```
with plan Lab - 0.6 with a creeker & with who a
* Solve 8 queen's problem using hill-climbing algorithm
   > Function to calculate the number of attacks in early current itale
          def cale attack (state): (state) educate som fet
                      for ; in souge (8):
                                  for j in earge (i+1, 8): (a) attack++
         if abs (istrate[i] - etate[j]) == obs (i-i) attack ++
          ectures attack
                                                                                                                                    double seuls
 2) Funtin + main for
           def hill-dimb ():
                         state = ( &audom. &ardint (0,7) for i in range (3.))
                       ener attack = rale attack (state)
              for iter is range (100): Hil energy the connection of the connecti
                                                                                           (8):
                                    for sour in range (8): (3) source (4)
                                            for for is range (8):
                                                           if state [sow] = col:
                                                                       neighbour = state
                                                                       reighbour [row] = col
                                    next thate = min (reighbours, key = cale_attack)
                                 rent attack = colo attak (next state)

if rest attack > = cur-attack; break
                                  state = nex. state doesd: (9 = 8) ti
                                  attack = next-attack
                                                                                                                            @ Display the Loans
                                                                                                                   for in sage (s):
3) Faretion to display the board (1)
        def display (state):
for i in range (8):
                                     for j is range (8);
if itate (i) == j: print ("a")
                                                   else: print (".")
```

```
* Solve the 8 queen's problem using to algorithm
1) Function to calculate the heuristic value of states
 The heusistic value for 8 queers is the number of atlanks
                                    : ( otite ) doubts who fel
     def cale_attacks (state):
                                (8) good (8)
             for in sarge (8):
               for j in range (8): (6 11) agree in
                    if state (i) = = state (i); atlank ++
    ++ date (1-1) 10n f abs(state[i]-state[j]) == (i-j) alleh ++
                                                wells artis
     setuen attack
3 Function to implement A# algorithm
                                            Odne, the
      def A*(") in a i sof (5.0) tribus mbus:) = itali
            import heaps # import a hocup data etruture
           etate=()
             g = 8 # queers left : (001) 2000
            for in range (8):
              for j is range (8): (8) spran it may sont
                      f = cale-attacks (state) + 9
                   heap puch (state, f)
            for in range (8) i woodsplan
                  were take = heap-pop () = state tour
                   state, append (energ_ state)
                  of rest other to the state beat
            if (9 = = 0): break
                                    state = nex state
                                 dualto - treas - attent
3 Display the board
     for in range (8):
                               I faction to display the boased
         for j is range (8):
             oleo: (2) == 1 = [i]: peint (2012)
             else: print ("/.") .(5) 800 a 6 00}
                           ) truly : , == 11) data fi
```