

My code is built on a function to understand various variations, this function analyzes situations and outputs required answers.

Take a function(patient system)

For g in lines:

If patient system starts with "create"

call create()

If patient system starts with "list"

call list()

If patient system starts with "remove"

call remove()

If patient system starts with "probability"

call probability()

If patient system starts with "recommendation"

call recommendation()

My code is built on a function to understand various variations, this function analyzes situations and outputs required answers.



```
65 def PatientSystem():
66     for g in lines:
67         if g.startswith("create"):
68             create(g)
69         if g.startswith("remove"):
70             remove(g)
71         if g.startswith("probability"):
72             probability(g)
73         if g.startswith("recommendation"):
74             recommen(g)
75         if g.startswith("list"):
76             tabloyap(g)
77     PatientSystem()
```

My first function "create" ensures that the sentence with create is detected at the beginning of the function of this function and applies its own operations to it. It disconnects the received sentence from the create keyword and detects the already written patient information and saves it to the system.

Take function create(g):

```

a =g[7::]

data.append(a.split(", "))

print feedback

```

```

19 def create(q):
20     a = g[7::]
21     data.append(a.split(", "))
22     s = 'Patient ' + a.split(", ")[0] + ' is recorded.\n'
23     file2.write(s)

```

My second function is "remove", this function deletes the records of the patients whose information has been entered into the system. The way it works is to see the "remove" data entered first, and then removes the sentence and deletes the list containing the information belonging to the remaining name from the system. If a data that is not in the system is encountered, a corresponding return After the function directs the entered input to the "remove" function, it writes the g[7::] expression starting from the 7th character of the g input, and then the remaining name is equal to the 0th index of our "i" value, which scans the lists with the "for" loop, then "i" Our list representing the value " " is being deleted from the system.

Take function remove(g):

For i in data

If g[7::]= i[0]

Dissappend i in data

print feedback

Else :

print feedback

```

10 def remove(g):
11     for i in data:
12         if g[7::] == i[0]:
13             data.remove(i)
14             cd = 'Patient ' + i[0] + ' is removed.' + "\n"
15             file2.write(cd)
16
17     caj = "Patient " + g.split()[1] + " cannot be removed due to absence." + "\n"
18     file2.write(caj)

```

The third function is the "probability" function, the operation style of it is to process the patient information and reveal the cancer risk. The way it works is to briefly enter the system, if the input starts with "probability", it enters the function, then the "probability" statement at the beginning is deleted, the list of the remaining name is similar to the "remove" function. With the "for" loop, the numerical information belonging to the patient is read. After these are processed in the formula related to the subject I found on the internet, the cancer risk is recorded in a new database and the risk is given as output.

Take function probability(g):

For i in data:

if g[12::] = i[0]

calculate $i[3]/(i[3]+1-i[1])$

print feedback

else:

print feedback

```
34 def probability(g):
35     for i in data:
36         if g.split()[1]==i[0]:
37             pay = int(i[3][0:2])/100000
38             payda = pay + 1 - float(i[1])
39             olasilik = (pay/payda*1000//1)/10
40             cgl = "Patient "+g.split()[1]+" has a probability "+str(olasilik)+" of having breast cancer." + "\n"
41             file2.write(cgl)
42             return
43     aksi_durum = "Probability for "+g.split()[1]+" cannot be calculated due to absence." + "\n"
44     file2.write(aksi_durum)
```

If the system extracts an unregistered patient name, it will be recognized by the system and output will be displayed accordingly.

the fourth function indicates whether there is a cure for registered names

the way it works, there is a name matching system similar to the previous "for" loops. If there is a match, it evaluates the numerical information of the person and indicates whether the treatment is available.

For i in data :

If g[15::]=i[0]:

if i[5]>30:

Print have a treatment:

else:

return

print NOT have a treatment

Else:

print NOT have a treatment

```
35 def recommen(g):
36     for i in data:
37         if g.split(" ")[1] == i[0]:
38             pay1 = int(i[3][0:2]) / 100000
39             payda1 = pay1 + 1 - float(i[1])
40             deger = (pay1 / payda1 * 1000 // 1) / 10
41             if deger > float(i[5][0:4]):
42                 cdl = "System suggest "+i[0]+" have a treatment" + "\n"
43                 file2.write(cdl)
44             else:
45                 cbl = "System suggests " + i[0] + " NOT have a treatment" + "\n"
46                 file2.write(cbl)
47             return
48     cfl = "Recommendation for "+g.split(" ")[1]+" cannot be calculated due to absence." + "\n"
49     file2.write(cfl)
```

The last function puts the patients in the table according to the information entered and presents their information in the form of output.

INFO 1	INFO2	INFO3	INFO4	INFO5	INFO6
NAME1	-----	-----	-----	-----	-----

```

NAME2      -----
NAME3      -----
Firstline=' INFO1      INFO2      INFO3      INFO4      INFO5      INFO6 \n'

```

For i in data:

A = i[0]

B =i[1]

C = i[2]

D = i[3]

E =i[4]

F =i[5]

```

50 def tabloyap():
51     firstline = 'Patient Diagnosis      Disease      Disease      Treatment      Treatment\n'
52     secondline = 'Name Accuracy      Name      Incidence      Name      Risk\n'
53     file2.write(firstline)
54     file2.write(secondline)
55     for i in data:
56         Name = i[0]
57         Diagnosis_Accuary= i[1]
58         Disease_Name = i[2]
59         Disease_Incidence = i[3]
60         Treatment_Name = i[4]
61         Treatment_Risk = str(float(i[5][0:4])*100)+"%"
62
63         tablo = f'{Name:<8}{Diagnosis_Accuary:<12}{Disease_Name:<16}{Disease_Incidence:<12}{Treatment_Name:<16}{Treatment_Risk}\n'
64         file2.write(tablo)

```

Tablo = f'A B C D E F\n'

Print(Tablo)

My reader code connect to text files and its read the inputs from text

```

def okuyucu():
    file1 = open('doctors_aid_inputs.txt', 'r')
    lines = file1.readlines()
    file1.close()
    return lines

```

List

```
doctors_aid_outputs - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
Patient Hayriye is recorded.
Patient Deniz is recorded.
Patient Ates is recorded.
Patient Hayriye has a probability 33.3 of having breast cancer.
Patient Toprak is recorded.
Patient Hypatia is recorded.
Patient Pakiz is recorded.
Patient Diagnosis
Name Accuracy Disease Name Disease Incidence Treatment Name Treatment Risk
Hayriye 0.999 Breast Cancer 50/100000 Surgery 40.0%
Deniz 0.9999 Lung Cancer 40/100000 Radiotherapy 50.0%
Ates 0.99 Thyroid Cancer 16/100000 Chemotherapy 2.0%
Toprak 0.98 Prostate Cancer 21/100000 Hormonotherapy 20.0%
Hypatia 0.9975 Stomach Cancer 15/100000 Immunotherapy 4.0%
Pakiz 0.9997 Colon Cancer 14/100000 Targeted Therapy30.0%
Patient Ates cannot be removed due to absence.
Patient Ates has a probability 1.5 of having breast cancer.
Patient Su is recorded.
Patient Diagnosis
Name Accuracy Disease Name Disease Incidence Treatment Name Treatment Risk
Hayriye 0.999 Breast Cancer 50/100000 Surgery 40.0%
Deniz 0.9999 Lung Cancer 40/100000 Radiotherapy 50.0%
Ates 0.99 Thyroid Cancer 16/100000 Chemotherapy 2.0%
Toprak 0.98 Prostate Cancer 21/100000 Hormonotherapy 20.0%
Hypatia 0.9975 Stomach Cancer 15/100000 Immunotherapy 4.0%
Pakiz 0.9997 Colon Cancer 14/100000 Targeted Therapy30.0%
Su 0.98 Breast Cancer 50/100000 Chemotherapy 20.0%
Patient Deniz has a probability 80.0 of having breast cancer.
Patient Pakiz has a probability 31.8 of having breast cancer.
```

BEDİRHAN GENÇASLAN 2210356065