|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Artificial Intelligence  Practical #8 | | |
|  |  |  |  |
| **Name** | Sandeep Jain | **Roll Number** | 21302C0058 |
| **Subject/Course:** | **Artificial Intelligence** | | |
| **Topic** | **Search Algorithm** | | |
|  | | | |
| 1. Solve the block of World problem. | | | |
| **Code in Python**  a=["B","C","D","A"]  k=['A','B','C','D']  print('THE INITIAL STATE IS:',a)  print('THE FINAL STATE IS:',k)  b=[]  c=[]  d=[]  while True:  add=str(input("WHICH BLOCK DO YOU WANT TP PICK UP:"))  if(add=='A'):  print('A IS PICKED UP AND KEPT ON GROUND')  b.append(add)  a.remove(add)  print("a=",a,"\n","b=",b)  add=input("WHICH BLOCK DO YOU WANT TP PICK UP:")  if(add=='D'):  print('D IS PICKED UP AND KEPT ON GROUND')  c.append(add)  a.remove(add)  print("a=",a,'\n',"b=",b,'\n',"c=",c)  add=input("WHICH BLOCK DO YOU WANT TP PICK UP:")  if(add=='C'):  print('C IS PICKED UP AND KEPT ON GROUND')  d.append(add)  a.remove(add)  print("a=",a,'\n',"b=",b,'\n',"c=",c,'\n',"d=",d)  add=input("WHICH BLOCK DO YOU WANT TP PICK UP:")  if(add=='B'):  print('B IS PICKED UP AND PLACED ON A')  b.append(add)  a.remove(add)  print("a=",a,'\n',"b=",b,'\n',"c=",c,'\n',"d=",d)  add=input("WHICH BLOCK DO YOU WANT TP PICK UP:")  if add=='C':  print('C IS PICKED UP AND PLACED ON B')  b.append(add)  d.remove(add)  print("a=",a,'\n',"b=",b,'\n',"c=",c,'\n',"d=",d)  add=input("WHICH BLOCK DO YOU WANT TP PICK UP:")  if add=='D':  print('D IS PICKED UP AND PLACED ON C')  b.append(add)  c.remove(add)  if k==b:  print("a=",a,'\n',"b=",b,'\n',"c=",c,'\n',"d=",d)  print('GOAL STATE HAS BEEN ACHIEVED')  break  elif add=='A'or'B'or'C':  print('ALREADY PICKED UP \n START OVER')  break  else:  print('WRONG INPUT IS GIVEN \n START OVER')  elif add=='A'or'B':  print('ALREADY PICKED UP \n START OVER')  break  elif add=='D':  print('BLOCKS SHOULD BE PICKED UP IN ORDER \n START OVER')  break  else:  print('WRONG INPUT IS GIVEN \n START OVER')  elif add=='C' or 'D':  print('BLOCKS SHOULD BE PICKED UP IN ORDER \n PICK B NEXT TIME')  break  elif add=='A':  print ('TO ACHIEVE GOAL STATE DONT PICK UP "A" PICK THE BLOCKS IN ORDER')  break  else:  print ('WRONG INPUT IS GIVEN \n START OVER')  break  elif add=='A'or'D':  print ('ALREDY PICKED UP \n START OVER')  break  elif add=='B':  print ('BLOCKS SHOULD BE PICKED UP IN ORDER \n START OVER')  break  else:  print ('WRONG INPUT IS GIVEN \n START OVER')  break  elif add=='B'or'C':  print ('BLOCKS SHOULD BE PICKED UP IN ORDER \n START OVER')  break  elif add=='A':  print ('A ALREADY PICKED UP \n START OVER')  break  else:  print('WRONG INPUT IS GIVEN \n START OVER')  break  elif add=='B'or'C'or'D':  print ('BLOCKS SHOULD BE PICKED UP IN ORDER \n START OVER')  break  else:  print ('WRONG INPUT IS GIVEN \n START OVER')  break  **Output in Python**  image | | | |
| 1. Solve constraint satisfaction problem | | | |
| **Code in Prolog**  **adjacent(1,2). adjacent(2,1).**  **adjacent(1,3). adjacent(3,1).**  **adjacent(1,4). adjacent(4,1).**  **adjacent(1,5). adjacent(5,1).**  **adjacent(2,3). adjacent(3,2).**  **adjacent(2,4). adjacent(4,2).**  **adjacent(3,4). adjacent(4,3).**  **adjacent(4,5). adjacent(5,4).**  **color(1,orange,x). color(1,orange,y).**  **color(2,pink,x). color(2,pink,y).**  **color(3,purple,x). color(3,purple,y).**  **color(4,red,x). color(4,pink,y).**  **color(5,pink,x). color(5,purple,y).**  **conflict(Coloring):-adjacent(A,B), color(A,Color,Coloring), color(B,Color,Coloring).**  **Output in Prolog** | | | |
|  | | | |
|  | | | |