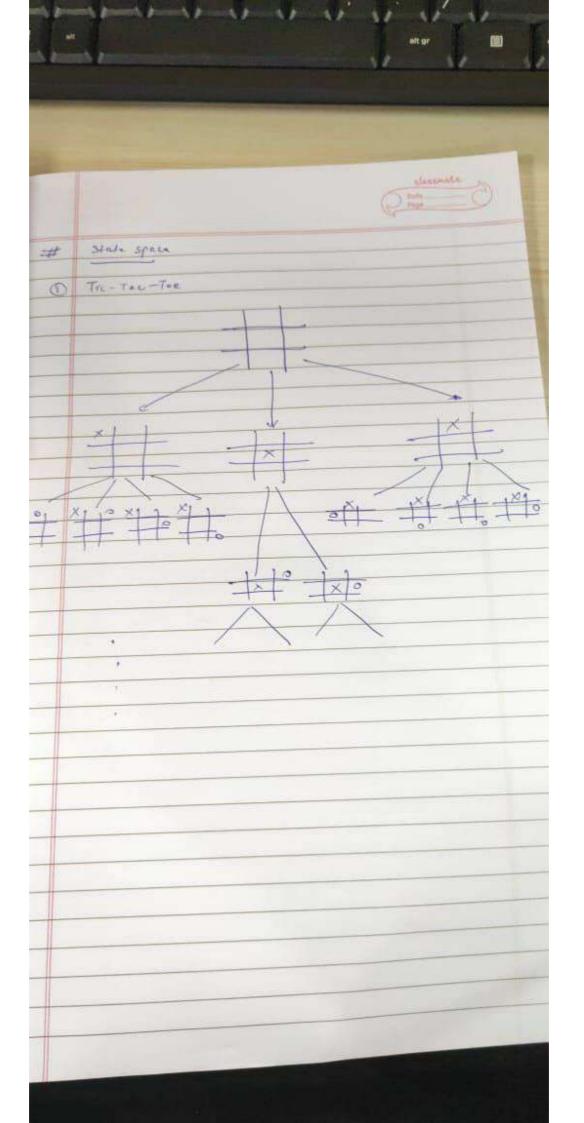
	classmate.
	Des 1/10/19
	LAGA
-15	Jankar and the day of a state of the state o
-	Implement a tra-tac-tac game
	Ingram title:
	Emplement a tig-tac-tre game.
	O'I - I district
*	Algorithms (honed) warmer stocks a recomment
	-function print_heard (board)
	print board formet and and and
	apartern variet anny
	function check winner (board)
	for each tow in board
	if all elements in row are some and not en
+	The state of the s
	for each column
-	if all elements in col are some and not en
-	Teturn the element
	of diagonal theoris yields same element and
+	not empty: return the element.
4	return the element.
1	return none
1 f	unetun is-board-full (board)!
4	return True if no empty spaces, the false.
f	inction main():
1.1.	instialise board with empty spaces
	Herration =0
	winner = none
	while winner is none.
	y Hereton is even:
	print_board (board)
	get user mput for now & column.

the prince binner binner binner
else rendently seems entry postain tox's binner to not Mane; board break winner message break break to introduction
else: Personally seems entry postain tox's  print winner manage  break introduction  the part of the printer (board)  the print winner manage  break introduction
else than to be comment board )  iteration to not Name:  break some introduction  break former message  break dinner message
else the test of in board  iteration to pot Mane:  iteration to the Mane:  break dinner message  break introduction  nath()  ame introduction
else teste to in board  place their test of postument (board)  brances a check time (board)  frint game introduction  break  break  break  break
else face to in board  else than to jee the top for the top is jee than to jee the top to post them to post the post them to post them
ette face to in board  iteration to let Nome:  typing some introduction  notice)  forest winner message  break introduction
elicitation tell select empty postum for it is increased to make winner (board)  which game introduction  main() game introduction
election teld select entity postum for it iteration teld select entity postum for it is not Nome:  If winner a creak winner (board)  Frint gome introduction  Break break introduction
elicitation to the board    idention to the posture to the posture to the property for the posture to posture to posture to property for the p
elle flace to in board  rendently select enjty postum for it  jenner = cheek_winner (board)  frint game introduction  break  break  hotograph
else thank to the board board )  iteration to to Name (board)  it winner to not Name (board)  frist winner manage  break tribroduction
else flace to in board  rendently select entity postum for it  jecretion to 1  winner to not Mone:  front winner memory  break  break  throduction
more there winner (board)  winner to not Name:  break winner memage  break
the to be and posture for 'x  check where (board)  the rot None:  the rot were at  the rotuntion
the to beard entry posture for 'x  check winner (board)  herroduction  textoduction
the in board  Check winner (board)  A pot None:  Introduction
the in board  there winner (board)  have menage
the place to in board  reaction to 1  something select entry postum for it  more to be the winner (board)  front winner message  break
the place to in board  rendonly seems entry posture for 'x  rendon to journan (board)  commer to pot Mone:
redonly seems empty posture for 'x
naturally scener empty postum for 'x
by seems empty postum for 'x
by solver emity posture for 'x
place to to board
- 0 -
validate Input
R Progressive
classiste



|--|

