care of this fact in mind that the choice and the door open should be same here for example if the participant has made a choice of a door that comprises of a goat and we are also opening that particular door please note the fact that this is not allowed here so we need to also take care of this fact that choice and the door open shouldn’t be same so i will just have a while loop here while door is equal to is equal to choice i will again and again choose randomly from the list goat door so i will just write door open is equal to random dot choice goat door you just need to write here i will write the comment here door open shouldn’t be equal to choice made by the participant ok so we have also done that after that we have to give a choice to the user whether he or she wants to swap or not so you just write swap int input we don’t need to write int here you just write input do you want to swap? He or she will answer in y or no or in y or n ok so done that after that you have to apply a if loop here so just write if ch is the choice made by the participant yes then it will be y then what you have to do here is if doors choice is equal to is equal to goat that means he made a choice of door that comprised of goat at initial stage then he will win here print player wins ok and you have to increment swap variable here else what you have to do is else print player lost, i will again explain the if else loop here what we are doing here is if ch is equal to is equal to y that means the player has chosen to swap if he has chosen to swap then we have to we have to take care of the fact that if he has chosen goat in initial stage in initial choice then if he swaps here then obviously he will get a bmw, because already a door comprising of goat has been opened and a door that has goat he has chosen in its initial stage and now he is swapping now if he will swap he will get bmw ok so he is winning through swap so we will increment the swap variable here else if he has chosen bmw at his first place and now he is swapping obviously he is going to lose this game so now we are done with if ch is equal to is equal to y and if he doesn’t chose to swap here what will he do here? If he doesn’t chose to swap here will write if doors choice is equal to is equal to goat. If his choice was goat initially then and he doesn’t chose to swap to then he is bound to lose the game so you will just write print player lost done else if he didn’t chose goat at his first place then he will win because he has chosen bmw in at in his initial choice so just write print player wins. Ok and now he has won because he hasn’t swap so you have to increment the don’t swap variable here so just increment the don’t swap is equal to don’t swap plus one. After that you have to print this number of swap and don’t swap wins, so let us try to run this programme so just save it Monte hall dot, sorry so let us try to run this programme again there is some error here i will just show you it is saying random is not defined because we haven’t imported the library random so you just write import random, list index out of range it is giving us an error list index out of range so how can you handle that? Let me think about it, we can easily handle that i think we need to initialised the list here we just initialise it with zero ok since we are only using three doors here multiplied it by three also initialise this particular list here that is goat door so you write zero here and multiplied with by two, so again run it name door is not defined so it is basically doors not door let me check if i have used it in some other place no so let us try to run it again, enter your choice so you have to enter your choice from of your door choice it can be zero one or two so i will enter zero, do you want to swap? I will choose swap, so player wins ok now the player has won because he has swap, again i will run it, enter your choice i will enter two here, do you want to swap? No. Player wins also here so again run enter your choice i will enter two, do you want to swap? Yes i want to swap the player lost because he or she swapped again run one do you want to swap? Yes. Player wins again enter your choice? Two. No, the player lost so how can we keep a track of this how can we find out the what is the optimal strategy here, i am really confused so what i will do here is i will run this programme again and again for example ten times and then try to find out what are the number of swap wins and number of don't swap wins here so that i will just take a variable here so i will just take a variable for example i took j, j is equal to zero and i have while loop here, while j is less than ten i will run ten times for that you have to indent it properly so indent it properly so since we have to run it ten times so i will just use a while loop here for that you have to indent it properly ok, so now this has to go here, this has to go here, this here now it has to be inside the while loop after that the choice thing will also inside the loop so you just write choice here this if loop was inside this if loop so this is nested basically this else is for the above if please indent it properly i will request you otherwise you will get an error after that this else so for the above if ok now that we are done with it. So let us try to run it. Enter your choice, zero. Do you want to swap? Yes. Enter your choice, two. Do you want to swap? Yes. Enter your choice, one. Do you want to swap? No. Two swap, yes. One swap, yes. So it is going on and on i think i didn't i think we didn't increment the j point j variable here, yes you need to increment the j variable here so just write j is equal to j plus one so that it just run for definite number of times ok so i will just do that, ok we have exhibit it now let us try to run it again so that we can get what is optimal strategy here. Enter your choice, zero. Do you want to swap? Yes. Player wins, do you want to swap? No. Do you want enter your choice, two. Do you want to swap? Yes. One Do you want to swap? No. Zero, do you want to swap? Yes. Enter your choice, two Do you want to swap? No. Enter your choice, two. Do you want to swap? Yes. Enter your choice, one. Do you want to swap? Yes. Enter your choice, one. Do you want to swap? Yes. Enter your choice, zero. Do you want to swap? Yes. So here number of swap wins as you can see is five and number of don't swap wins is one so is swapping is the optimal strategy here? Do you really think so? Is swapping the optimal strategy here? Let us try to run this once again and try to find out what is the optimal strategy, so let us try to do that again here zero, Do you want to swap? Yes. One, no, two Do you want to swap? No. One, no. Two, yes. Zero, yes. Two, yes. One, no. Zero, no. One, yes. So here also number of swap wins are greater than number of don't swap wins so i will run it once again so has to get the clear cut idea of what is happening here? Enter your choice, zero. Do you want to swap? Yes. Player lost, one, Do you want to swap? Yes. Two. Do you want to swap? Yes. One, Do you want to swap? Yes. Two, Do you want to swap? Yes. Enter your choice, one. Do you want to swap? Yes. Zero, do you want to swap?

No. Two, no. One, no. Zero, no. So number of swap and number of don’t swap wins are equal here so we are getting difference distribution but in every case that we have run here number of swap wins are either greater than number of don’t swap wins or they are equal to number of don’t swap wins. So we can say that we are not sure of this fact that we will always win if we swap but there is higher probability of you winning if you swap. Yes this seems right from the programme that we are doing here, number of swap wins are always greater than equal to number of don’t swap wins so we can say that there is higher probability of you winning if you swap and there is lower probability of winning if you don’t swap, so now that we are done with the programme, we will just go through the programme once again so what we are really doing here is first of all you have to make two list, first one is doors and second one is goat door we are taking only three doors here, zeroth door, first door and second door and we have two goat doors here, two of them comprise of goats and one of them comprise of bmw, next what we are doing here is we are taking the track of number of swap wins and number of don’t swap wins through these variables after that we will have a while loop, while we are using a while loop here because we need to keep a track of number of swap wins and number of don’t swap wins ok, so i am running this programme ten times, you can run it fifty times, hundred times and check what is the optimal strategy? According to me it is number of swap wins it is swapping strategy and it will work in your case too please check it after that and taking the variable ‘x’ here this particular door except door will comprise of bmw then we are using a for loop here because the rest of the doors would comprise of goats except door comprises of bmw and rest of the doors comprises of goat, now what we are doing here is we are using a new key word here continue why i am using here because except door already comprises of bmw and if i is equal to is equal to ‘x’ here then we have to continue, we have to bring the control of the programme to the start of the loop because we are not considering ‘x’ here we are considering list of the doors here and else if i is not equal to ‘x’ then doors at i will comprise of goat, i also need to append in goat door list because that is keeping a track of doors that is comprise of goats after that you have to enter your choice ok the choice the participant chooses after that door open now the host will open one of the doors ok please note this fact in mind the open door and the choice shouldn’t be equal ok? they shouldn’t be equal so we are using the while loop here so door open is equal to random choice door since we are already using a goat door list here i will just make a choice from the list goat door ok, after that you have to make the user input whether he or she wants to swap or not ok, you just ask the user do you want to swap y or n if ch is equal to is equal to y then if he has if ch is equal to is equal to y that means he or she wants to swap and he has made a choice of goat at first place and if he has made a choice of goat at first place and now he is choosing to swap then he is bound to win, now he will get a bmw so in this particular case the player wins he has won because he has chosen to swap so we will increment the number of swap wins else he will lose the game after that if he doesn’t chose to swap and he has chosen goat in his initial choice then he is bound to lose the game, else if he has chosen bmw and he is not interested to swap that means he is going to win the game, in this particular case number of don’t swap wins will increment it will be incremented. Now since we are using while j less than ten we are using j variable here we need to increment j here so we incremented j here after that we are printing number of swap wins and number of don’t swap wins. I hope you understood this particular programme on Monte hall and this programming screen cast is useful to you guys thank you happy leaning.