Sagar Bangari (1BM22CS231)

Lab-1 Tic Tac Toe Game

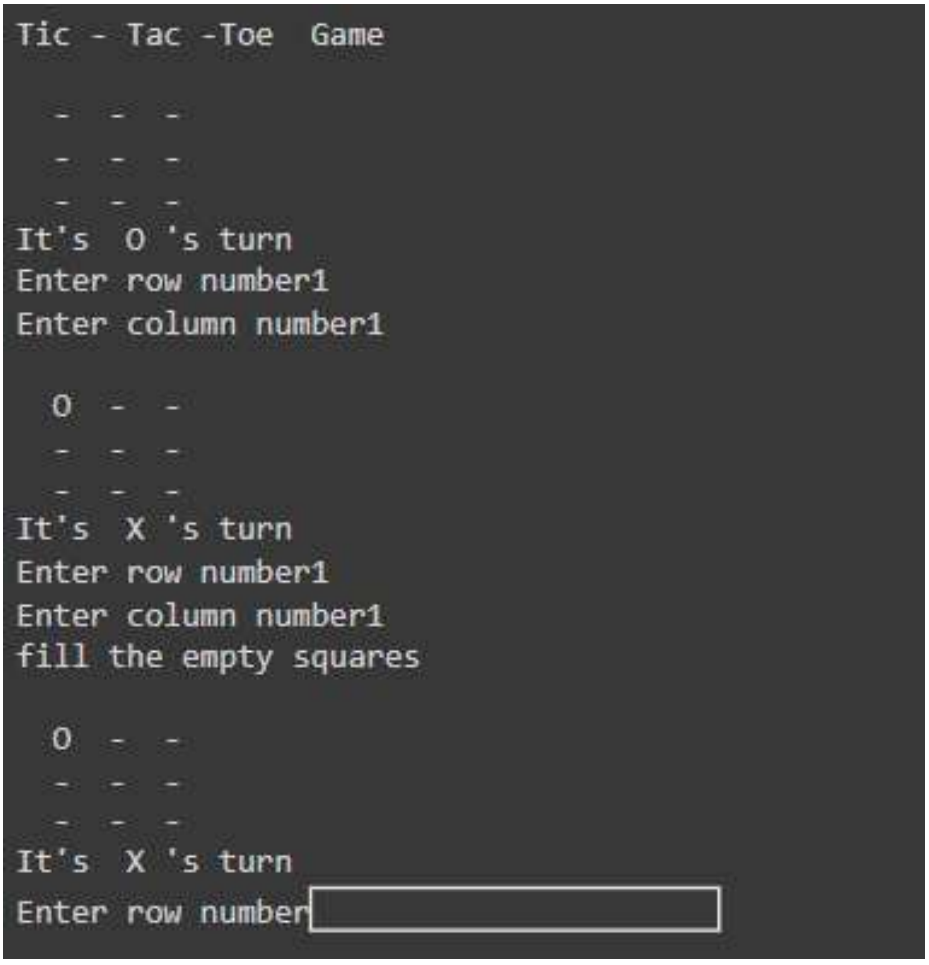
Code:

```
import random
array = ([['-', '-', '-'],['-',  '-', '-'],['-', '-', '-']])
x = 1
def board_filled():
   for i in range(3):
     for j in range(3):
       if(array[i][j]=='-'):
         return False
   return True

def player_won(a):
   for i in range(3):
     for j in range(3):
       if(array[i][j]!=a):
         break
       if j==2:
         return True

   for i in range(3):
     for j in range(3):
       if(array[j][i]!=a):
         break
       if j==2:
         return True
```

```
    if (array[0][0]==a and array[1][1]==a and array[2][2]==a):
        return True
    if (array[0][2]==a and array[1][1]==a and array[2][0]==a):
        return True
    return False

def show():
    for row in array:
      print ("".join(row))
    print()

def start():
    x = random.choice(['X','O'])
    while (1):
      show()
      row = int(input("Enter row number"))
      column = int(input("Enter column number"))
      if(array[row][column] != '-'):
          print("fill the empty squares")
          continue
      array[row][column] = x

      if( player_won(x) ):
        print("Player ", x , "has won!!")
        break
      if( board_filled() ):
```

```
      print("Game is drawn")
      break
    if(x=='X'):
      x = 'O'
      continue
    if(x=='O'):
      x = 'X'
  show()
start()
```

1) Wrong entry output:



2) Draw output:

```
Tic - Tac -Toe  Game

  -  -  -
  -  -  -
  -  -  -
It's  X 's turn
Enter row number1
Enter column number2

  -  X  -
  -  -  -
  -  -  -
It's  O 's turn
Enter row number1
Enter column number1

  O  X  -
  -  -  -
  -  -  -
It's  X 's turn
Enter row number2
Enter column number2

  O  X  -
  -  X  -
  -  -  -
It's  O 's turn
Enter row number1
Enter column number3

  O  X  O
  -  X  -
  -  -  -
It's  X 's turn
```

```
Enter row number2
Enter column number3

  O  X  O
  -  X  X
  -  -  -
It's  O 's turn
Enter row number2
Enter column number1

  O  X  O
  O  X  X
  -  -  -
It's  X 's turn
Enter row number3
Enter column number1

  O  X  O
  O  X  X
  X  -  -
It's  O 's turn
Enter row number3
Enter column number2

  O  X  O
  O  X  X
  X  O  -
It's  X 's turn
Enter row number3
Enter column number3
Game is drawn
  O  X  O
  O  X  X
  X  O  X
```

3)Win output:

```
Tic - Tac -Toe  Game

  -  -  -
  -  -  -
  -  -  -
It's  X 's turn
Enter row number1
Enter column number1

  X  -  -
  -  -  -
  -  -  -
It's  O 's turn
Enter row number1
Enter column number2

  X  O  -
  -  -  -
  -  -  -
It's  X 's turn
Enter row number2
Enter column number2

  X  O  -
  -  X  -
  -  -  -
It's  O 's turn
Enter row number1
Enter column number3

  X  O  O
  -  X  -
  -  -  -
It's  X 's turn
Enter row number3
Enter column number3
Player  X has won!!
  X  O  O
  -  X  -
  -  -  X
```

Vacuum World Cleaner

Code:

```
print("----Vacuum  Cleaner-----")
c = 0
rooms = int(input("Enter number of rooms(2/4)"))
if(rooms==2):
  Goal_state = {'A':0 , 'B': 0}
  Curr_state = {'A': 0  , 'B': 0}
else:
  Goal_state = {'A':0 , 'B': 0 ,'C':0 ,'D': 0}
  Curr_state = {'A': 0  , 'B': 0 , 'C':0 ,'D': 0}

if(rooms==2):
  r = (input("Enter robot position (A/B): "))
else:
  r = (input("Enter robot position (A/B/C/D): "))

Curr_state['A'] = int(input("Enter 0 or 1 for dust in  position A: "))
Curr_state['B'] = int(input("Enter 0 or 1 for dust in position B: "))
if(rooms==4):
  Curr_state['C'] = int(input("Enter 0 or 1 for dust in  position C: "))
  Curr_state['D'] = int(input("Enter 0 or 1 for dust in position D: "))

print("------------------")
def suck(loc, c):
    print(f"Location {loc} is dirty")
    print("Suck operation done")
```

```
    print(f"Position {loc} is cleaned")
    c += 1
    print("Cost =", c)
    Curr_state[loc] = 0
    print("Curr_state",Curr_state)
    print("------------------")
    return c

def left():
    print("Position C is clean")
    print("Curr_state",Curr_state)
    print("Moving Left")
    print("------------------")
    return 'D'

def right():
    print("Position A is clean")
    print("Curr_state",Curr_state)
    print("Moving Right")
    print("------------------")
    return 'B'

def up():
    print("Position D is clean")
    print("Curr_state",Curr_state)
    print("Moving Up")
    print("------------------")
```

```
    return 'A'


def down():
    print("Position B is clean")
    print("Curr_state",Curr_state)
    print("Moving Down")
    print("------------------")
    return 'C'


def vacuum_cleaner(loc, sta, c):
    if sta == 1:
        c = suck(loc, c)
    elif loc == 'A':
        loc = right()
    elif loc == 'C':
        loc = left()
    elif loc == 'B':
        loc = down()
    elif loc == 'D':
        loc = up()
    return loc, c


while True:
    print("Robot location ",r)
    sta = Curr_state[r]
    r, c = vacuum_cleaner(r, sta, c)


    if (Goal_state==Curr_state):
        print("All positions are clean!")
```

```
print("Goal state")

print(Goal_state)

print("Total cost is ",c)

break
```

Output:

```
----Vacuum  Cleaner-----
Enter number of rooms(2/4)4
Enter robot position (A/B/C/D): B
Enter 0 or 1 for dust in  position A: 0
Enter 0 or 1 for dust in position B: 1
Enter 0 or 1 for dust in  position C: 1
Enter 0 or 1 for dust in position D: 0
-----------------
Robot location  B
Location B is dirty
Suck operation done
Position  B is cleaned
Cost = 1
Curr_state {'A': 0, 'B': 0, 'C': 1, 'D': 0}
-----------------
Robot location  B
Position B is clean
Curr_state {'A': 0, 'B': 0, 'C': 1, 'D': 0}
Moving Down
-----------------
Robot location  C
Location C is dirty
Suck operation done
Position  C is cleaned
Cost = 2
Curr_state {'A': 0, 'B': 0, 'C': 0, 'D': 0}
-----------------
All positions are clean!
Goal state
{'A': 0, 'B': 0, 'C': 0, 'D': 0}
Total cost is  2
```