The trace for me starts with murderer(_5382) which I assume the _5382 means that the variable for murderer is undeclared. It then calls hair(_5382, brown) because it is the if of the murderer function. Hair(x, brown) has an if statement which is attire(x, pincenez) which calls attire(sir_raymond, tattered_cuffs) because it is the if of attire(mr_woodley, pincenez). attire(sir_raymond, tattered_cuffs) then calls room(X, 16) because if something has tattered_cuffs they stayed in room 16, which we find out sir_raymond didn't stay in room 16, but in room 10. Because it failed it backtracks back to attire(_5382, pincenez) but this time calls attire(mr_woodley, tattered_cuffs) and then room(mr_woodley, 16) which it fines that is true because mr_woodley stayed in room 16. The program then backtracks to attire(sir_raymond, pincenez) which leads to hair(sir_raymond, brown) and then all the back to the start murderer(_5382) which now _5382 is set to sir_raymond because he has brown hair and pincenez which all of these the murderer has. To me prolog checks all ifs down a path before checking the function that if comes from to make sure it checks all possibilities and if the possibilities fail in that path it backtracks and goes down a different path.