COMPUTER SCIENCE PROJECT

ANSH GOYAL
XII-A
12

RAGHAV AGGARWAL
XII-A
39

TIC TAC TOE TOURNAMENT ORGANISER

INDEX

CERTIFICATE

This is to certify that **ANSH GOYAL and RAGHAV AGARWAL** of class XII Science has prepared the investigatory computer project entitled '**TIC TAC TOE TOURNAMENT ORGANISER**'. The report is the result of their efforts and endeavors.

The report is found worthy of acceptance as final project report for the subject Chemistry of class XII. They have prepared the report under my guidance.

Mrs. Puja Malhotra

PGT COMPUTER SCIENCE

APEEJAY SCHOOL PITAMPURA

ACKNOWLEDGEMENT

Primarily we would like to thank God for being able to complete this project with success. Then we would like to thank my chemistry teacher Mrs. Puja Malhotra ,whose valuable guidance has been the ones that helped us patch this project and make it full proof success her suggestions and her instructions has served as the major contributor towards the completion of the project.

Then we would like to thank our parents and friends who have helped us with their valuable suggestions and guidance has been helpful in various phases of the completion of the project.

Last but not the least we would like to thank our classmates who have helped us a lot.

Ansh Goyal

XII-A

Raghav Aggarwal

XII-A

INTRODUCTION

The following project created fulfills the requirement of a tournament organizer of the game called tic-tac-toe.

Tic-tac-toe is a game of crosses and circles in which a participant has to get three same signs in a row either diagonally, horizontally or vertically.

To the game the fun of competing with friends in a round robin format have been added in which you can compete in a single group with the winner being the topper of group or multiple groups where the winner of each group then compete in a single group round robin format.

SOURCE CODE

```
import mysql.connector

import random

mysqlcon=mysql.connector.connect(host="localhost",use
r="root",passwd="",database="project")

mysqlc=mysqlcon.cursor()

print("OX"*30)

print("Welcome to tictactoe tournament we hope you do
your best")

gameno=0

notimes=0

unique=[]

schedule_error=0

player_error=0

while True:

    print("OX"*30)

    choice=int(input("what you want to do \n 1.enter
names of players \n 2.View standings \n 3.Play next
game \n 4.Create/View schedule \n 5.Exit \n 6.Update
Name\n"))
    if choice==1:
        print("OX"*30)
        player_error=1
        playerdata=[]
        groups=1
        players=1
        print("Entering names of Players: ")
        while groups%players==0:
            groups=int(input("how many groups: "))
            players=int(input("how many players: "))
            if players%groups!=0 or players/groups==1:
                print("Error: Number of players must be
divisible by",groups,"and quotient shouldnt be 1")
                break
            k=0
            for i in range(0,groups):
                k+=1
                l="Group"+str(k)
                mysqlc.execute("create table {}(player_id
char,player_name varchar(20),games_played int,win
int,loss int,draw int,points int)".format(l))
                print("entering names for group",i+1)
                for j in range(0,players//groups):
                    player_id=input("enter player id: ")
                    while player_id in unique:
                        player_id=input("enter unique id: ")
                    unique.append(player_id)
                    player_name=input("enter player name: ")
                    playerdata.append([player_id,player_name])
                    mysqlc.execute("insert into
{}(player_id,player_name,games_played,win,loss,draw,p
oints)
values({},'{}',0,0,0,0,0)".format(l,player_id,player_name))
            print("OX"*30)
            break


    elif choice==2:
```

```
    print("OX"*30)
    if player_error==0:
        print("Error: First Enter Players")
        continue
    mysqlc.execute("show tables")
    tables=mysqlc.fetchall()
    print(tables)
    sgroup=input("which group no you want to see or champions")
    if sgroup=="champions":
        mysqlc.execute("select * from champions order by points desc")
    else:
        shgroup="group"+sgroup
        mysqlc.execute("select * from {} order by points desc".format(shgroup))
    print("THE TABLE IS AS \n[PLAYER_ID,PLAYER_NAME,GAMES_PLAYED,WIN,LOSS,DRAW,POINTS]")
    for i in range(0,players//groups):
            standings=mysqlc.fetchone()
            print(standings)
        print("OX"*30)
elif choice==5:
    print("OX"*30)
    k=0
    for i in range(0,groups):
            k+=1
            l="Group"+str(k)
            mysqlc.execute("drop table {}".format(l))
    if groups>1:
        mysqlc.execute("drop table champions")
    print("THANK YOU FOR PLAYING")
    print("OX"*30)
    break
elif choice==4:
    print("OX"*30)
    schedule_error=1
    if player_error==0:
        print("Error: First Enter Player Names")
        continue
    fsch_fake=[]
    for mynameiskhan in range(1,groups+1):
        notimes=0
        mysqlc.execute("select g1.player_name,g2.player_name from group"+str(mynameiskhan)+ " g1 cross join group"+str(mynameiskhan)+ " g2 where g1.player_name!=g2.player_name order by g1.player_id+g2.player_id")
        schedul=mysqlc.fetchall()
        notimes+=1
        if notimes==1:
            schedule=schedul.copy()
        schedu=schedule.copy()
        oddsc=[]
        evensc=[]
        while len(schedu)!=0:
            hkhkhkh=random.randrange(0,len(schedule))
            if hkhkhkh%2==0:
                if schedule[hkhkhkh] not in evensc:
                    evensc.append(schedule[hkhkhkh])
                    schedu.remove(schedule[hkhkhkh])
            else:
                if schedule[hkhkhkh] not in oddsc:
                    oddsc.append(schedule[hkhkhkh])
                    schedu.remove(schedule[hkhkhkh])
        fsch=[]
        for element in evensc:
            fsch.append(element)
        for element in oddsc:
            fsch.append(element)
        fsch_fake.append(fsch)
        print("THE SCHEDULE FOR
GROUP",mynameiskhan, "IS AS FOLLOWED
\n[player1,player2]")
        for something in fsch:
            print(something)
```

```
        print("OX"*30)
    elif choice==3:
        if schedule_error==0:
            print("Error: First Create a Schedule")
            continue
        if groups==1:
            print("OX"*30)
            totalgames=1
            for hhhhhhhh in range(1,players+1):
                totalgames=totalgames*hhhhhhhh
            for jjjjjjjjjjj in range(1,totalgames+1):
                gameno+=1
                x=fsch[gameno-1][0]
                y=fsch[gameno-1][1]
                t=[["-","-","-"],["-","-","-"],["-","-","-"]]
#tic tac toe main matrice
                for llllllll in t:
                    print(llllllll)
                i=1
                c="3"
                while i<10 and c!="1" and c!="2":
                    if i%2==0:
                        print("its your turn",y)
                        k=input("enter column (1-3) -> ")
                        while k.isalpha()==True:
                            k=input("Enter no. between 1 and 3 -> ")
                        while int(k)>3 or int(k)<1:
                            k=input("Enter no. between 1 and 3 -> ")
                        m=input("enter row (1-3) -> ")
                        while m.isalpha()==True:
                            m=input("Enter no. between 1 and 3 -> ")
                        while int(m)>3 or int(m)<1:
                            m=input("Enter no. between 1 and 3 -> ")
                        while t[int(m)-1][int(k)-1]=="O" or t[int(m)-1][int(k)-1]=="X":
                            k=input("Input unoccupied column (1-3) -> ")
                            while k.isalpha()==True:
```

```
                    k=input("Enter no. between 1 and 3 ->
")
                while  int(k)>3 or int(k)<1:
                    k=input("Enter no. between 1 and 3 ->
")
                m=input("Input unoccupied row (1-3) ->
")
                while m.isalpha()==True:
                    m=input("Enter no. between 1 and 3 ->
")
                while int(m)>3 or int(m)<1:
                    m=input("Enter no. between 1 and 3 ->
")
            t[int(m)-1][int(k)-1]="O"
#entering the sign in specified position
            i+=1
            print()
            for a in range(0,3):
#printing matrice
                for b in range(0,3):
```

```
        print(t[a][b],end=" ")
    print()
if t[0][0]==t[0][1]==t[0][2]=="O":
    c="2"
    print(y,"won")
elif t[1][0]==t[1][1]==t[1][2]=="O":
    c="2"
    print(y,"won")
elif t[2][0]==t[2][1]==t[2][2]=="O":
    c="2"
    print(y,"won")
elif t[0][0]==t[1][0]==t[2][0]=="O":
    c="2"
    print(y,"won")
elif t[0][1]==t[1][1]==t[2][1]=="O":
    c="2"
    print(y,"won")
elif t[0][2]==t[1][2]==t[2][2]=="O":
    c="2"
```

```
        print(y,"won")
    elif t[0][0]==t[1][1]==t[2][2]=="O":
        c="2"
        print(y,"won")
    elif t[0][2]==t[1][1]==t[2][0]=="O":
        c="2"
        print(y,"won")
    elif i==9 and c!="1" and c!="2":
        c="3"
        print("draw it is")
    else:
        c="0"
        print()
        continue
else:   #chance of player 2
    print("its your turn",x)
    k=input("enter column (1-3) -> ")
    while k.isalpha()==True:
        k=input("Enter no. between 1 and 3 -> ")
        while int(k)>3 or int(k)<1:
            k=input("Enter no. between 1 and 3 -> ")
        m=input("enter row (1-3) -> ")
        while m.isalpha()==True:
            m=input("Enter no. between 1 and 3 -> ")
        while int(m)>3 or int(m)<1:
            m=input("Enter no. between 1 and 3 -> ")
        while t[int(m)-1][int(k)-1]=="O" or t[int(m)-1][int(k)-1]=="X":
            k=input("Input unoccupied column (1-3) -> ")
            while k.isalpha()==True:
                k=input("Enter no. between 1 and 3 -> ")
            while  int(k)>3 or int(k)<1:
                k=input("Enter no. between 1 and 3 -> ")
            m=input("Input unoccupied row (1-3) -> ")
            while m.isalpha()==True:
```

```
                                m=input("Enter no. between 1 and 3 -> ")
                            while int(m)>3 or int(m)<1:
                                m=input("Enter no. between 1 and 3 -> ")
                        t[int(m)-1][int(k)-1]="X"
                        i+=1
                        print("")
                        for a in range(0,3):
                            for b in range(0,3):
                                print(t[a][b],end=" ")
                            print()
                        if t[0][0]==t[0][1]==t[0][2]=="X":
                            c="1"
                            print("Player",x,"won")
                        elif t[1][0]==t[1][1]==t[1][2]=="X":
                            c="1"
                            print("Player",x,"won")
                        elif t[2][0]==t[2][1]==t[2][2]=="X":
```

```
    c="1"

    print("Player",x,"won")

elif t[0][0]==t[1][0]==t[2][0]=="X":

    c="1"

    print("Player",x,"won")

elif t[0][1]==t[1][1]==t[2][1]=="X":

    c="1"

    print("Player",x,"won")

elif t[0][2]==t[1][2]==t[2][2]=="X":

    c="1"

    print("Player",x,"won")

elif t[0][0]==t[1][1]==t[2][2]=="X":

    c="1"

    print("Player",x,"won")

elif t[0][2]==t[1][1]==t[2][0]=="X":

    c="1"

    print("Player",x,"won")

elif i==9 and c!="1" and c!="2":

    c="3"
```

```
                            print("draw it is")
                        else:
                            c="0"
                            print()
                        continue
                if c=="2":
                    mysqlc.execute("update group1 set games_played=games_played+1 where player_name='{}'".format(y))
                    mysqlc.execute("update group1 set win=win+1 where player_name='{}'".format(y))
                    mysqlc.execute("update group1 set points=points+2 where player_name='{}'".format(y))
                    mysqlc.execute("update group1 set loss=loss+1 where player_name='{}'".format(x))
                    mysqlc.execute("update group1 set games_played=games_played+1 where player_name='{}'".format(x))
                    mysqlcon.commit()
                elif c=="1":
```

```
            mysqlc.execute("update group1 set games_played=games_played+1 where player_name='{}'".format(x))
            mysqlc.execute("update group1 set win=win+1 where player_name='{}'".format(x))
            mysqlc.execute("update group1 set points=points+2 where player_name='{}'".format(x))
            mysqlc.execute("update group1 set loss=loss+1 where player_name='{}'".format(y))
            mysqlc.execute("update group1 set games_played=games_played+1 where player_name='{}'".format(y))
            mysqlcon.commit()
        elif c=="3":
            mysqlc.execute("update group1 set games_played=games_played+1 where player_name='{}'".format(x))
            mysqlc.execute("update group1 set draw=draw+1 where player_name='{}'".format(x))
            mysqlc.execute("update group1 set points=points+1 where player_name='{}'".format(x))
```

```
                mysqlc.execute("update group1 set games_played=games_played+1 where player_name='{}'".format(y))
                mysqlc.execute("update group1 set draw=draw+1 where player_name='{}'".format(y))
                mysqlc.execute("update group1 set points=points+1 where player_name='{}'".format(y))
                mysqlcon.commit()
        mysqlc.execute("select player_name from group1 where points=(select max(points) from group1)")
        winner=mysqlc.fetchall()
        print("winner is",winner[0])
        print("OX"*30)
    elif groups>1:#----------------------------------------------------------------------------
        print("OX"*30)
        for abb in range (1,groups +1):
            nummb=len(fsch_fake[0])
            boooooo="group"+str(abb)
            for cdd in range(1,nummb+1):
```

```
          print("Group", abb, "matches...")
          gameno+=1
          x=fsch_fake[abb-1][gameno-1][0]
#fsch_fake=[[('b', 'a'), ('a', 'b')], [('d', 'c'), ('c', 'd')]]
          y=fsch_fake[abb-1][gameno-1][1]
          t=[["-","-","-"],["-","-","-"],["-","-","-"]] #tic tac
toe main matrice
          print(cdd,')', x, "VS", y)
          for llllllll in t:
            print(llllllll)
          i=1
          c="3"
          while i<10 and c!="1" and c!="2":
            if i%2==0:
              print("its your turn",y)
              k=input("enter column (1-3) -> ")
              while k.isalpha()==True:
                k=input("Enter no. between 1 and 3 ->
")
```

```
while int(k)>3 or int(k)<1:
    k=input("Enter no. between 1 and 3 -> ")
m=input("enter row (1-3) -> ")
while m.isalpha()==True:
    m=input("Enter no. between 1 and 3 -> ")
while int(m)>3 or int(m)<1:
    m=input("Enter no. between 1 and 3 -> ")
while t[int(m)-1][int(k)-1]=="O" or t[int(m)-1][int(k)-1]=="X":
    k=input("Input unoccupied column (1-3) -> ")
    while k.isalpha()==True:
        k=input("Enter no. between 1 and 3 -> ")
    while  int(k)>3 or int(k)<1:
        k=input("Enter no. between 1 and 3 -> ")
```

```
            m=input("Input unoccupied row (1-3) -
> ")
            while m.isalpha()==True:
                m=input("Enter no. between 1 and 3
-> ")
            while int(m)>3 or int(m)<1:
                m=input("Enter no. between 1 and 3
-> ")
        t[int(m)-1][int(k)-1]="O"
#entering the sign in specified position
        i+=1
        print()
        for a in range(0,3):
#printing matrice
            for b in range(0,3):
                print(t[a][b],end=" ")
            print()
        if t[0][0]==t[0][1]==t[0][2]=="O":
            c="2"
            print(y,"won")
```

```
elif t[1][0]==t[1][1]==t[1][2]=="O":
    c="2"
    print(y,"won")
elif t[2][0]==t[2][1]==t[2][2]=="O":
    c="2"
    print(y,"won")
elif t[0][0]==t[1][0]==t[2][0]=="O":
    c="2"
    print(y,"won")
elif t[0][1]==t[1][1]==t[2][1]=="O":
    c="2"
    print(y,"won")
elif t[0][2]==t[1][2]==t[2][2]=="O":
    c="2"
    print(y,"won")
elif t[0][0]==t[1][1]==t[2][2]=="O":
    c="2"
    print(y,"won")
elif t[0][2]==t[1][1]==t[2][0]=="O":
```

```
                                c="2"
                                print(y,"won")
                            elif i==9 and c!="1" and c!="2":
                                c="3"
                                print("draw it is")
                            else:
                                c="0"
                                print()
                                continue
                        else:   #chance of player 2
                            print("its your turn",x)
                            k=input("enter column (1-3) -> ")
                            while k.isalpha()==True:
                                k=input("Enter no. between 1 and 3 ->
")
                            while int(k)>3 or int(k)<1:
                                k=input("Enter no. between 1 and 3 ->
")
                            m=input("enter row (1-3) -> ")
```

```
while m.isalpha()==True:
    m=input("Enter no. between 1 and 3 -> ")
while int(m)>3 or int(m)<1:
    m=input("Enter no. between 1 and 3 -> ")
while t[int(m)-1][int(k)-1]=="O" or t[int(m)-1][int(k)-1]=="X":
    k=input("Input unoccupied column (1-3) -> ")
    while k.isalpha()==True:
        k=input("Enter no. between 1 and 3 -> ")
    while  int(k)>3 or int(k)<1:
        k=input("Enter no. between 1 and 3 -> ")
    m=input("Input unoccupied row (1-3) -> ")
    while m.isalpha()==True:
        m=input("Enter no. between 1 and 3 -> ")
```

```
            while int(m)>3 or int(m)<1:
                m=input("Enter no. between 1 and 3
-> ")
        t[int(m)-1][int(k)-1]="X"
        i+=1
        print("")
        for a in range(0,3):
            for b in range(0,3):
                print(t[a][b],end=" ")
            print()
        if t[0][0]==t[0][1]==t[0][2]=="X":
            c="1"
            print("Player",x,"won")
        elif t[1][0]==t[1][1]==t[1][2]=="X":
            c="1"
            print("Player",x,"won")
        elif t[2][0]==t[2][1]==t[2][2]=="X":
            c="1"
            print("Player",x,"won")
```

```
elif t[0][0]==t[1][0]==t[2][0]=="X":
    c="1"
    print("Player",x,"won")
elif t[0][1]==t[1][1]==t[2][1]=="X":
    c="1"
    print("Player",x,"won")
elif t[0][2]==t[1][2]==t[2][2]=="X":
    c="1"
    print("Player",x,"won")
elif t[0][0]==t[1][1]==t[2][2]=="X":
    c="1"
    print("Player",x,"won")
elif t[0][2]==t[1][1]==t[2][0]=="X":
    c="1"
    print("Player",x,"won")
elif i==9 and c!="1" and c!="2":
    c="3"
    print("draw it is")
else:
```

```
            c="0"
            print()
        continue
if c=="2":
    mysqlc.execute("update {} set games_played=games_played+1 where player_name='{}'".format(boooooo,x))
    mysqlc.execute("update {} set win=win+1 where player_name='{}'".format(boooooo,x))
    mysqlc.execute("update {} set points=points+2 where player_name='{}'".format(boooooo,x))
    mysqlc.execute("update {} set loss=loss+1 where player_name='{}'".format(boooooo,x))
    mysqlc.execute("update {} set games_played=games_played+1 where player_name='{}'".format(boooooo,x))
    mysqlcon.commit()
elif c=="1":
```

```
            mysqlc.execute("update {} set
games_played=games_played+1 where
player_name='{}'".format(boooooo,y))
            mysqlc.execute("update {} set win=win+1
where player_name='{}'".format(boooooo,y))
            mysqlc.execute("update {} set
points=points+2 where
player_name='{}'".format(boooooo,y))
            mysqlc.execute("update {} set loss=loss+1
where player_name='{}'".format(boooooo,y))
            mysqlc.execute("update {} set
games_played=games_played+1 where
player_name='{}'".format(boooooo,y))
            mysqlcon.commit()
          elif c=="3":
            mysqlc.execute("update {} set
games_played=games_played+1 where
player_name='{}'".format(boooooo,x))
            mysqlc.execute("update {} set
draw=draw+1 where
player_name='{}'".format(boooooo,x))
```

```
              mysqlc.execute("update {} set
points=points+1 where
player_name='{}'".format(boooooo,x))
              mysqlc.execute("update {} set
games_played=games_played+1 where
player_name='{}'".format(boooooo,y))
              mysqlc.execute("update {} set
draw=draw+1 where
player_name='{}'".format(boooooo,y))
              mysqlc.execute("update {} set
points=points+1 where
player_name='{}'".format(boooooo,y))
              mysqlcon.commit()
            print("OX"*30)
            mysqlc.execute("select * from {} order by
points desc".format(boooooo))
            print("THE TABLE IS AS
\n[PLAYER_ID,PLAYER_NAME,GAMES_PLAYED,WIN,LOSS,
DRAW,POINTS]")
            for i in range(0,players//groups):
              standings=mysqlc.fetchone()
```

```
                print(standings)
            print("OX"*30)
        gameno=0
    #-----------------------------------------------------
    print("OX"*30)
    print("welcome group winners for the ultimate win")
    print("the schedule is as followed")
    boomboom=0
    mysqlc.execute("create table champions(player_id char,player_name varchar(20),games_played int,win int,loss int,draw int,points int)")
    for ddoodod in range(1,groups+1):
        kiro="group"+str(ddoodod)
        mysqlc.execute("select player_name from {} where points=(select max(points) from {})".format(kiro,kiro))
        finalistssss=mysqlc.fetchone()
        mysqlc.execute("select player_id from {} where points=(select max(points) from {})".format(kiro,kiro))
```

```
            finalistsss=mysqlc.fetchone()
            mysqlc.execute("insert into
champions(player_id,player_name,games_played,win,los
s,draw,points)
values({},'{}',0,0,0,0,0)".format(finalistsss[0],finalistssss[0]
))
        mysqlc.execute("select * from champions")
        table=mysqlc.fetchall()
        print(table)
        notimes=0
        mysqlc.execute("select
g1.player_name,g2.player_name from champions g1
cross join champions g2 where
g1.player_name!=g2.player_name order by
g1.player_id+g2.player_id")
        schedul=mysqlc.fetchall()
        notimes+=1
        if notimes==1:
            schedule=schedul.copy()
        schedu=schedule.copy()
        oddsc=[]
```

```
evensc=[]
while len(schedu)!=0:
    hkhkhkh=random.randrange(0,len(schedule))
    if hkhkhkh%2==0:
        if schedule[hkhkhkh] not in evensc:
            evensc.append(schedule[hkhkhkh])
            schedu.remove(schedule[hkhkhkh])
    else:
        if schedule[hkhkhkh] not in oddsc:
            oddsc.append(schedule[hkhkhkh])
            schedu.remove(schedule[hkhkhkh])
fsch=[]
for element in evensc:
    fsch.append(element)
for element in oddsc:
    fsch.append(element)
fsch_fake.append(fsch)
print("THE SCHEDULE FOR GROUP CHAMPIONS IS AS FOLLOWED \n [player1,player2]")
```

```
            for something in fsch:
                print(something)
            print("OX"*30)
            #-----------------------------------
            totalgames=1
            for hhhhhhhh in range(1,groups+1):
                totalgames=totalgames*hhhhhhhh
            for jjjjjjjjjjj in range(1,totalgames+1):
                gameno+=1
                x=fsch[gameno-1][0]
                y=fsch[gameno-1][1]
                t=[["-","-","-"],["-","-","-"],["-","-","-"]]
#tic tac toe main matrice
                for llllllll in t:
                    print(llllllll)
                i=1
                c="3"
                while i<10 and c!="1" and c!="2":
                    if i%2==0:
```

```
print("its your turn",y)
k=input("enter column (1-3) -> ")
while k.isalpha()==True:
    k=input("Enter no. between 1 and 3 -> ")
while int(k)>3 or int(k)<1:
    k=input("Enter no. between 1 and 3 -> ")
m=input("enter row (1-3) -> ")
while m.isalpha()==True:
    m=input("Enter no. between 1 and 3 -> ")
while int(m)>3 or int(m)<1:
    m=input("Enter no. between 1 and 3 -> ")
while t[int(m)-1][int(k)-1]=="O" or t[int(m)-1][int(k)-1]=="X":
    k=input("Input unoccupied column (1-3) -> ")
    while k.isalpha()==True:
        k=input("Enter no. between 1 and 3 -> ")
    while  int(k)>3 or int(k)<1:
```

```
                                k=input("Enter no. between 1 and 3 ->
")
                            m=input("Input unoccupied row (1-3) ->
")
                            while m.isalpha()==True:
                                m=input("Enter no. between 1 and 3 ->
")
                            while int(m)>3 or int(m)<1:
                                m=input("Enter no. between 1 and 3 ->
")
                        t[int(m)-1][int(k)-1]="O"
#entering the sign in specified position
                        i+=1
                        print()
                        for a in range(0,3):
#printing matrice
                            for b in range(0,3):
                                print(t[a][b],end=" ")
                            print()
                        if t[0][0]==t[0][1]==t[0][2]=="O":
```

```
    c="2"
    print(y,"won")
elif t[1][0]==t[1][1]==t[1][2]=="O":
    c="2"
    print(y,"won")
elif t[2][0]==t[2][1]==t[2][2]=="O":
    c="2"
    print(y,"won")
elif t[0][0]==t[1][0]==t[2][0]=="O":
    c="2"
    print(y,"won")
elif t[0][1]==t[1][1]==t[2][1]=="O":
    c="2"
    print(y,"won")
elif t[0][2]==t[1][2]==t[2][2]=="O":
    c="2"
    print(y,"won")
elif t[0][0]==t[1][1]==t[2][2]=="O":
    c="2"
```

```
        print(y,"won")
    elif t[0][2]==t[1][1]==t[2][0]=="O":
        c="2"
        print(y,"won")
    elif i==9 and c!="1" and c!="2":
        c="3"
        print("draw it is")
    else:
        c="0"
        print()
        continue
else:   #chance of player 2
    print("its your turn",x)
    k=input("enter column (1-3) -> ")
    while k.isalpha()==True:
        k=input("Enter no. between 1 and 3 -> ")
    while int(k)>3 or int(k)<1:
        k=input("Enter no. between 1 and 3 -> ")
    m=input("enter row (1-3) -> ")
```

```
            while m.isalpha()==True:
                m=input("Enter no. between 1 and 3 -> ")
            while int(m)>3 or int(m)<1:
                m=input("Enter no. between 1 and 3 -> ")
            while t[int(m)-1][int(k)-1]=="O" or t[int(m)-1][int(k)-1]=="X":
                k=input("Input unoccupied column (1-3) -> ")
                while k.isalpha()==True:
                    k=input("Enter no. between 1 and 3 -> ")
                while  int(k)>3 or int(k)<1:
                    k=input("Enter no. between 1 and 3 -> ")
                m=input("Input unoccupied row (1-3) -> ")
                while m.isalpha()==True:
                    m=input("Enter no. between 1 and 3 -> ")
                while int(m)>3 or int(m)<1:
```

```
        m=input("Enter no. between 1 and 3 -> ")
t[int(m)-1][int(k)-1]="X"
i+=1
print("")
for a in range(0,3):
    for b in range(0,3):
        print(t[a][b],end=" ")
    print()
if t[0][0]==t[0][1]==t[0][2]=="X":
    c="1"
    print("Player",x,"won")
elif t[1][0]==t[1][1]==t[1][2]=="X":
    c="1"
    print("Player",x,"won")
elif t[2][0]==t[2][1]==t[2][2]=="X":
    c="1"
    print("Player",x,"won")
elif t[0][0]==t[1][0]==t[2][0]=="X":
    c="1"
    print("Player",x,"won")
elif t[0][1]==t[1][1]==t[2][1]=="X":
    c="1"
    print("Player",x,"won")
elif t[0][2]==t[1][2]==t[2][2]=="X":
    c="1"
    print("Player",x,"won")
elif t[0][0]==t[1][1]==t[2][2]=="X":
    c="1"
    print("Player",x,"won")
elif t[0][2]==t[1][1]==t[2][0]=="X":
    c="1"
    print("Player",x,"won")
elif i==9 and c!="1" and c!="2":
    c="3"
    print("draw it is")
else:
    c="0"
```

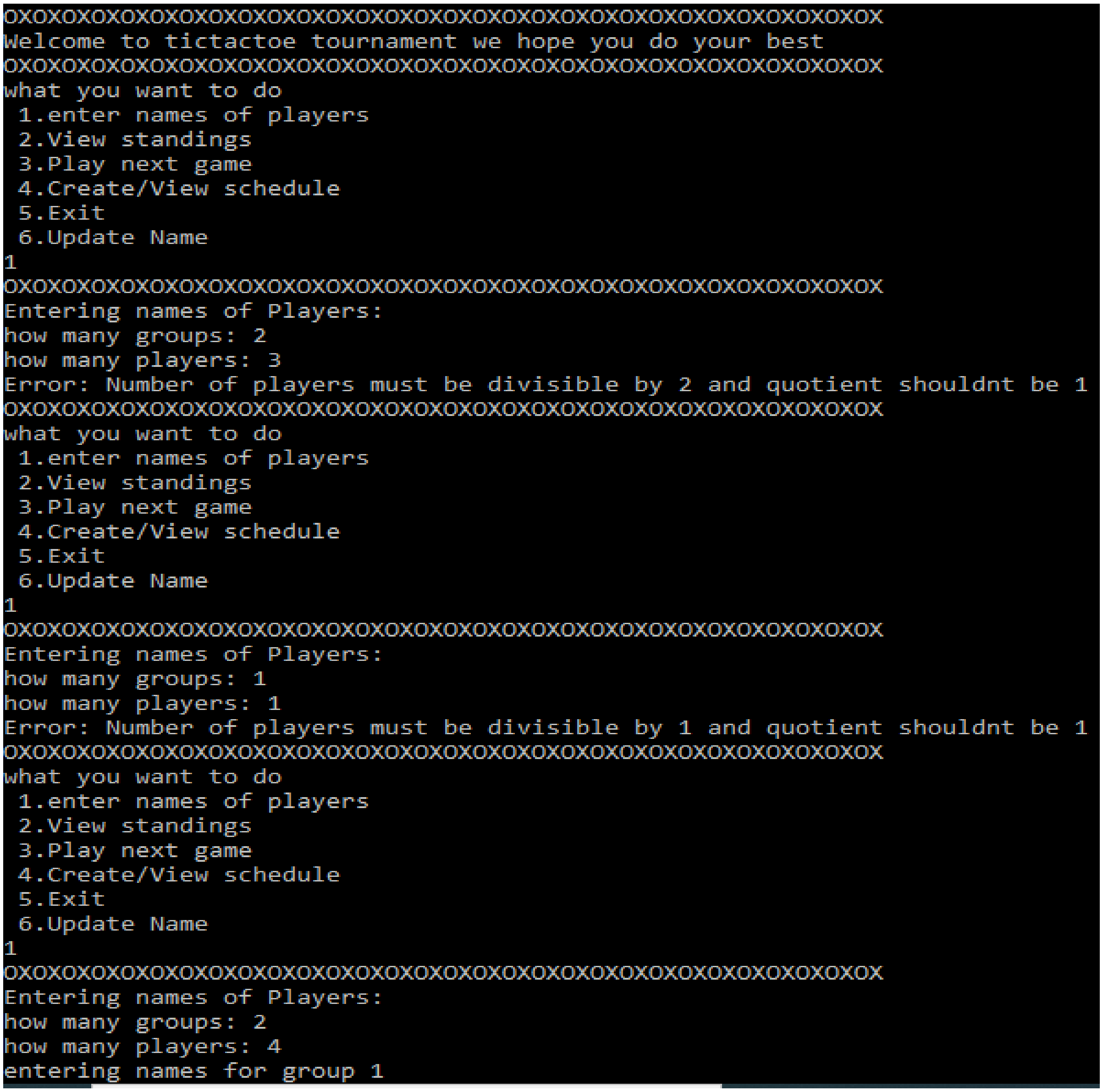
```
              print()
            continue
        if c=="2":
          mysqlc.execute("update champions set
games_played=games_played+1 where
player_name='{}'".format(y))
          mysqlc.execute("update champions set
win=win+1 where player_name='{}'".format(y))
          mysqlc.execute("update champions set
points=points+2 where player_name='{}'".format(y))
          mysqlc.execute("update champions set
loss=loss+1 where player_name='{}'".format(x))
          mysqlc.execute("update champions set
games_played=games_played+1 where
player_name='{}'".format(x))
          mysqlcon.commit()
        elif c=="1":
          mysqlc.execute("update champions set
games_played=games_played+1 where
player_name='{}'".format(x))
```

```
                mysqlc.execute("update champions set win=win+1 where player_name='{}'".format(x))
                mysqlc.execute("update champions set points=points+2 where player_name='{}'".format(x))
                mysqlc.execute("update champions set loss=loss+1 where player_name='{}'".format(y))
                mysqlc.execute("update champions set games_played=games_played+1 where player_name='{}'".format(y))
                mysqlcon.commit()
            elif c=="3":
                mysqlc.execute("update champions set games_played=games_played+1 where player_name='{}'".format(x))
                mysqlc.execute("update champions set draw=draw+1 where player_name='{}'".format(x))
                mysqlc.execute("update champions set points=points+1 where player_name='{}'".format(x))
                mysqlc.execute("update champions set games_played=games_played+1 where player_name='{}'".format(y))
```

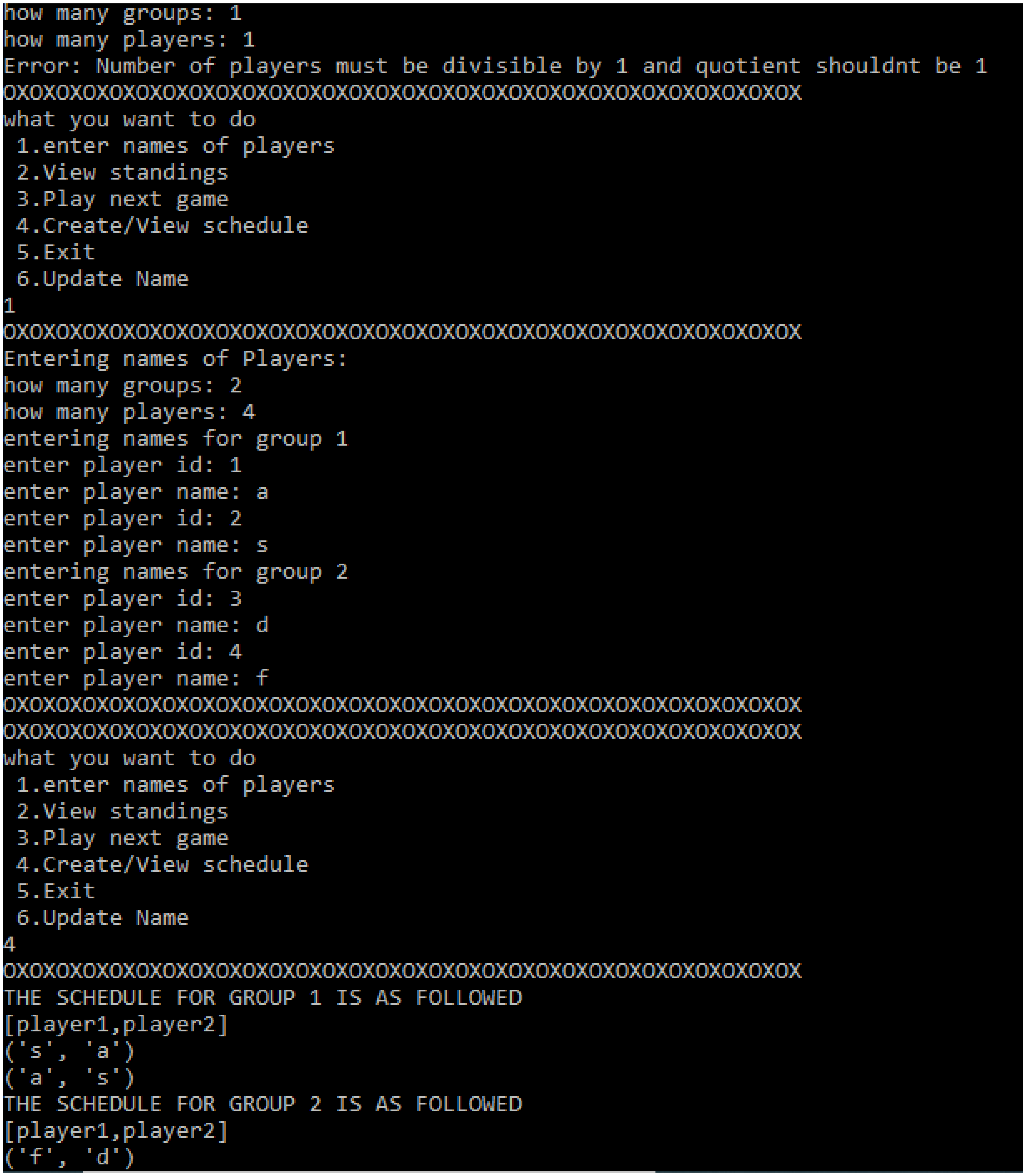
```
                    mysqlc.execute("update champions set draw=draw+1 where player_name='{}'".format(y))
                    mysqlc.execute("update champions set points=points+1 where player_name='{}'".format(y))
                    mysqlcon.commit()
            mysqlc.execute("select player_name from champions where points=(select max(points) from champions)")
            winner=mysqlc.fetchall()
            print("winner is",winner[0])
            print("OX"*30)

        print("OX"*30)

    elif choice==6:
        print("OX"*30)
        if groups==1:
            p=input("enter new name")
            q=input("enter original name")
```

```
      mysqlc.execute("update group1 set
player_name='{}' where player_name='{}'".format(p,q))
    print("OX"*30)
```

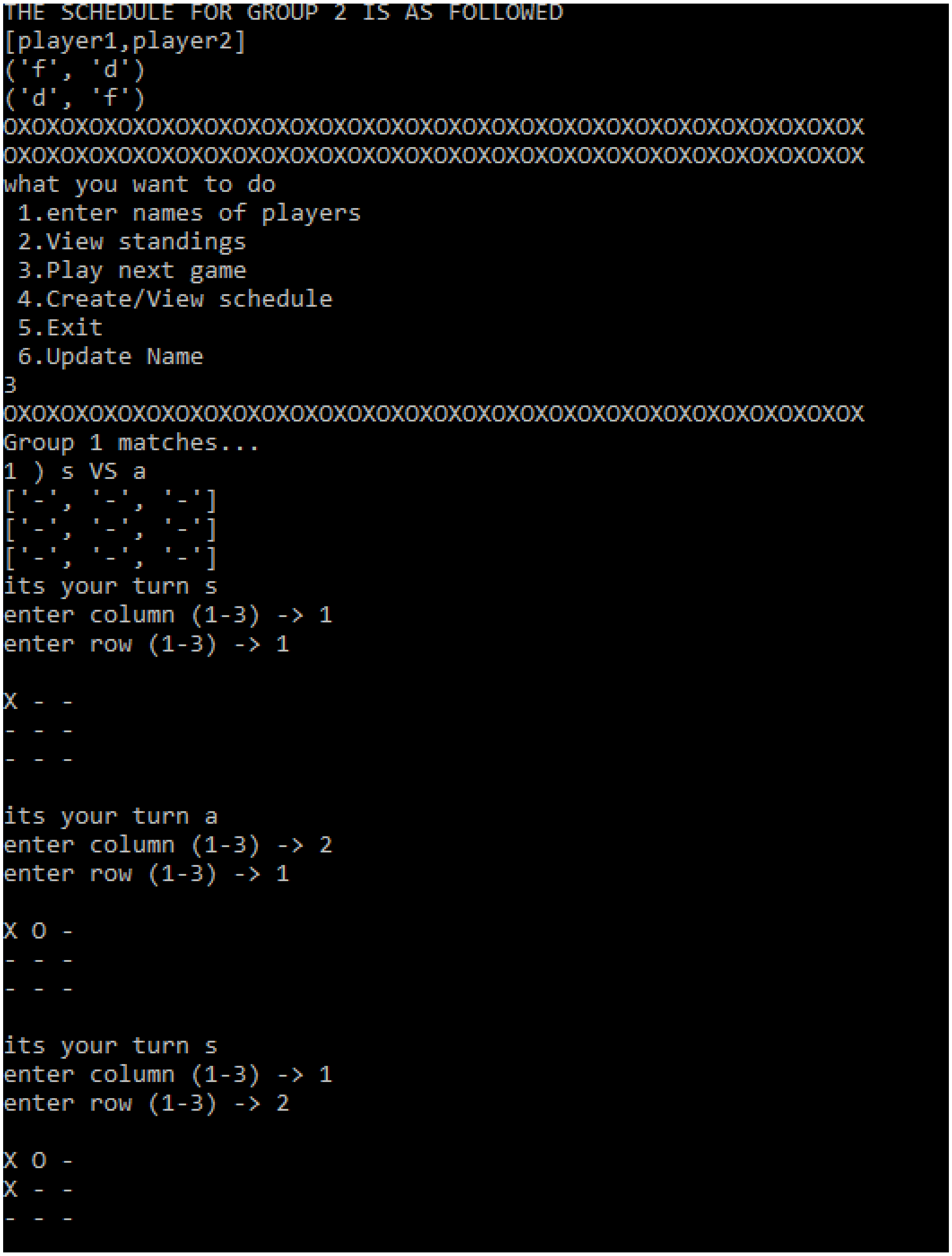
OUTPUT



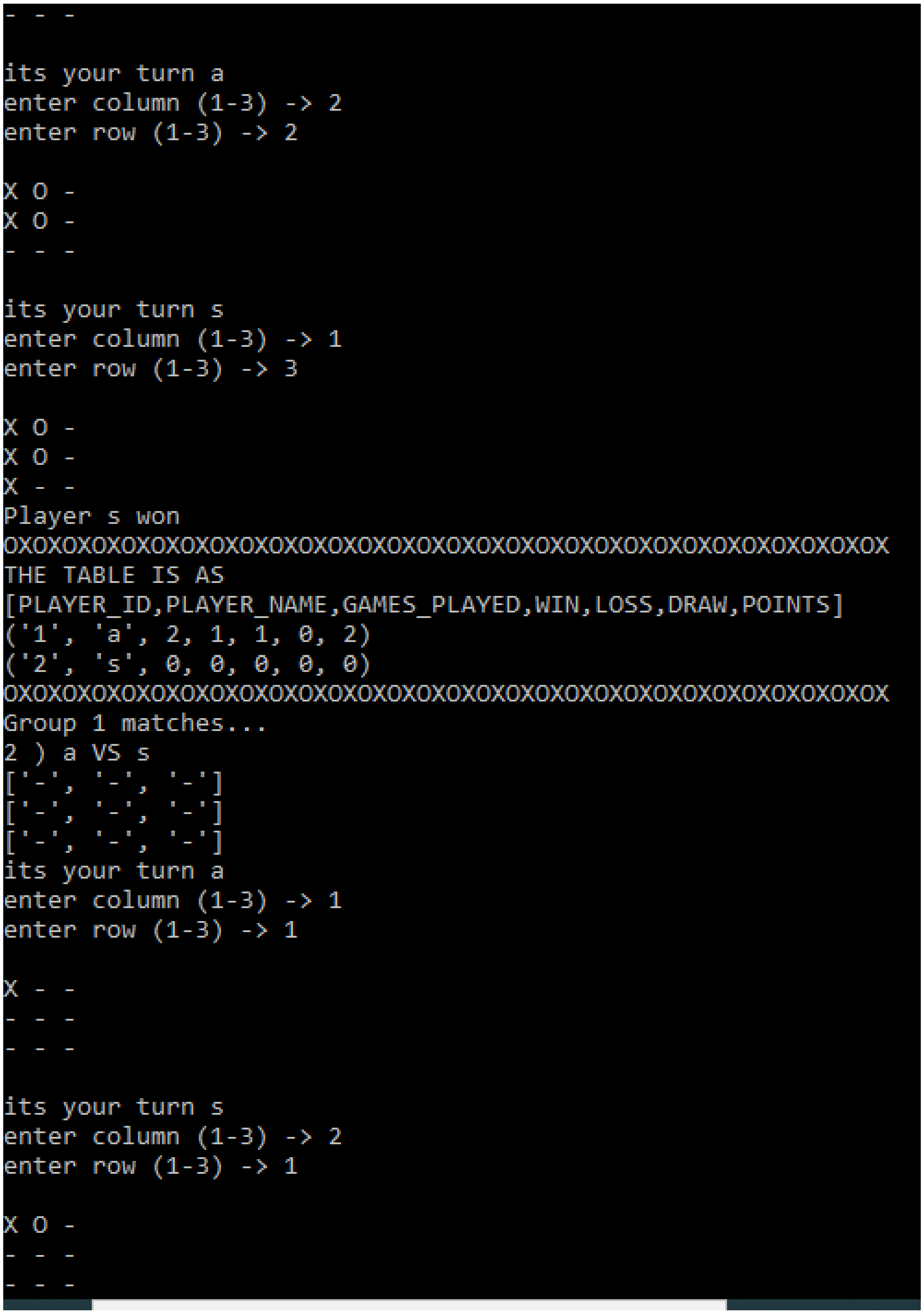
```
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
Welcome to tictactoe tournament we hope you do your best
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
what you want to do
 1.enter names of players
 2.View standings
 3.Play next game
 4.Create/View schedule
 5.Exit
 6.Update Name
1
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
Entering names of Players:
how many groups: 2
how many players: 3
Error: Number of players must be divisible by 2 and quotient shouldnt be 1
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
what you want to do
 1.enter names of players
 2.View standings
 3.Play next game
 4.Create/View schedule
 5.Exit
 6.Update Name
1
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
Entering names of Players:
how many groups: 1
how many players: 1
Error: Number of players must be divisible by 1 and quotient shouldnt be 1
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
what you want to do
 1.enter names of players
 2.View standings
 3.Play next game
 4.Create/View schedule
 5.Exit
 6.Update Name
1
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
Entering names of Players:
how many groups: 2
how many players: 4
entering names for group 1
```



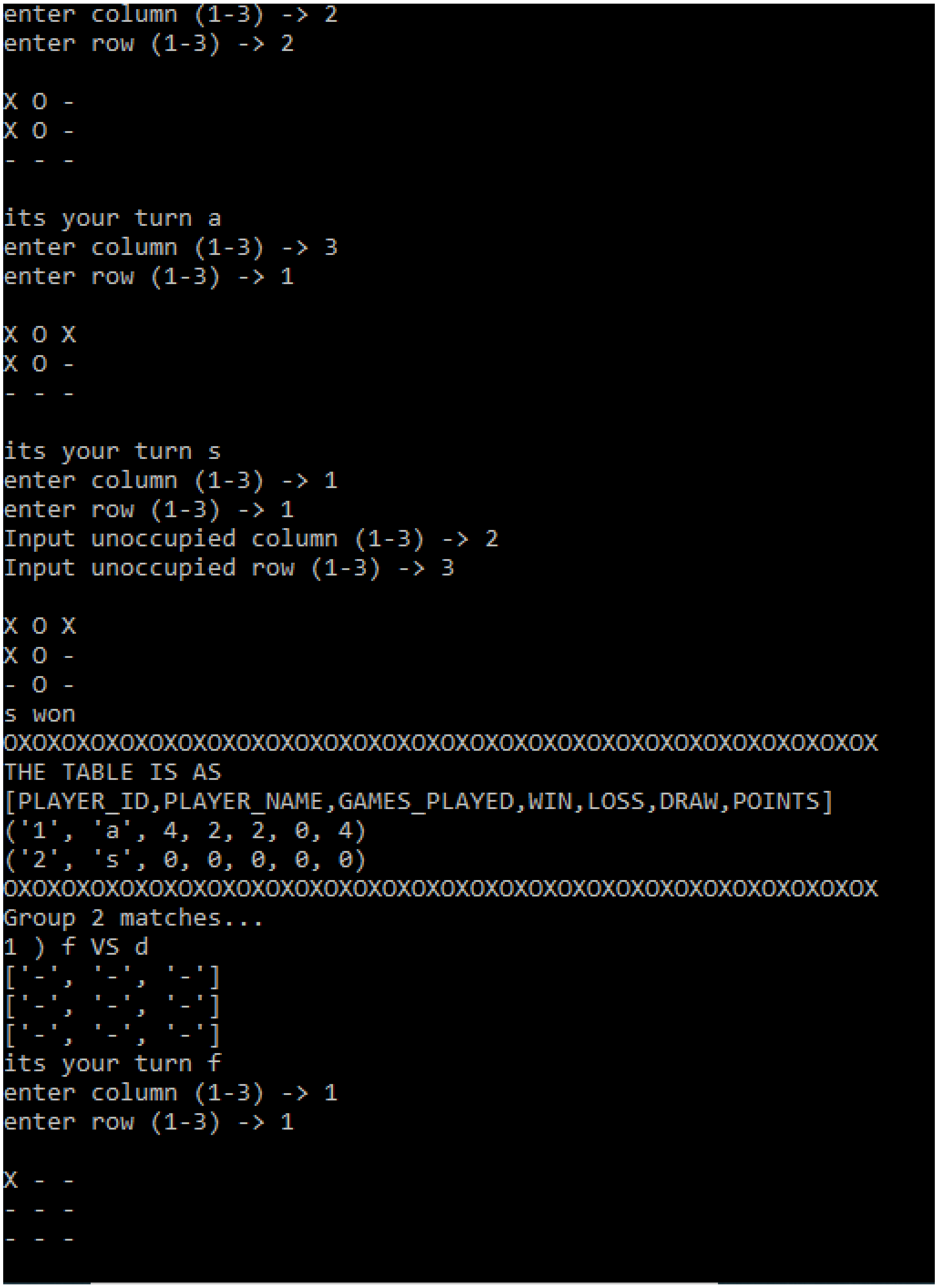
```
how many groups: 1
how many players: 1
Error: Number of players must be divisible by 1 and quotient shouldnt be 1
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
what you want to do
 1.enter names of players
 2.View standings
 3.Play next game
 4.Create/View schedule
 5.Exit
 6.Update Name
1
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
Entering names of Players:
how many groups: 2
how many players: 4
entering names for group 1
enter player id: 1
enter player name: a
enter player id: 2
enter player name: s
entering names for group 2
enter player id: 3
enter player name: d
enter player id: 4
enter player name: f
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
what you want to do
 1.enter names of players
 2.View standings
 3.Play next game
 4.Create/View schedule
 5.Exit
 6.Update Name
4
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
THE SCHEDULE FOR GROUP 1 IS AS FOLLOWED
[player1,player2]
('s', 'a')
('a', 's')
THE SCHEDULE FOR GROUP 2 IS AS FOLLOWED
[player1,player2]
('f', 'd')
```



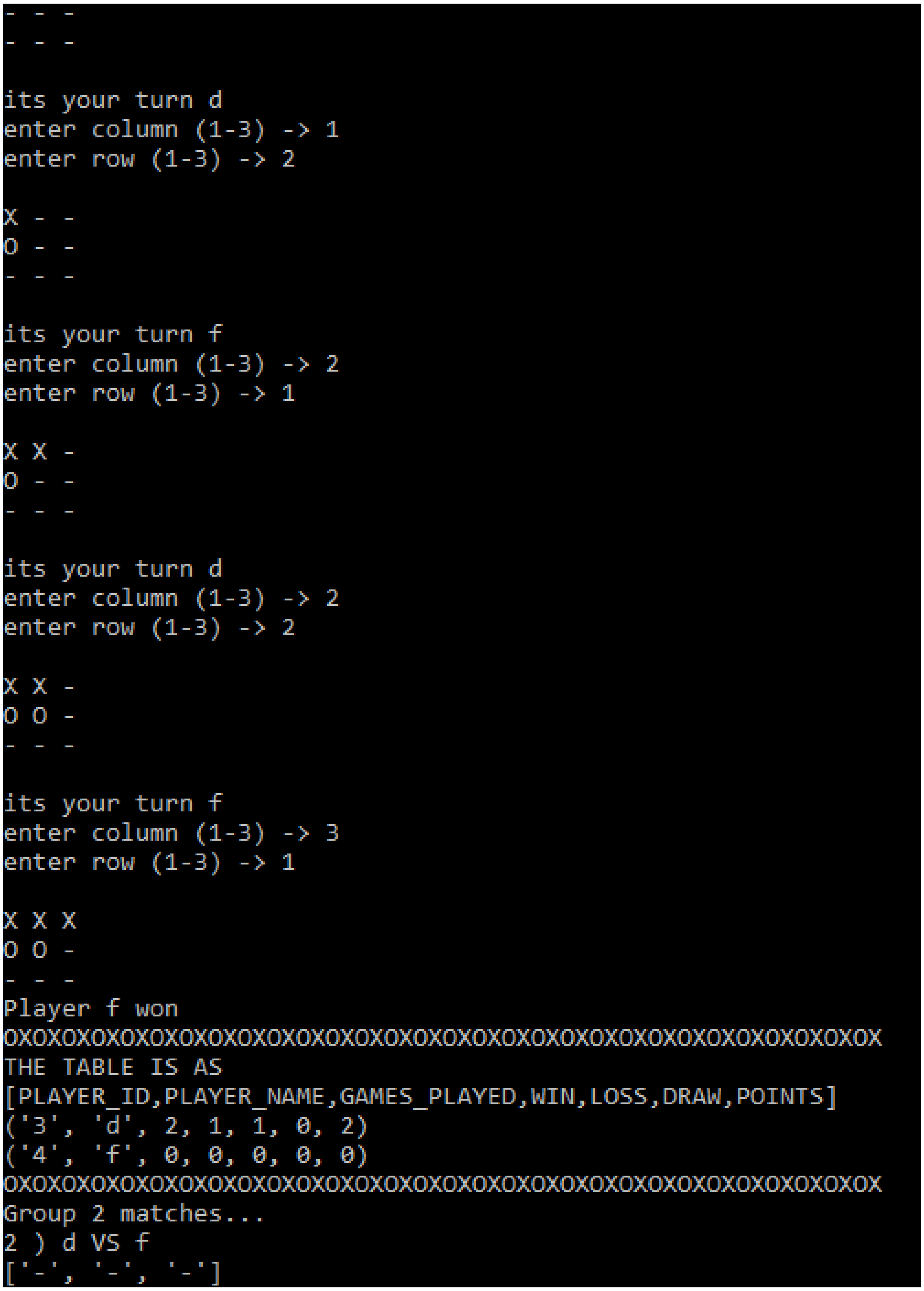
```
THE SCHEDULE FOR GROUP 2 IS AS FOLLOWED
[player1,player2]
('f', 'd')
('d', 'f')
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
what you want to do
 1.enter names of players
 2.View standings
 3.Play next game
 4.Create/View schedule
 5.Exit
 6.Update Name
3
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
Group 1 matches...
1 ) s VS a
['-', '-', '-']
['-', '-', '-']
['-', '-', '-']
its your turn s
enter column (1-3) -> 1
enter row (1-3) -> 1

X - -
- - -
- - -

its your turn a
enter column (1-3) -> 2
enter row (1-3) -> 1

X O -
- - -
- - -

its your turn s
enter column (1-3) -> 1
enter row (1-3) -> 2

X O -
X - -
- - -
```



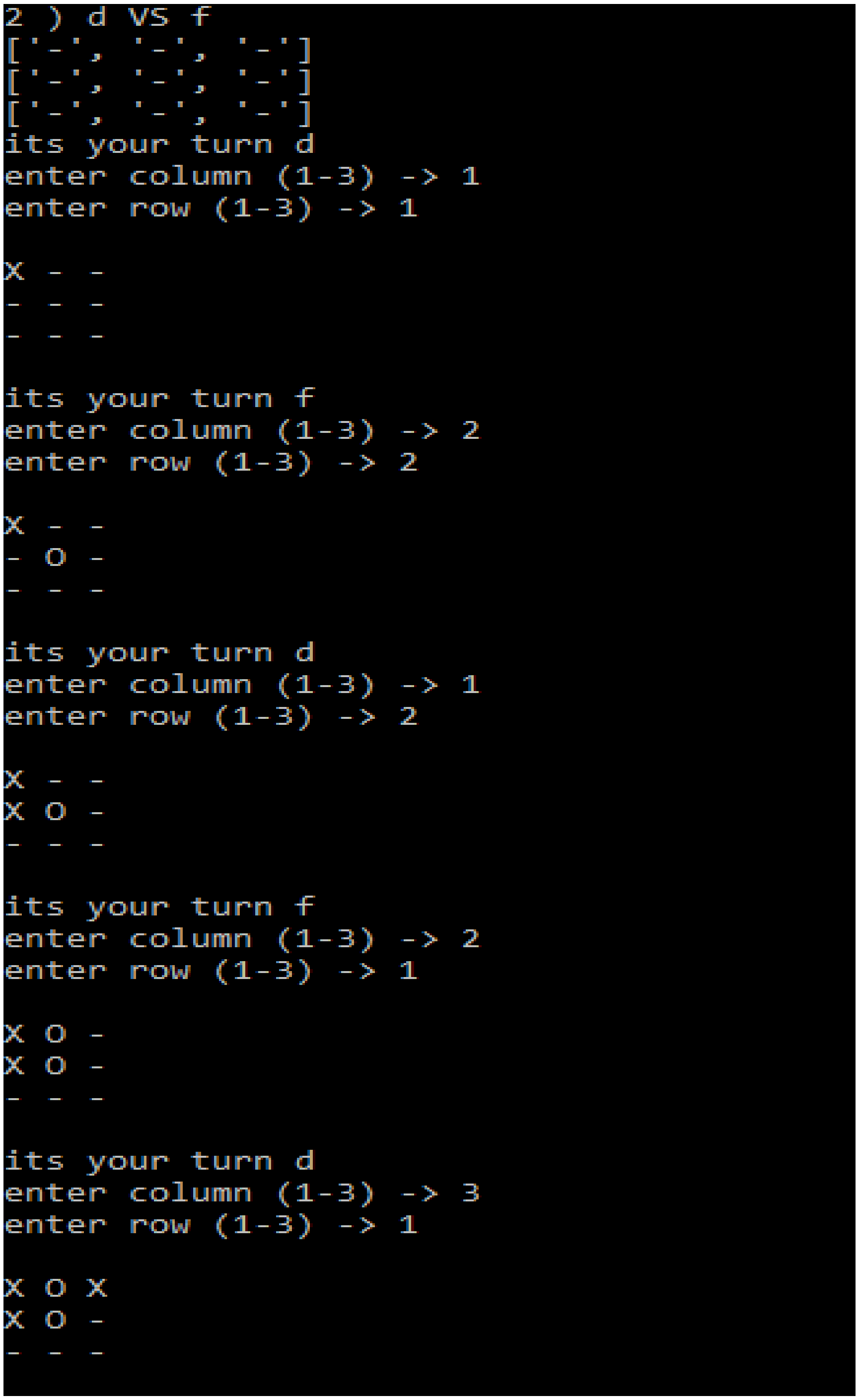
```
- - -

its your turn a
enter column (1-3) -> 2
enter row (1-3) -> 2

X O -
X O -
- - -

its your turn s
enter column (1-3) -> 1
enter row (1-3) -> 3

X O -
X O -
X - -
Player s won
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
THE TABLE IS AS
[PLAYER_ID,PLAYER_NAME,GAMES_PLAYED,WIN,LOSS,DRAW,POINTS]
('1', 'a', 2, 1, 1, 0, 2)
('2', 's', 0, 0, 0, 0, 0)
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
Group 1 matches...
2 ) a VS s
['-', '-', '-']
['-', '-', '-']
['-', '-', '-']
its your turn a
enter column (1-3) -> 1
enter row (1-3) -> 1

X - -
- - -
- - -

its your turn s
enter column (1-3) -> 2
enter row (1-3) -> 1

X O -
- - -
- - -
```



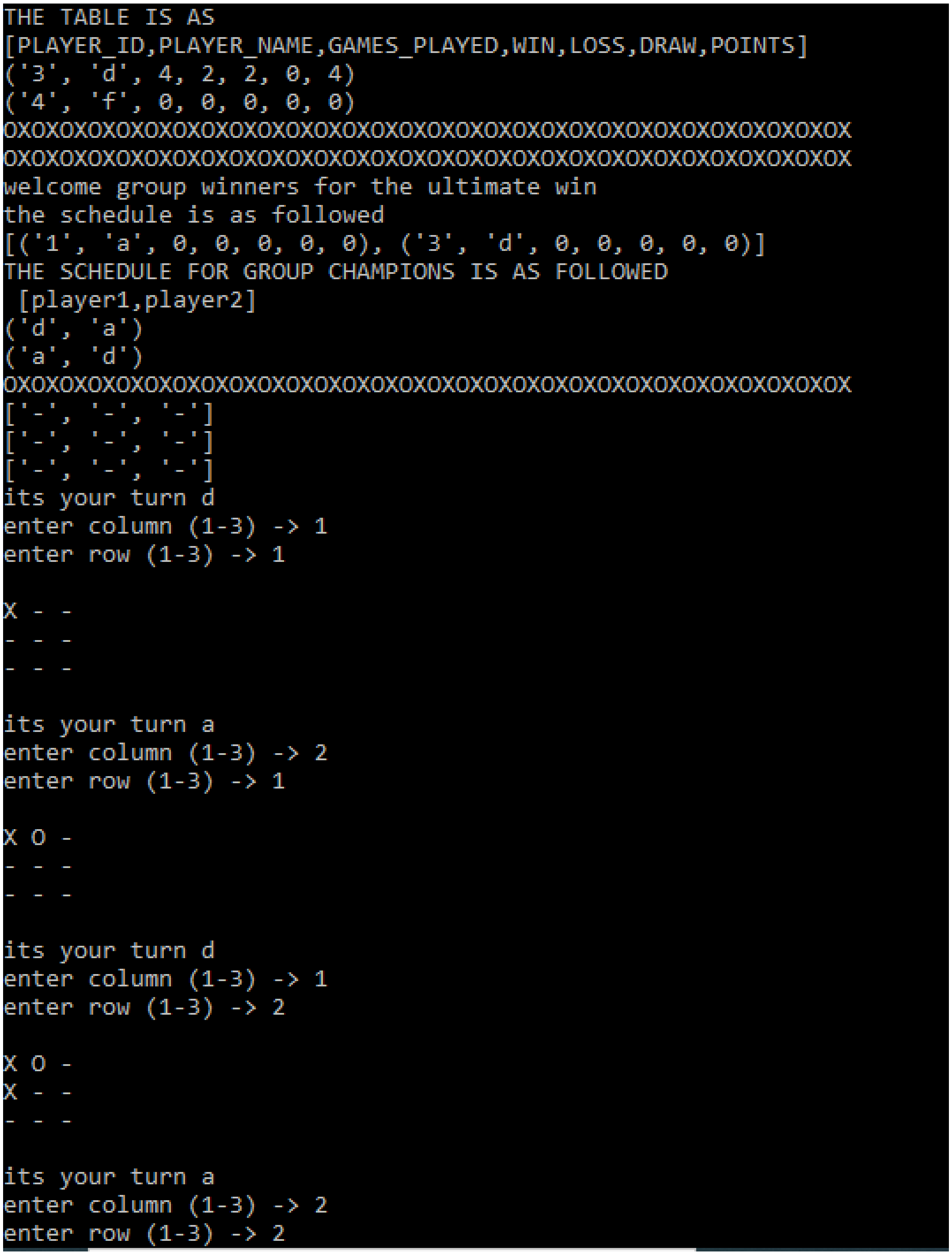
```
enter column (1-3) -> 2
enter row (1-3) -> 2

X O -
X O -
- - -

its your turn a
enter column (1-3) -> 3
enter row (1-3) -> 1

X O X
X O -
- - -

its your turn s
enter column (1-3) -> 1
enter row (1-3) -> 1
Input unoccupied column (1-3) -> 2
Input unoccupied row (1-3) -> 3

X O X
X O -
- O -
s won
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
THE TABLE IS AS
[PLAYER_ID,PLAYER_NAME,GAMES_PLAYED,WIN,LOSS,DRAW,POINTS]
('1', 'a', 4, 2, 2, 0, 4)
('2', 's', 0, 0, 0, 0, 0)
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
Group 2 matches...
1 ) f VS d
['-', '-', '-']
['-', '-', '-']
['-', '-', '-']
its your turn f
enter column (1-3) -> 1
enter row (1-3) -> 1

X - -
- - -
- - -
```



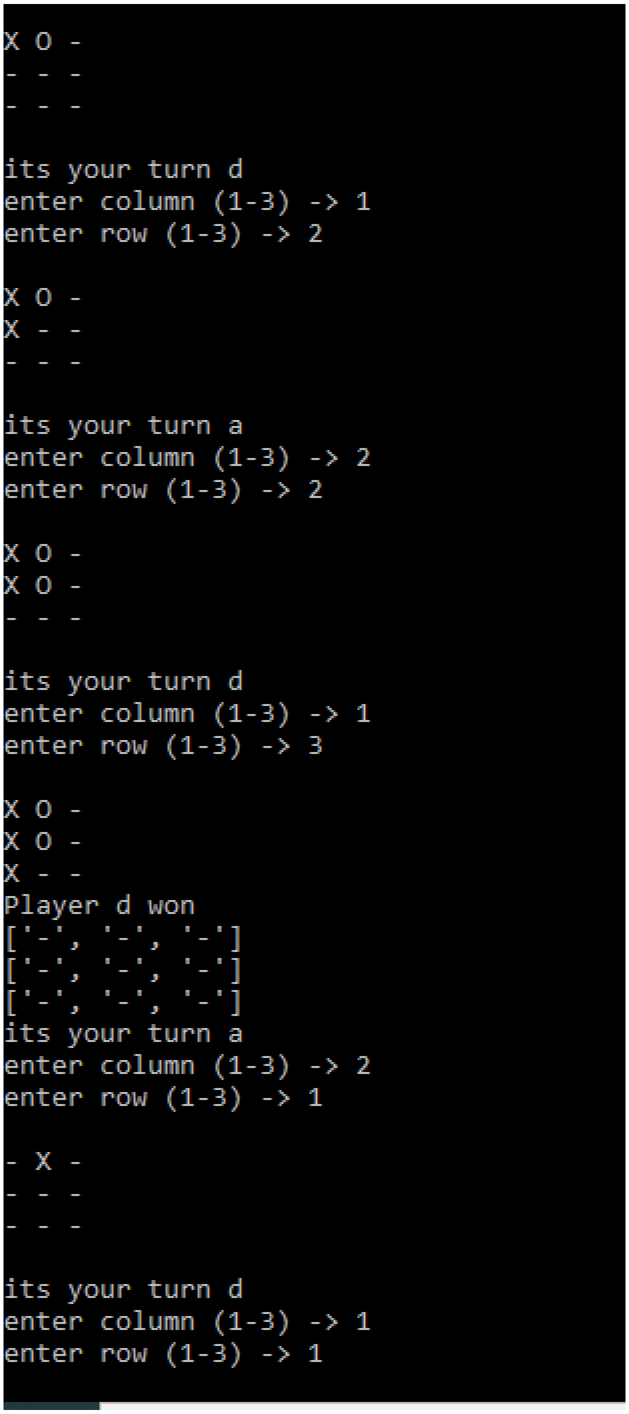
```
- - -
- - -

its your turn d
enter column (1-3) -> 1
enter row (1-3) -> 2

X - -
O - -
- - -

its your turn f
enter column (1-3) -> 2
enter row (1-3) -> 1

X X -
O - -
- - -

its your turn d
enter column (1-3) -> 2
enter row (1-3) -> 2

X X -
O O -
- - -

its your turn f
enter column (1-3) -> 3
enter row (1-3) -> 1

X X X
O O -
- - -
Player f won
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
THE TABLE IS AS
[PLAYER_ID,PLAYER_NAME,GAMES_PLAYED,WIN,LOSS,DRAW,POINTS]
('3', 'd', 2, 1, 1, 0, 2)
('4', 'f', 0, 0, 0, 0, 0)
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
Group 2 matches...
2 ) d VS f
['-', '-', '-']
```



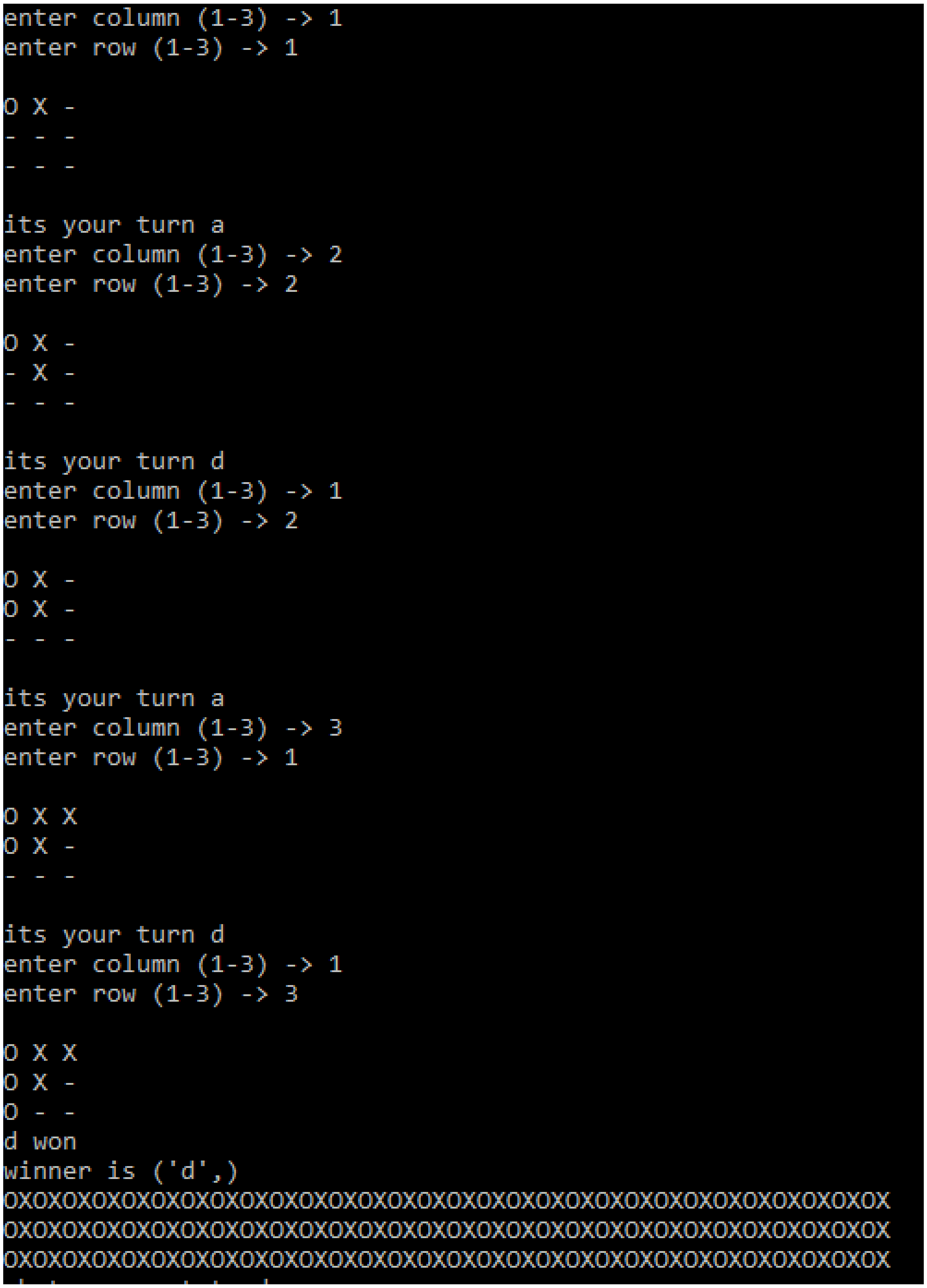
```
2 ) d VS f
['-', '-', '-']
['-', '-', '-']
['-', '-', '-']
its your turn d
enter column (1-3) -> 1
enter row (1-3) -> 1

X - -
- - -
- - -

its your turn f
enter column (1-3) -> 2
enter row (1-3) -> 2

X - -
- O -
- - -

its your turn d
enter column (1-3) -> 1
enter row (1-3) -> 2

X - -
X O -
- - -

its your turn f
enter column (1-3) -> 2
enter row (1-3) -> 1

X O -
X O -
- - -

its your turn d
enter column (1-3) -> 3
enter row (1-3) -> 1

X O X
X O -
- - -
```



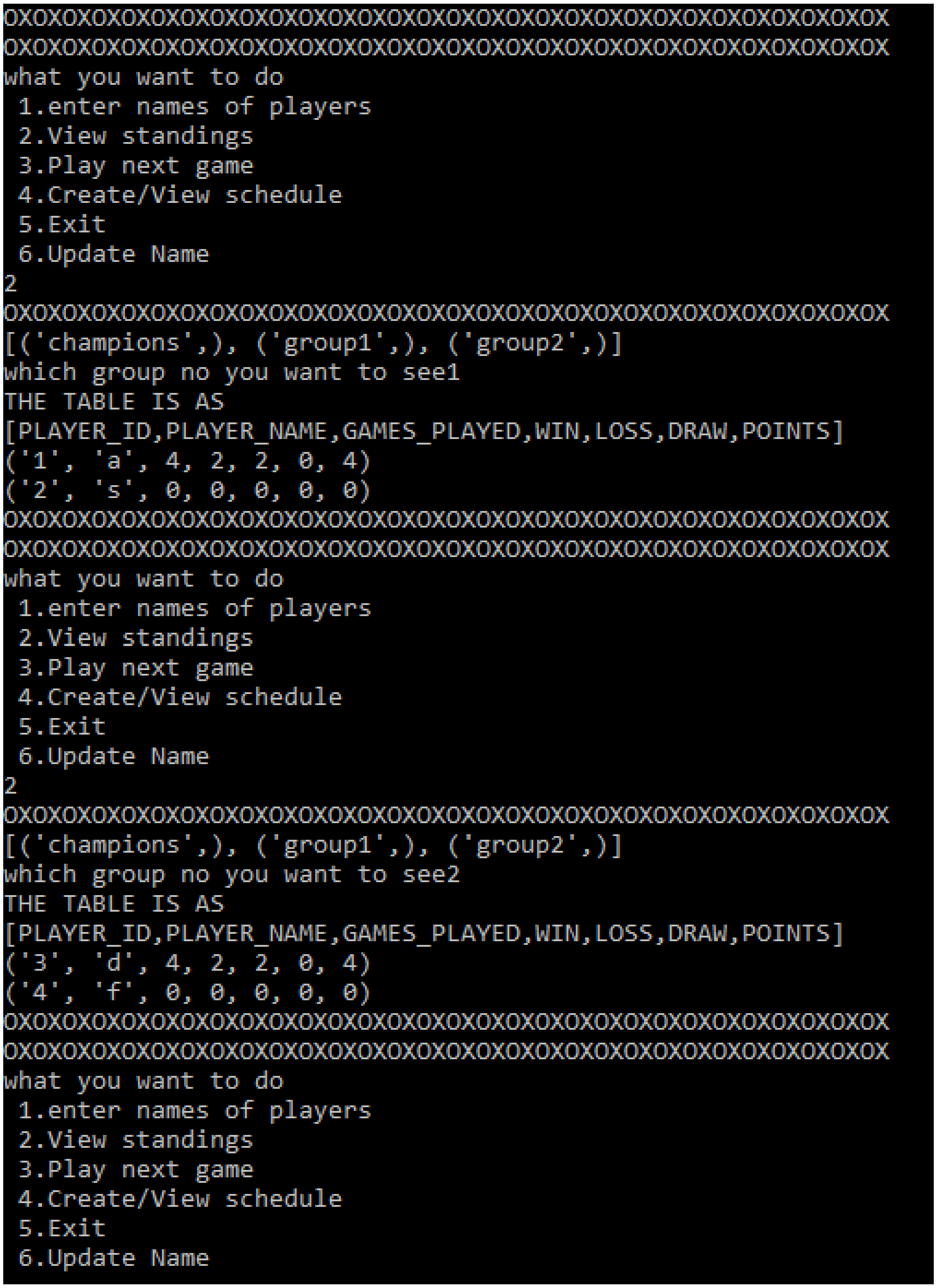
```
THE TABLE IS AS
[PLAYER_ID,PLAYER_NAME,GAMES_PLAYED,WIN,LOSS,DRAW,POINTS]
('3', 'd', 4, 2, 2, 0, 4)
('4', 'f', 0, 0, 0, 0, 0)
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
welcome group winners for the ultimate win
the schedule is as followed
[('1', 'a', 0, 0, 0, 0, 0), ('3', 'd', 0, 0, 0, 0, 0)]
THE SCHEDULE FOR GROUP CHAMPIONS IS AS FOLLOWED
 [player1,player2]
('d', 'a')
('a', 'd')
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
['-', '-', '-']
['-', '-', '-']
['-', '-', '-']
its your turn d
enter column (1-3) -> 1
enter row (1-3) -> 1

X - -
- - -
- - -

its your turn a
enter column (1-3) -> 2
enter row (1-3) -> 1

X O -
- - -
- - -

its your turn d
enter column (1-3) -> 1
enter row (1-3) -> 2

X O -
X - -
- - -

its your turn a
enter column (1-3) -> 2
enter row (1-3) -> 2
```



```
X O -
- - -
- - -

its your turn d
enter column (1-3) -> 1
enter row (1-3) -> 2

X O -
X - -
- - -

its your turn a
enter column (1-3) -> 2
enter row (1-3) -> 2

X O -
X O -
- - -

its your turn d
enter column (1-3) -> 1
enter row (1-3) -> 3

X O -
X O -
X - -
Player d won
['-', '-', '-']
['-', '-', '-']
['-', '-', '-']
its your turn a
enter column (1-3) -> 2
enter row (1-3) -> 1

- X -
- - -
- - -

its your turn d
enter column (1-3) -> 1
enter row (1-3) -> 1
```



```
enter column (1-3) -> 1
enter row (1-3) -> 1

O X -
- - -
- - -

its your turn a
enter column (1-3) -> 2
enter row (1-3) -> 2

O X -
- X -
- - -

its your turn d
enter column (1-3) -> 1
enter row (1-3) -> 2

O X -
O X -
- - -

its your turn a
enter column (1-3) -> 3
enter row (1-3) -> 1

O X X
O X -
- - -

its your turn d
enter column (1-3) -> 1
enter row (1-3) -> 3

O X X
O X -
O - -
d won
winner is ('d',)
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
```



```
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
what you want to do
 1.enter names of players
 2.View standings
 3.Play next game
 4.Create/View schedule
 5.Exit
 6.Update Name
2
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
[('champions',), ('group1',), ('group2',)]
which group no you want to see1
THE TABLE IS AS
[PLAYER_ID,PLAYER_NAME,GAMES_PLAYED,WIN,LOSS,DRAW,POINTS]
('1', 'a', 4, 2, 2, 0, 4)
('2', 's', 0, 0, 0, 0, 0)
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
what you want to do
 1.enter names of players
 2.View standings
 3.Play next game
 4.Create/View schedule
 5.Exit
 6.Update Name
2
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
[('champions',), ('group1',), ('group2',)]
which group no you want to see2
THE TABLE IS AS
[PLAYER_ID,PLAYER_NAME,GAMES_PLAYED,WIN,LOSS,DRAW,POINTS]
('3', 'd', 4, 2, 2, 0, 4)
('4', 'f', 0, 0, 0, 0, 0)
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
OXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOXOX
what you want to do
 1.enter names of players
 2.View standings
 3.Play next game
 4.Create/View schedule
 5.Exit
 6.Update Name
```

ENHANCEMENTS

Though we tried our level best given the things that we had learnt in class XII and XI but still there were some thing that we hoped would be included in our project with few of them listed below.

- Graphical viewpoint of the game
- Eliminator type game mode
- Entry of computer as participant in the tournament

HARDWARES AND SOFTWARES USED

- Python 3.7
- MySQL 8.0
- Device Name: HP348
- Processor: Intel® Core™ i5-6200U CPU @ 2.30GHz 2.40 GHz
- Installed Ram: 8.00 GB
- System Type: 64-bit Operating System, x64 based processor
- Windows: Windows 10 Education

BIBLIOGRAPHY

- SUMITA ARORA (Computer Science with Python XII)
- SUMITA ARORA (Computer Science with Python XI)
- www.geekforgeeks.com
- www.ubiq.co