Assigment 4:

Introduction about my code:

My code is a code with four stages. The first stage receives "input" from the given files. The second stage processes the data according to the game style. The third stage determines how the game will continue. The last stage is for making tables. I run the functions in this order, even if they are not in the order I mentioned in the code.

By reading the layout of the ships in the first lines of the code, I understand the coordinates of the ships by keeping the coordinates in the dictionaries. At the same time, I create using the "-" signs that will be necessary for tabulation. I use dictionary definitions and 2 more loops that read the "Player.in" files after 2 "for loops". These loops allow me to save the moves that the players will play in the dictionary and store them for the future.

```
import sys
try:


        f1=open(sys.argv[1],"r")
        f2=open(sys.argv[2],"r")
        f3= open("Battleship.out.txt","w")
        f4=open(sys.argv[3],"r")
        f5=open(sys.argv[4],"r")
        f6=open("OptionalPlayer1.txt","r")
        f7=open("OptionalPlayer2.txt","r")

        # The if elses I put for debugging are useful for detecting the IO error.
        if sys.argv[1]!="Player1.txt":
            print("IOError: input file(s) Player1.txt is/are not
        reachable.")
            f3.write("IOError: input file(s) Player1.txt is/are not
        reachable."+"\n")
            raise IOError
        if sys.argv[2]!="Player2.txt":
            print("IOError: input file(s) Player2.txt is/are not
        reachable.")
            f3.write("IOError: input file(s) Player2.txt is/are not
        reachable."+"\n")
            raise IOError
        if sys.argv[3]!="Player1.in":
            print("IOError: input file(s) Player1.in is/are not reachable.")
            f3.write("IOError: input file(s) Player1.in is/are not
        reachable."+"\n")
            raise IOError
        if sys.argv[4]!="Player2.in":
            print("IOError: input file(s) Player2.in is/are not reachable.")
            f3.write("IOError: input file(s) Player2.in is/are not
        reachable."+"\n")
            raise IOError
        lines7= f7.readlines()
        lines6= f6.readlines()
```

```
lines = f1.readlines()
lines2 = f2.readlines()
lines4 = f4.readlines()
lines5 = f5.readlines()
data1 = {}
data2 = {}
data5 = {}
data4 = {}
karaliste1={}
karaliste2={}
alfabe = []
a=0
s=0
list2 = []
list3 = []
list4 = []
list5 = []
list6 = []
for g in lines:
    a+=1
```

a is a counter I use to show the number of rows

```
    list1 = []
    g = g.strip("\n")
    for c in g.split(";"):
        s += 1
        list1.append("-")
        if s%10==0:
            ff = str((s-1) // 10+1) + "," + str(10)
        else:
            ff = str((s-1) // 10+1) + "," + str(s % 10)
        if c == "C":
            list2.append(ff)
        if c == "B":
            list3.append(ff)
        if c == "P":
            list4.append(ff)
        if c == "S":
            list5.append(ff)
        if c == "D":
            list6.append(ff)
## The "s" counter here is a counter I created to find the digits of
the coordinate.


    data1[a]=list1
karaliste1["C"]=list2
karaliste1["S"]=list5
karaliste1["D"]=list6

a=0
```

```
s=0
list2 = []
list3 = []
list4 = []
list5 = []
list6 = []
for g in lines2:
    a += 1
    list1 = []
    g = g.strip("\n")
    for c in g.split(";"):
        s += 1
        list1.append("-")
        if s % 10 == 0:
            ff = str((s - 1) // 10 + 1) + "," + str(10)
        else:
            ff = str((s - 1) // 10 + 1) + "," + str(s % 10)
        if c == "C":
            list2.append(ff)
        if c == "B":
            list3.append(ff)
        if c == "P":
            list4.append(ff)
        if c == "S":
            list5.append(ff)
        if c == "D":
            list6.append(ff)

    data2[a] = list1
karaliste2["C"] = list2
karaliste2["S"] = list5
karaliste2["D"] = list6
# I put the "indexes" in the lines I "split" in accordance with the
input format into the dictionary
for g in lines4:
    s=1
    list = []
    a =g[:-1].split(";")
    for i in a:
        list2=[]
        list.append(i.split(",")[0]+","+str(ord(i[-1])-ord("A")+1))
    data4=list
    s+=1

for g in lines5:
    s=1
    list = []
    a =g[:-1].split(";")
    for i in a:
        list2=[]
```

```
            list.append(i.split(",")[0]+","+str(ord(i[-1])-ord("A")+1))
        data5=list
        s+=1
```

After this stage comes the functions that play the game.

At this stage, the "-" sign becomes "0" or "x" depending on the coordinate given in the "data1,2" dictionaries I prepared for tabulation. And "karaliste1,2", the dictionaries containing the coordinates of the ships

takes it out.

```
def oyna4(c,d,o,ü):
    falag=True
    d=d-1
    for j in karaliste2:
        if str(c)+","+str(d+1) in karaliste2[j]:
            karaliste2[j].remove(str(c) + "," + str(d + 1))
            if data2[c][d]=="-":
                try:
                    data6.remove(str(c) + "," + str(d + 1))
                except ValueError:
                    pass
                data2[c][d]="X"
        elif j == "P4":
            if c <= 10:
                if d <=10:
                    if data2[c][d] == "-":
                        data2[c][d] = "O"
```

```
# Since we have to prepare a table at the end of each move, I call the table making function among the "oyna" functions I call for each move.

    tabloyap(c, d,o,ü)
```

```
def oyna5(c,d,o,ü):
    d=d-1
    for j in karaliste1:
        if str(c)+","+str(d+1) in karaliste1[j]:
            karaliste1[j].remove(str(c)+","+str(d+1))
            if data1[c][d]=="-":
                try:
                    data6.remove(str(c) + "," + str(d + 1))
                except ValueError:
                    pass
                data1[c][d]="X"
        elif j == "P4":
            if c <= 10:
                if d <=10:
                    if data1[c][d] == "-":
                        data1[c][d] = "O"
```

```
    tabloyap(c,d,o,ü)
```

The "gur" and "dur" functions that I use to determine the number of ships allow me to find the number of battleship and patrol ships, respectively.

```
def gur(a):
    sas=0
    for i in a:
        if i[0] == "B" and a[i] == []:
            sas+=1
    return sas
def dur(a):
    sas=0
    for i in a:
        if i[0] == "P" and a[i] == []:
            sas+=1
    return sas
```

In this part, where I process the "optionalplayer.txt" file that we use for extracting the coordinates of the ships, I determine the coordinates with the texts in the file and enter these "inputs" into the "karalist1" dictionary, where I keep the coordinates.

```
def vaycanına(i):
    sas = []
    cas=[]
    for g in i:
        d=g.split(",")
        a =int(d[0][3::])
        b= ord(d[1][0])-ord("A")+1
        if g[0]=="B":
# the last 4th letter here is actually the letters I use to detect the words right down left up
            if g[-4]=="f":
                das=[]
                for ı in range(4):
                    das.append(str(a)+","+str(int(b)-ı))
                sas.append(das)
            if g[-4] == "h":
                das=[]
                for ı in range(4):
                    das.append(str(a)+","+str(int(b)+ı))
                sas.append(das)
            if g[-4] == "w":
                das=[]
                for ı in range(0,4):
                    das.append(str(a+ı)+","+str(int(b)))
                sas.append(das)
```

```
            if g[-4] == "u":
                das=[]
                for ı in range(4):
                    das.append(str(a-ı)+","+str(int(b)))
                sas.append(das)
            karaliste1[g[0:2]]=das
```

With the for loop, I determine all the coordinates of the ship by shifting it up and down according to the length of the ship.

```
        elif g[0]=="P":
            if g[-4]=="f":
                for ı in range(2):
                    das.append(str(a)+","+str(int(b)-ı))
                cas.append(das)
            if g[-4] == "h":
                das=[]
                for ı in range(2):
                    das.append(str(a)+","+str(int(b)+ı))
                cas.append(das)
            if g[-4] == "w":
                das=[]
                for ı in range(2):
                    das.append(str(a+ı)+","+str(int(b)))
                cas.append(das)
            if g[-4] == "u":
                das=[]
                for ı in range(2):
                    das.append(str(a-ı)+","+str(int(b)))
                cas.append(das)
            karaliste1[g[0:2]]=das
vaycanına(lines6)
def vaycanına2(i):
    sas = []
    cas=[]
    for g in i:
        d=g.split(",")
        a =int(d[0][3::])
        b= ord(d[1][0])-ord("A")+1
        if g[0]=="B":
            if g[-4]=="f":
                das=[]
                for ı in range(4):
                    das.append(str(a)+","+str(int(b)-ı))
                sas.append(das)
            if g[-4] == "h":
                das=[]
                for ı in range(4):
                    das.append(str(a)+","+str(int(b)+ı))
                sas.append(das)
```

```
            if g[-4] == "w":
                das=[]
                for ı in range(0,4):
                    das.append(str(a+ı)+","+str(int(b)))
                sas.append(das)
            if g[-4] == "u":
                das=[]
                for ı in range(4):
                    das.append(str(a-ı)+","+str(int(b)))
                sas.append(das)
            karaliste2[g[0:2]]=das

        elif g[0]=="P":
            if g[-4]=="f":
                for ı in range(2):
                    das.append(str(a)+","+str(int(b)-ı))
                cas.append(das)
            if g[-4] == "h":
                das=[]
                for ı in range(2):
                    das.append(str(a)+","+str(int(b)+ı))
                cas.append(das)
            if g[-4] == "w":
                das=[]
                for ı in range(2):
                    das.append(str(a+ı)+","+str(int(b)))
                cas.append(das)
            if g[-4] == "u":
                das=[]
                for ı in range(2):
                    das.append(str(a-ı)+","+str(int(b)))
                cas.append(das)
            karaliste2[g[0:2]]=das
vaycanına2(lines7)
```

The table function allows to extract the tables of the instant players' boards according to the moves. After the desired information is written, the updated "-" "o" "x" letters in "data1,2" are written according to the lines I have recorded in the dictionary.

the number of ships is determined by whether the lists I keep in the "karaliste" function are filled or not.

Battleship and Patrol boat ships, on the other hand, are transferred to the print as number of hits by means of the "gur.dur" functions, which I am sorry to say, and the number of hits is revealed.

I take the coordinates from the dictionary where I keep the moves of the players through "data4,5" and reflect the coordinates played in that round as output.

```
#diffirences between “tabloyap” and “tabloyap2” “tabloyap” for player 1
“tabloyap2” for player2

def tabloyap(i,ı,o,ü):
    print("Player1\'s Move"+"\n")
    f3.write("Player1\'s Move"+"\n"+"\n")
    print("Round:"+str(10*o+ü),end="
")
    f3.write("Round:"+str(10*o+ü)+"
")
    print("Grid Size 10x10"+"\n")
    f3.write("Grid Size 10x10"+"\n"+"\n")
    print("Player1\'s Hidden board",end="       ")
    f3.write("Player1\'s Hidden board"+"       ")
    print("Player2\'s Hidden board"+"\n")
    f3.write("Player2\'s Hidden board"+"\n"+"\n")
    print("  A B C D E F G H I J",end="          ")
    f3.write("  A B C D E F G H I J"+"         ")
    print("A B C D E F G H I J")
    f3.write("A B C D E F G H I J"+"\n")
    for i in range(1,11):
        if i ==10:
            print(str(i)+" ".join(data1[i]),"       "+str(i)+"
".join(data2[i]))
            aaa=str(i)+" ".join(str(data1[i]))+"       "+str(i)+"
".join(str(data2[i]))+"\n"
            f3.write(aaa)
        else:
            print(i,end=" ")
            print(" ".join(data1[i]),"       "+str(i)+" "+"
".join(data2[i]))
            aaa=str(str(i)+" ".join(data1[i])+"       "+str(i)+"
".join(data2[i])+" "+"\n")
            f3.write(aaa)
    if len(karaliste1["C"])==0:
        print("Carrier: "+"X",end="                  ")
        f3.write("Carrier: "+"X"+"                 ")
        if len(karaliste2["C"]) == 0:
            print("Carrier: " + "X")
            f3.write("Carrier: " + "X"+"\n")
        else:
            print("Carrier: " + "-")
            f3.write("Carrier: " + "-"+"\n")
    else:
        print("Carrier: "+"-",end="                  ")
        f3.write("Carrier: "+"-"+"                 ")
        if len(karaliste2["C"]) == 0:
            print("Carrier: " + "X")
            f3.write("Carrier: " + "X"+"\n")
        else:
```

```
            print("Carrier: " + "-")
            f3.write("Carrier: " + "-"+"\n")
    print("Battleship: "+"X"*gur(karaliste1)+"-"*(2-gur(karaliste1)),end="
")
    f3.write("Battleship: "+"X"*gur(karaliste1)+"-"*(2-gur(karaliste1))+"
")
    print("Battleship: "+"X"*gur(karaliste2)+"-"*(2-gur(karaliste2)))
    f3.write("Battleship: "+"X"*gur(karaliste1)+"-"*(2-
gur(karaliste1))+"\n")
    if len(karaliste1["D"])==0:
        print("Destroyer: "+"X",end="                 ")
        f3.write("Destroyer: "+"X"+"                 ")
        if len(karaliste2["D"]) == 0:
            print("Destroyer: " + "X")
            f3.write("Destroyer: " + "X"+"\n")
        else:
            print("Destroyer: " + "-")
            f3.write("Destroyer: " + "-"+"\n")

    else:
        print("Destroyer: "+"-",end="                 ")
        f3.write("Destroyer: "+"-"+"                 ")
        if len(karaliste2["D"]) == 0:
            print("Destroyer: " + "X")
            f3.write("Destroyer: " + "X"+"\n")
        else:
            print("Destroyer: " + "-")
            f3.write("Destroyer: " + "-"+"\n")
    if len(karaliste1["S"])==0:
        f3.write("Submarine: "+"X"+"                 ")
        print("Submarine: "+"X",end="                 ")
        if len(karaliste2["S"]) == 0:
            print("Submarine: " + "X")
            f3.write("Submarine: " + "X"+"\n")
        else:
            print("Submarine: " + "-")
            f3.write("Submarine: " + "-"+"\n")
    else:
        print("Submarine: "+"-",end="                 ")
        f3.write("Submarine: "+"-"+"                 ")
        if len(karaliste2["S"]) == 0:
            print("Submarine: " + "X")
            f3.write("Submarine: " + "X"+"\n")
        else:
            print("Submarine: " + "-")
            f3.write("Submarine: " + "-"+"\n")
    print("Patrol Boat: "+"X"*dur(karaliste1)+"-"*(4-
dur(karaliste1)),end="            ")
    f3.write("Patrol Boat: "+"X"*dur(karaliste1)+"-"*(4-dur(karaliste1))+"
")
```

```
    print("Patrol Boat: "+"X"*dur(karaliste2)+"-"*(4-
dur(karaliste2)),"\n")
    f3.write("Patrol Boat: "+"X"*dur(karaliste2)+"-"*(4-
dur(karaliste2)),"\n"+"\n")
    print(str(c)+","+str(d))
    f3.write(str(c)+","+str(d)+"\n")
def tabloyap2(i,ı,o,ü):
    print("Player1\'s Move"+"\n")
    f3.write("Player1\'s Move"+"\n"+"\n")
    print("Round:"+str(10*o+ü),end="
")
    f3.write("Round:"+str(10*o+ü)+"
")
    print("Grid Size 10x10"+"\n")
    f3.write("Grid Size 10x10"+"\n"+"\n")
    print("Player1\'s Hidden board",end="       ")
    f3.write("Player1\'s Hidden board"+"       ")
    print("Player2\'s Hidden board"+"\n")
    f3.write("Player2\'s Hidden board"+"\n"+"\n")
    print("  A B C D E F G H I J",end="         ")
    f3.write("  A B C D E F G H I J"+"         ")
    print("A B C D E F G H I J")
    f3.write("A B C D E F G H I J"+"\n")
    for i in range(1,11):
        if i ==10:
            print(str(i)+" ".join(data1[i]),"       "+str(i)+"
".join(data2[i]))
            f3.write(str(i)+" ".join(data1[i])+"       "+str(i)+"
".join(data2[i])+"\n")
        else:
            print(i,end=" ")
            print(" ".join(data1[i]),"       "+str(i)+" "+"
".join(data2[i]))
            f3.write(str(i)+" ".join(data1[i])+"       "+str(i)+"
".join(data2[i])+"\n")

    if len(karaliste1["C"])==0:
        print("Carrier: "+"X",end="                  ")
        f3.write("Carrier: "+"X"+"                  ")
        if len(karaliste2["C"]) == 0:
            print("Carrier: " + "X")
            f3.write("Carrier: " + "X"+"\n")
        else:
            print("Carrier: " + "-")
            f3.write("Carrier: " + "-"+"\n")
    else:
        print("Carrier: "+"-",end="                  ")
        f3.write("Carrier: "+"-"+"                  ")
        if len(karaliste2["C"]) == 0:
            print("Carrier: " + "X")
```

```
            f3.write("Carrier: " + "X"+"\n")
        else:
            print("Carrier: " + "-")
            f3.write("Carrier: " + "-"+"\n")
    print("Battleship: "+"X"*gur(karaliste1)+"-"*(2-gur(karaliste1)),end="
")
    f3.write("Battleship: "+"X"*gur(karaliste1)+"-"*(2-gur(karaliste1))+"
")
    print("Battleship: "+"X"*gur(karaliste2)+"-"*(2-gur(karaliste2)))
    f3.write("Battleship: "+"X"*gur(karaliste1)+"-"*(2-
gur(karaliste1))+"\n")
    if len(karaliste1["D"])==0:
        print("Destroyer: "+"X",end="                 ")
        f3.write("Destroyer: "+"X"+"                 ")
        if len(karaliste2["D"]) == 0:
            print("Destroyer: " + "X")
            f3.write("Destroyer: " + "X"+"\n")
        else:
            print("Destroyer: " + "-")
            f3.write("Destroyer: " + "-"+"\n")

    else:
        print("Destroyer: "+"-",end="                 ")
        f3.write("Destroyer: "+"-"+"                 ")
        if len(karaliste2["D"]) == 0:
            print("Destroyer: " + "X")
            f3.write("Destroyer: " + "X"+"\n")
        else:
            print("Destroyer: " + "-")
            f3.write("Destroyer: " + "-"+"\n")
    if len(karaliste1["S"])==0:
        f3.write("Submarine: "+"X"+"                 ")
        print("Submarine: "+"X",end="                 ")
        if len(karaliste2["S"]) == 0:
            print("Submarine: " + "X")
            f3.write("Submarine: " + "X"+"\n")
        else:
            print("Submarine: " + "-")
            f3.write("Submarine: " + "-"+"\n")
    else:
        print("Submarine: "+"-",end="                 ")
        f3.write("Submarine: "+"-"+"                 ")
        if len(karaliste2["S"]) == 0:
            print("Submarine: " + "X")
            f3.write("Submarine: " + "X"+"\n")
        else:
            print("Submarine: " + "-")
            f3.write("Submarine: " + "-"+"\n")
    print("Patrol Boat: "+"X"*dur(karaliste1)+"-"*(4-
dur(karaliste1)),end="            ")
```

```
    f3.write("Patrol Boat: "+"X"*dur(karaliste1)+"-"*(4-dur(karaliste1))+"
")
    print("Patrol Boat: "+"X"*dur(karaliste2)+"-"*(4-
dur(karaliste2)),"\n")
    f3.write("Patrol Boat: "+"X"*dur(karaliste2)+"-"*(4-
dur(karaliste2)),"\n"+"\n")
    print(str(c)+","+str(d))
    f3.write(str(c)+","+str(d)+"\n")
```

The last for loop is actually a loop that keeps the game flowing. Actually, since my code is baseless, I made it with 2 "for" loops. Instead, another method could be used, but unfortunately it's in this state in a hurry.

these for loops act on the basis of "i" and "i", which allows me to indicate which move we are at first, takes the index 10*i+i from the "data4,5" dictionaries and the coordinates corresponding to this index are thrown into the "oyna" functions and the function plays the game according to these coordinates.

```
falag = True
for i in range(10):
    for ı in range(10):
        falag=True
        if falag==True:
            if 10*i+ı >= len(data4):
                break
            else:
                a = data4[10*i+ı]
                e=a.split(",")
                c = int(e[0])
                d = int(e[1])
                oyna4(c,d,i,ı)
                b = data5[10*i+ı]
                e=b.split(",")
                c = int(e[0])
                d = int(e[1])
                oyna5(c,d,i,ı)
                asas=0
                for s in karaliste1:
                    if karaliste1[s]==[]:
                        asas+=1
                afaf=0
                for s in karaliste2:
                    if s==[]:
                        afaf+=1
                n=0
                a=0
                if afaf==9:
                    n=1
```

```
                if asas==9:
                    a=1
                if a and n == 1:
                    print("Draw!")
                    f3.write("Draw"+"\n")
                    break

                if n==1:
                    print("Player1 Wins!")
                    f3.write("Player1 Wins!"+"\n")
                    break
                if a==1:
                    print("Player2 Wins!")
                    f3.write("Player2 Wins!"+"\n")
                    break
```

We determine the cauldron with the counters in this loop, these counters count whether the lists in the dictionary where the coordinates of the ships are kept are empty, and if all the lists are empty, then that player loses the game.

In the last part of my code, there is a for loop and excepts that update the "tableyap3" function "data1,2", which indicates the final state.

```
   for j in karaliste1:
        for g in karaliste1[j]:
            if karaliste1[j]!=[]:
                nn=g.split(",")
                c=int(nn[0])
                v=int(nn[1])
                data1[c][v-1]=j
                if j[0]=="B" or "P":
                    data2[c][v - 1] = j[0]
                else:
                    data2[c][v - 1] = j
    for j in karaliste2:
        for g in karaliste2[j]:
            if karaliste2[j]!=[]:
                nn=g.split(",")
                c=int(nn[0])
                v=int(nn[1])
                if j[0]=="B" or "P":
                    data2[c][v - 1] = j[0]
                else:
                    data2[c][v - 1] = j
    def tabloyap3():
        print("Final Information"+"\n")
        f3.write("Final Information"+"\n"+"\n")
        print("Grid Size 10x10"+"\n")
        f3.write("Grid Size 10x10"+"\n"+"\n")
        print("Player1\'s Hidden board",end="       ")
        f3.write("Player1\'s Hidden board"+"       ")
```

```
        print("Player2\'s Hidden board"+"\n")
        f3.write("Player2\'s Hidden board"+"\n"+"\n")
        print("  A B C D E F G H I J",end="         ")
        f3.write("  A B C D E F G H I J"+"         ")
        print(" A B C D E F G H I J")
        f3.write(" A B C D E F G H I J"+"\n")
        for i in range(1,11):
            if i ==10:
                print(str(i)+" ".join(data1[i]),"       "+str(i)+"
".join(data2[i]))
                f3.write(str(i)+" ".join(data1[i])+"       "+str(i)+"
".join(data2[i])+"\n")
            else:
                print(i,end=" ")
                print(" ".join(data1[i]),"       "+str(i)+" "+"
".join(data2[i]))
                f3.write(str(i)+" "+" ".join(data1[i])+"       "+str(i)+"
"+" ".join(data2[i])+"\n")
        if len(karaliste1["C"])==0:
            print("Carrier:     "+"X",end="               ")
            f3.write("Carrier:     "+"X"+"               ")
            if len(karaliste2["C"]) == 0:
                print("Carrier:     " + "X")
                f3.write("Carrier:     "    + "X"+"\n")
            else:
                print("Carrier:     " + "-")
                f3.write("Carrier:     " + "-"+"\n")
        else:
            print("Carrier:     "+"-",end="               ")
            f3.write("Carrier:     "+"-"+"               ")
            if len(karaliste2["C"]) == 0:
                print("Carrier:     " + "X")
                f3.write("Carrier:     " + "X"+"\n")
            else:
                print("Carrier:     " + "-")
                f3.write("Carrier:     " + "-"+"\n")
        print("Battleship:  "+"X"*gur(karaliste1)+"-"*(2-
gur(karaliste1)),end="              ")
        f3.write("Battleship:  "+"X"*gur(karaliste1)+"-"*(2-
gur(karaliste1))+"              ")
        print("Battleship:  "+"X"*gur(karaliste2)+"-"*(2-gur(karaliste2)))
        f3.write("Battleship:  "+"X"*gur(karaliste2)+"-"*(2-
gur(karaliste2))+"\n")
        if len(karaliste1["D"])==0:
            print("Destroyer:   "+"X",end="               ")
            f3.write("Destroyer:   "+"X"+"               ")
            if len(karaliste2["D"]) == 0:
                print("Destroyer:   " + "X")
                f3.write("Destroyer:   " + "X"+"\n")
            else:
```

```
                print("Destroyer:   " + "-")
                f3.write("Destroyer:   " + "-"+"\n")

        else:
            print("Destroyer:   "+"-",end="               ")
            f3.write("Destroyer:   "+"-"+"               ")
            if len(karaliste2["D"]) == 0:
                print("Destroyer:   " + "X")
                f3.write("Destroyer:    " + "X"+"\n")
            else:
                print("Destroyer:   " + "-")
                f3.write("Destroyer:   " + "-"+"\n")
        if len(karaliste1["S"])==0:
            f3.write("Submarine:   "+"X"+"               ")
            print("Submarine:   "+"X",end="               ")
            if len(karaliste2["S"]) == 0:
                print("Submarine:   " + "X")
                f3.write("Submarine:   " + "X"+"\n")
            else:
                print("Submarine:   " + "-")
                f3.write("Submarine:   " + "-"+"\n")
        else:
            print("Submarine:   "+"-",end="               ")
            f3.write("Submarine:   "+"-"+"               ")
            if len(karaliste2["S"]) == 0:
                print("Submarine:     " + "X")
                f3.write("Submarine:   " + "X"+"\n")
            else:
                print("Submarine:   " + "-")
                f3.write("Submarine:   " + "-"+"\n")
        print("Patrol Boat: "+"X"*dur(karaliste1)+"-"*(4-
dur(karaliste1)),end="            ")
        f3.write("Patrol Boat: "+"X"*dur(karaliste1)+"-"*(4-
dur(karaliste1))+"            ")
        print("Patrol Boat: "+"X"*dur(karaliste2)+"-"*(4-
dur(karaliste2)),"\n")
        f3.write("Patrol Boat: "+"X"*dur(karaliste2)+"-"*(4-
dur(karaliste2))+"\n"+"\n")
        print(str(c)+","+str(d))
        f3.write(str(c)+","+str(d)+"\n")

    tabloyap3()

except IOError:
    pass

except IndexError:
    print("IndexError: Please use correct input")
    f3.write("IndexError: Please use correct input"+"\n")
    pass
```

```
except ValueError:
    print("ValueError: Please use correct input")
    f3.write("ValueError: Please use correct input"+"\n")
    pass
except AssertionError:
    print("AssertionError: Invalid Operation.")
    f3.write("AssertionError: Invalid Operation."+"\n")
    pass
except:
    print(" kaBOOM: run for your life!")
    f3.write(" kaBOOM: run for your life!"+"\n")
```