**Artificial Intelligence**

***CSC 411***

***Assignment***

**No # 04**



Mian Afzaal Zahoor (01-134182-094)

Suleman (01-134182-059)

BS (CS) – 6A

**Department of Computer Sciences**

**BAHRIA UNIVERSITY, ISLAMABAD**

**Objective**

The objective of this assignment is to inculcate the concepts of first order logic and expert system. This assignment will help to understand the concepts of unification, recursion, backward and forward chaining using logical programming in prolog.

**Part-II: Designing an Interface for Expert System for Searching Room in Building:**

For designing an interface in you need to use python. We will be discussing about how to design GUI in python in Lab 9. For your understanding you can refer to below given figure, which shows interface of an expert system. Please note that for above part your interface should have the option of adding facts and rules in knowledgebase.

For Part-I interface should take values of destination room in case locating it in a particular building, while for implementing path and depth first search GUI should take values of initial and final room from the user. Below is an example of grid which you need to display in your python GUI, for finding a particular room on first floor.

Apart from this grid your interface should have text boxes in which user writes the room which he wants to search and results should be given in the form of grid and text( sample given in Part-I).

Table

Description automatically generated

**ANSWER**

**Python Code**

#<<-------------------------LIST OF ALL THE IMPORTED MODULES------------------------->>

from pyswip import Prolog

import pytholog as pl

from tkinter import \*

from tkinter import messagebox

import tkinter

import tkinter as tk

#<<-------------------------DISPLAYING MAP WITH BUTTONS FUNCTION------------------------->>

def createmap():

#DISPLAYING MAP OF XC FLOOR 1 USING BUTTONS

statshop = tk.Button(root, text = "STAT-SHOP", width = 10, bg = "gray", fg = "white")

statshop.place(x = 210, y = 310)

xc1 = tk.Button(root, text = "XC-1", width = 10, bg = "gray", fg = "white")

xc1.place(x = 290, y = 310)

xc2 = tk.Button(root, text = "XC-2", width = 10, bg = "gray", fg = "white")

xc2.place(x = 370, y = 310)

xc3 = tk.Button(root, text = "XC-3", width = 10, bg = "gray", fg = "white")

xc3.place(x = 450, y = 310)

xc4 = tk.Button(root, text = "XC-4", width = 10, bg = "gray", fg = "white")

xc4.place(x = 530, y = 310)

sacs = tk.Button(root, text = "SA-CS", width = 10, bg = "gray", fg = "white")

sacs.place(x = 610, y = 310)

saeee = tk.Button(root, text = "SA-EEE", width = 10, bg = "gray", fg = "white")

saeee.place(x = 690, y = 310)

lobby = tk.Button(root, text = "LOBBY", width = 10, bg = "gray", fg = "white")

lobby.place(x = 770, y = 310)

xc5 = tk.Button(root, text = "XC-5", width = 10, bg = "gray", fg = "white")

xc5.place(x = 850, y = 310)

xc6 = tk.Button(root, text = "XC-6", width = 10, bg = "gray", fg = "white")

xc6.place(x = 930, y = 310)

xc7 = tk.Button(root, text = "XC-7", width = 10, bg = "gray", fg = "white")

xc7.place(x = 1010, y = 310)

xc8 = tk.Button(root, text = "XC-8", width = 10, bg = "gray", fg = "white")

xc8.place(x = 1090, y = 310)

eelab = tk.Button(root, text = "EE-LAB", width = 10, bg = "gray", fg = "white")

eelab.place(x = 1170, y = 310)

mainlobby = tk.Button(root, text = "MAIN LOBBY", width = 147, bg = "gray", fg = "white")

mainlobby.place(x = 210, y = 337)

hodcs = tk.Button(root, text = "HOD-CS", width = 13, bg = "gray", fg = "white")

hodcs.place(x = 210, y = 364)

hodmedia = tk.Button(root, text = "HOD-MEDIA", width = 13, bg = "gray", fg = "white")

hodmedia.place(x = 314, y = 364)

hodeee = tk.Button(root, text = "HOD-EEE", width = 13, bg = "gray", fg = "white")

hodeee.place(x = 418, y = 364)

wrlady = tk.Button(root, text = "WR-LADY", width = 13, bg = "gray", fg = "white")

wrlady.place(x = 522, y = 364)

admin = tk.Button(root, text = "ADMIN", width = 13, bg = "gray", fg = "white")

admin.place(x = 626, y = 364)

stairs = tk.Button(root, text = "STAIRS", width = 13, bg = "gray", fg = "white")

stairs.place(x = 730, y = 364)

girlscom = tk.Button(root, text = "GIRLS-COM", width = 13, bg = "gray", fg = "white")

girlscom.place(x = 834, y = 364)

wrgents = tk.Button(root, text = "WR\_GENTS", width = 13, bg = "gray", fg = "white")

wrgents.place(x = 938, y = 364)

stairs = tk.Button(root, text = "STAIRS", width = 13, bg = "gray", fg = "white")

stairs.place(x = 1042, y = 364)

facultycub = tk.Button(root, text = "FACULTY-CUB", width = 13, bg = "gray", fg = "white")

facultycub.place(x = 1146, y = 364)

#DISPLAY BOX 03

dispbox3 = tk.Label(root, text = "GROUND FLOOR MAP", bg = "white", fg = "blue", font = "Ariel 15 bold")

dispbox3.place(x = 580, y = 400)

#DISPLAYING MAP OF XC FLOOR 2 USING BUTTONS

xc9 = tk.Button(root, text = "XC-9", width = 11, bg = "gray", fg = "white")

xc9.place(x = 210, y = 440)

xc10 = tk.Button(root, text = "XC-10", width = 11, bg = "gray", fg = "white")

xc10.place(x = 297, y = 440)

xc11 = tk.Button(root, text = "XC-11", width = 11, bg = "gray", fg = "white")

xc11.place(x = 383, y = 440)

xc12 = tk.Button(root, text = "XC-12", width = 11, bg = "gray", fg = "white")

xc12.place(x = 470, y = 440)

xc13 = tk.Button(root, text = "XC-13", width = 11, bg = "gray", fg = "white")

xc13.place(x = 556, y = 440)

courtroom = tk.Button(root, text = "SA-CS", width = 11, bg = "gray", fg = "white")

courtroom.place(x = 643, y = 440)

compoff = tk.Button(root, text = "SA-EEE", width = 11, bg = "gray", fg = "white")

compoff.place(x = 729, y = 440)

xc15 = tk.Button(root, text = "LOBBY", width = 11, bg = "gray", fg = "white")

xc15.place(x = 816, y = 440)

xc16 = tk.Button(root, text = "XC-5", width = 11, bg = "gray", fg = "white")

xc16.place(x = 902, y = 440)

xc17 = tk.Button(root, text = "XC-6", width = 11, bg = "gray", fg = "white")

xc17.place(x = 989, y = 440)

xc18 = tk.Button(root, text = "XC-7", width = 11, bg = "gray", fg = "white")

xc18.place(x = 1075, y = 440)

lawlib = tk.Button(root, text = "XC-8", width = 11, bg = "gray", fg = "white")

lawlib.place(x = 1162, y = 440)

mainlobby = tk.Button(root, text = "MAIN LOBBY", width = 147, bg = "gray", fg = "white")

mainlobby.place(x = 210, y = 467)

hodlaw = tk.Button(root, text = "HOD-LAW", width = 13, bg = "gray", fg = "white")

hodlaw.place(x = 210, y = 494)

fcr1 = tk.Button(root, text = "FC ROOMS-1", width = 13, bg = "gray", fg = "white")

fcr1.place(x = 314, y = 494)

fclaw = tk.Button(root, text = "FC LAW", width = 13, bg = "gray", fg = "white")

fclaw.place(x = 418, y = 494)

salaw = tk.Button(root, text = "SA-LAW", width = 13, bg = "gray", fg = "white")

salaw.place(x = 522, y = 494)

stairs = tk.Button(root, text = "STAIRS", width = 13, bg = "gray", fg = "white")

stairs.place(x = 626, y = 494)

wrlady = tk.Button(root, text = "WR-LADY", width = 13, bg = "gray", fg = "white")

wrlady.place(x = 730, y = 494)

fcoff = tk.Button(root, text = "FC OFFICES", width = 13, bg = "gray", fg = "white")

fcoff.place(x = 834, y = 494)

wrgents = tk.Button(root, text = "WR-GENTS", width = 13, bg = "gray", fg = "white")

wrgents.place(x = 938, y = 494)

stairs = tk.Button(root, text = "STAIRS", width = 13, bg = "gray", fg = "white")

stairs.place(x = 1042, y = 494)

fcr2 = tk.Button(root, text = "FC ROOMS-2", width = 13, bg = "gray", fg = "white")

fcr2.place(x = 1146, y = 494)

#DISPLAY BOX 04

dispbox4 = tk.Label(root, text = "FIRST FLOOR MAP", bg = "white", fg = "blue", font = "Ariel 15 bold")

dispbox4.place(x = 595, y = 530)

#DISPLAYING MAP OF XC FLOOR 3 USING BUTTONS

xc19 = tk.Button(root, text = "XC-19", width = 11, bg = "gray", fg = "white")

xc19.place(x = 210, y = 570)

xc20 = tk.Button(root, text = "XC-20", width = 11, bg = "gray", fg = "white")

xc20.place(x = 297, y = 570)

xc21 = tk.Button(root, text = "XC-21", width = 11, bg = "gray", fg = "white")

xc21.place(x = 383, y = 570)

xc22 = tk.Button(root, text = "XC-22", width = 11, bg = "gray", fg = "white")

xc22.place(x = 470, y = 570)

xc23 = tk.Button(root, text = "XC-23", width = 11, bg = "gray", fg = "white")

xc23.place(x = 556, y = 570)

xc24 = tk.Button(root, text = "XC-24", width = 11, bg = "gray", fg = "white")

xc24.place(x = 643, y = 570)

xc25 = tk.Button(root, text = "XC-25", width = 11, bg = "gray", fg = "white")

xc25.place(x = 729, y = 570)

xc26 = tk.Button(root, text = "XC-26", width = 11, bg = "gray", fg = "white")

xc26.place(x = 816, y = 570)

xc27 = tk.Button(root, text = "XC-27", width = 11, bg = "gray", fg = "white")

xc27.place(x = 902, y = 570)

xc28 = tk.Button(root, text = "XC-28", width = 11, bg = "gray", fg = "white")

xc28.place(x = 989, y = 570)

xc29 = tk.Button(root, text = "XC-29", width = 11, bg = "gray", fg = "white")

xc29.place(x = 1075, y = 570)

xclab = tk.Button(root, text = "XC LAB", width = 11, bg = "gray", fg = "white")

xclab.place(x = 1162, y = 570)

mainlobby = tk.Button(root, text = "MAIN LOBBY", width = 147, bg = "gray", fg = "white")

mainlobby.place(x = 210, y = 597)

mediahouse = tk.Button(root, text = "MEDIA HS", width = 13, bg = "gray", fg = "white")

mediahouse.place(x = 210, y = 624)

newsroom = tk.Button(root, text = "NEWS ROOM", width = 13, bg = "gray", fg = "white")

newsroom.place(x = 314, y = 624)

eslab = tk.Button(root, text = "ES LAB", width = 13, bg = "gray", fg = "white")

eslab.place(x = 418, y = 624)

stairs = tk.Button(root, text = "STAIRS", width = 13, bg = "gray", fg = "white")

stairs.place(x = 522, y = 624)

wrlady = tk.Button(root, text = "WR-LADY", width = 13, bg = "gray", fg = "white")

wrlady.place(x = 626, y = 624)

wrgents = tk.Button(root, text = "WR-GENTS", width = 13, bg = "gray", fg = "white")

wrgents.place(x = 730, y = 624)

stairs = tk.Button(root, text = "STAIRS", width = 13, bg = "gray", fg = "white")

stairs.place(x = 834, y = 624)

cctvroom = tk.Button(root, text = "CCTV ROOM", width = 13, bg = "gray", fg = "white")

cctvroom.place(x = 938, y = 624)

itroom = tk.Button(root, text = "IT ROOM", width = 13, bg = "gray", fg = "white")

itroom.place(x = 1042, y = 624)

cslab = tk.Button(root, text = "CS LAB", width = 13, bg = "gray", fg = "white")

cslab.place(x = 1146, y = 624)

#DISPLAY BOX 05

dispbox5 = tk.Label(root, text = "SECOND FLOOR MAP", bg = "white", fg = "blue", font = "Ariel 15 bold")

dispbox5.place(x = 580, y = 660)

#<<-------------------------FUNCTIONS FOR EXECUTING QURIES------------------------->>

prolog = Prolog

prolog.consult("ProjectCodeNew.pl")

def whereis():

room1 = entrybox1.get()

kb = pl.KnowledgeBase("AI\_ASSIGNMENT")

kb.from\_file("projectCodeNew.pl")

print(kb.query(pl.Expr("whereis("+room1+")")))

print(kb.rule\_search(pl.Expr("whereis("+room1+")")))

result = prolog.query(f"whereis("+room1+")")

print(result)

Output.insert(tk.END, result)

createmap()

def exist():

room1 = entrybox1.get()

floor = entrybox3.get()

kb = pl.KnowledgeBase("AI\_ASSIGNMENT")

kb.from\_file("projectCodeNew.pl")

print(kb.query(pl.Expr("exist("+room1+", floor"+floor+")")))

print(kb.rule\_search(pl.Expr("exist("+room1+", floor"+floor+")")))

result = prolog.query(f"exist("+room1+", floor"+floor+")")

print(result)

Output.insert(tk.END, result)

createmap()

def pathfrom():

room1 = entrybox1.get()

room2 = entrybox2.get()

kb = pl.KnowledgeBase("AI\_ASSIGNMENT")

kb.from\_file("projectCodeNew.pl")

print(kb.query(pl.Expr("pathfrom("+room1+", "+room2+", F)")))

print(kb.rule\_search(pl.Expr("pathfrom("+room1+", "+room2+", F)")))

result = prolog.query(f"pathfrom("+room1+", "+room2+", F)")

print(result)

Output.insert(tk.END, result)

createmap()

def connected():

room1 = entrybox1.get()

room2 = entrybox2.get()

floor = entrybox3.get()

kb = pl.KnowledgeBase("AI\_ASSIGNMENT")

kb.from\_file("projectCodeNew.pl")

print(kb.query(pl.Expr("connected("+room1+", "+room2+", floor"+floor+", F)")))

print(kb.rule\_search(pl.Expr("connected("+room1+", "+room2+", floor"+floor+", F)")))

result = prolog.query(f"connected("+room1+", "+room2+", floor"+floor+", F)")

print(result)

Output.insert(tk.END, result)

createmap()

def dfs():

room1 = entrybox1.get()

room2 = entrybox2.get()

kb = pl.KnowledgeBase("AI\_ASSIGNMENT")

kb.from\_file("projectCodeNew.pl")

print(kb.query(pl.Expr("dfs("+room1+", "+room2+", F)")))

print(kb.rule\_search(pl.Expr("dfs("+room1+", "+room2+", F)")))

result = prolog.query(f"dfs("+room1+", "+room2+", F)")

print(result)

Output.insert(tk.END, result)

createmap()

#<<-------------------------MAIN CODE WITH GUI USED FOR USER FRIENDLY ENVIRONMENT------------------------->>

root = tk.Tk()

root.title("XC CAMPUS MAP")

root.geometry("1400x800")

root.config(bg = "green")

#DISPLAY BOX 01

dispbox1 = tk.Label(root, text = "ENTER ROOM NUMBER 01", bg = "white", fg = "blue", font = "Ariel 15 bold")

dispbox1.place(x = 580, y = 10)

#ENTRY BOX 01 FOR QUERY

entrybox1 = tk.Entry(root, width = 30, bg = "light cyan", fg = "red")

entrybox1.place(x = 620, y = 50)

#DISPLAY BOX 02

dispbox2 = tk.Label(root, text = "ENTER ROOM NUMBER 02", bg = "white", fg = "blue", font = "Ariel 15 bold")

dispbox2.place(x = 580, y = 80)

#ENTRY BOX 02 FOR QUERY

entrybox2 = tk.Entry(root, width = 30, bg = "light cyan", fg = "red")

entrybox2.place(x = 620, y = 120)

#DISPLAY BOX 03

dispbox2 = tk.Label(root, text = "ENTER FLOOR NUMBER", bg = "white", fg = "blue", font = "Ariel 15 bold")

dispbox2.place(x = 590, y = 150)

#ENTRY BOX 03 FOR QUERY

entrybox3 = tk.Entry(root, width = 30, bg = "light cyan", fg = "red")

entrybox3.place(x = 620, y = 190)

#OUTPUT BOX TO DISPLAY RESULT

Output = tk.Text(root, height = 5, width = 70, bg = "white", fg = "black")

Output.place(x = 430, y = 220)

#DISPLAY BUTTON 01

whereis = tk.Button(root, text = "'WHERE IS' QUERY", width = 25, bg = "#856ff8", fg = "white", padx = 10, pady = 10, command = whereis)

whereis.place(x = 200, y = 700)

#DISPLAY BUTTON 02

exist = tk.Button(root, text = "'EXIST' QUERY", width = 25, bg = "#856ff8", fg = "white", padx = 10, pady = 10, command = exist)

exist.place(x = 400, y = 700)

#DISPLAY BUTTON 03

pathfrom = tk.Button(root, text = "'PATH FROM' QUERY", width = 25, bg = "#856ff8", fg = "white", padx = 10, pady = 10, command = pathfrom)

pathfrom.place(x = 600, y = 700)

#DISPLAY BUTTON 04

connected = tk.Button(root, text = "'CONNECTED' QUERY", width = 25, bg = "#856ff8", fg = "white", padx = 10, pady = 10, command = connected)

connected.place(x = 800, y = 700)

#DISPLAY BUTTON 05

dfs = tk.Button(root, text = "'DFS' QUERY", width = 25, bg = "#856ff8", fg = "white", padx = 10, pady = 10, command = dfs)

dfs.place(x = 1000, y = 700)

#EXIT BUTTON

dfs = tk.Button(root, text = "EXIT", width = 25, bg = "#856ff8", fg = "white", padx = 10, pady = 10, command = root.destroy)

dfs.place(x = 600, y = 750)

root.mainloop()

**ScreenShot**





**Prolog Code**

%FLOOR 1 RIGHT SIDE OF XC

adjacent(stationaryshop, xc1, floor1, right).

adjacent(xc1, xc2, floor1, right).

adjacent(xc2, xc3, floor1, right).

adjacent(xc3, xc4, floor1, right).

adjacent(xc4, sAcs, floor1, right).

adjacent(sAcs, sAEEE, floor1, right).

adjacent(sAEEE, lobby, floor1, right).

adjacent(lobby, xc5, floor1, right).

adjacent(xc5, xc6, floor1, right).

adjacent(xc6, xc7, floor1, right).

adjacent(xc7, xc8, floor1, right).

adjacent(xc8, electricLab, floor1, right).

%FLOOR 1 LEFT SIDE OF XC

adjacent(hodcs, hodMedia, floor1, left).

adjacent(hodMedia, hodEEE, floor1, left).

adjacent(hodEEE, ladiesWR, floor1, left).

adjacent(ladiesWR, adminoffice, floor1, left).

adjacent(adminoffice, stairs, floor1, left).

adjacent(stairs, girlscommonroom, floor1, left).

adjacent(girlscommonroom, gentsWR, floor1, left).

adjacent(gentsWR, stairs, floor1, left).

adjacent(stairs, facultyroom, floor1, left).

%FLOOR 2 RIGHT SIDE OF XC

adjacent(xc9, xc10, floor2, right).

adjacent(xc10, xc11, floor2, right).

adjacent(xc11, xc12, floor2, right).

adjacent(xc12, xc13, floor2, right).

adjacent(xc13, lawcourtroom, floor2, right).

adjacent(lawcourtroom, computeroffice, floor2, right).

adjacent(computeroffice, xc15, floor2, right).

adjacent(xc15, xc16, floor2, right).

adjacent(xc16, xc17, floor2, right).

adjacent(xc17, xc18, floor2, right).

adjacent(xc18, lawLibrary, floor2, right).

%FLOOR 2 LEFT SIDE OF XC

adjacent(hodlaw, facultyroom1, floor2, left).

adjacent(facultyroom1, facultyoflaw, floor2, left).

adjacent(facultyoflaw, sAlaw, floor2, left).

adjacent(sAlaw, stairs, floor2, left).

adjacent(stairs, ladiesWR, floor2, left).

adjacent(ladiesWR, facultyoff, floor2, left).

adjacent(facultyoff, gentsWR, floor2, left).

adjacent(gentsWR, stairs, floor2, left).

adjacent(stairs, facultyroom2, floor2, left).

%FLOOR 3 RIGHT SIDE OF XC

adjacent(xc19, xc20, floor3, right).

adjacent(xc20, xc21, floor3, right).

adjacent(xc21, xc22, floor3, right).

adjacent(xc22, xc23, floor3, right).

adjacent(xc23, xc24, floor3, right).

adjacent(xc24, xc25, floor3, right).

adjacent(xc25, xc26, floor3, right).

adjacent(xc26, xc27, floor3, right).

adjacent(xc27, xc28, floor3, right).

adjacent(xc28, xc29, floor3, right).

adjacent(xc29, xclab, floor3, right).

%FLOOR 3 LEFT SIDE OF XC

adjacent(mediahouse, newsroom, floor3, left).

adjacent(newsroom, eslab, floor3, left).

adjacent(eslab, stairs, floor3, left).

adjacent(stairs, ladiesWR, floor3, left).

adjacent(ladiesWR, gentsWR, floor3, left).

adjacent(gentsWR, stairs, floor3, left).

adjacent(stairs, cctvroom, floor3, left).

adjacent(cctvroom, itroom, floor3, left).

adjacent(itroom, cslab, floor3, left).

%DEFINING RULES

%RULE FOR GETTING THE ROOM INFORMTION

whereis(Room):- adjacent(Room, Droom, Fl, Adj), write(Room), write(" IS ADJACENT TO "), write(Droom), write(" ON THE "), write(Adj), write(" SIDE OF THE "), write(Fl).

%RULE FOR IF THE ROOM EXISTS ON THE GIVEN FLOOR

exist(X, Floor):- adjacent(X, \_, Floor, \_), write(X), write(" EXISTS ON THE "), write(Floor); adjacent(\_, X, Floor, \_), write(X), write(" EXISTS ON THE "), write(Floor).

%RULE FOR GETTING THE PATH FROM ONE ROOM TO ANOTHER

pathfrom(X,Y,[X,Y]):- adjacent(X,Y,\_,\_).

pathfrom(X,Y,[X|Xs]):- adjacent(X,W,\_,\_), pathfrom(W,Y,Xs).

%RULE FOR GETTING INFORMATION ABOUT THE CONNECTED ROOMS

connected(X, Y, F, [X, Y]) :- adjacent(X, Y, F, \_), write("ROOMS ARE CONNECTED AT "), write(F), write(" ALONG PATH").

connected(X, Y, F, [X|Xs]) :- adjacent(X, W, F, \_), connected(W, Y, F, Xs).

%RULE FOR DEFINING DEPTH FIRST SEARCH

dfs(Goal,Goal,[Goal]) :- write("PATH RECIEVED FROM DFS").

dfs(Node,Goal,[Node|Path]) :- adjacent(Node,NewNode,\_,\_), dfs(NewNode,Goal,Path).

**Queries and Outputs**

**Whereis**



**Exist**



**Path**



**Depth first Search**



**Connected**

