Programming Problem 1

For this problem, we intend to use Backtracking similar to the way it is implemented in N-Queens problem.

The problem can be easily represented as follows:

The list of Persons are:

{ Frank[F], George[G], Heather[H], Irene[I], George[G] }

The list of Items that were ordered are:

{ Amplifier[A], Banister[B], Candelabrum[C], Doorknob[D], Elephant[E] }

The list of addresses are:

{ Kirkwood[K], Lake Avenue[L], Maxwell Street[M], North Avenue[N], Orange Drive[O] }

We have the following conditions from the given statement which we push into a data structure. In case any of the following condition is satisfied, then the combination we have is wrong since no one got the item they ordered.

Person	Item	Address	
{ Frank	, Door Knob	,	}
{ Heather	,	, Orange Drive	}
{ Jerry	, Item[Heather]	,	}
{ Person[Candelabrum]	, Bannister	,	}
{ Person[Bannister]	, Item[Irene]	,	}
{ Person[Kirkwood]	, Item[George]	,	}
{ Person[Lake Avenue]	,	, Kirkwood	}
{ Person[North Avenue]	, Elephant	,	}
{ Person[Elephant]	,	, Maxwell Street	}
{ Person[Maxwell Street]	, Amplifier,	,	}

Now we prepare a list for each person with the possible items and addresses:

Person	Item	Address
Frank	{ A, B, C, E }	{ K, L, N, O }
George	{ A, B, C, D }	{ K, L, M, N, O }
Heather	{ A, B, C, D, E}	{ A, B, C, D, E }
Irene	{ C, D, E, A }	{ L, M, N, O }
Jerry	{ A, B, C, D, E}	{ K, L, M, N, O }

We have the above list with the possible combinations.

Algorithm to find the Goal States:
————————————————

Begin:

Create a master list of all possible items and address

While Master list of Items and Address is not null

Move to first person

while Person is not null:

assign item and address to person

remove the assigned item and address from the list of other persons

move on to the next person

check conditions

if (all conditions are satisfied)

append the combination to goal state

Remove the combination from the master list

End

Following this algorithm, we'll arrive at the list of all possible goal states.