| *#!------------------------------------ !#* """  Name : Tanjim Reza  Student ID : 20101065  Course : CSE422 - Artificial Intelligence  Section : 07  Lab Assignment: 3  """ *#!------------------------------------ !#* |
| --- |



| **import** math **import** random   *#! Task: 02*  student\_id = "25485465" student\_id = input("Enter Student ID:")  *#replace all 0 with 8 in student id* student\_id = student\_id.replace("0", "8") print(student\_id) fifth\_digit = student\_id[4] last\_two\_reversed = student\_id[-1:-3:-1] minimum\_points\_to\_win = int(last\_two\_reversed) high = math.ceil(int(last\_two\_reversed)\*1.5) minimum = int(fifth\_digit) maximum = int(high) shuffles = int(student\_id[3]) points = []  **for** i **in** range(shuffles):  points.append(random.randint(minimum, maximum)) *#! Manually added to verify*  *# points = [66, 74, 14, 73, 19, 26, 32, 40]* *# points = [5, 10, 15, 20, 25, 30, 35, 40]* *# points = [50, 15, 65, 78, 21, 56, 78, 89]* **def** **MiniMaxAlphaBetaPrunning**(position, depth, alpha, beta, isMaximizingPlayer):  **if** depth == 0:  **return** utilityFunction(position)  **if** isMaximizingPlayer:  maxEval = -math.inf  **for** i **in** range(2):  eval = **MiniMaxAlphaBetaPrunning**(position\*2+i, depth-1, alpha, beta, **False**)  maxEval = max(maxEval, eval)  alpha = max(alpha, eval)  **if** beta <= alpha:  **break**   **return** maxEval  **else**:  minEval = +math.inf  **for** i **in** range(2):  eval = **MiniMaxAlphaBetaPrunning**(position\*2+i, depth-1, alpha, beta, **True**)  minEval = min(minEval, eval)  beta = min(beta, eval)  **if** beta <= alpha:  **break**  **return** minEval   **def** **utilityFunction**(position):  **return** points[position]  result = int(**MiniMaxAlphaBetaPrunning**(0,3, -math.inf, math.inf, **True**))  output = f"Generated {shuffles} random points between the minimum and maximum points\n\ limits: {points}\n\ Total points to win: {minimum\_points\_to\_win} \n\ Achieved point by applying alpha-beta pruning = {result}\n\ The winner is {'Optimus Prime' **if** result >= minimum\_points\_to\_win **else** 'Megatron'}\n"  *#! Task: 02* shuffle\_answers = [] **for** shuffle **in** range(shuffles):  random.shuffle(points)  result = **MiniMaxAlphaBetaPrunning**(0,3, -math.inf, math.inf, **True**)  shuffle\_answers.append(result)  count = 0 **for** result **in** shuffle\_answers:  **if** result >= minimum\_points\_to\_win:  count += 1 output += f"\nAfter the shuffle:\n\ List of all points values from each shuffle:{shuffle\_answers}\n\ The maximum value of all shuffles: {max(shuffle\_answers)}\n\ Won {count} times out of {shuffles} number of shuffles\n"   print(output) |
| --- |