# BBM103 ASSINGMENT 2 REPORTCLINICAL DECISION REPORT SYSTEM

2210356110 ŞÜKRİYE ÖZTÜRK

NOVEMBER 25, 2022
HACETTEPE UNIVERSITY

# CONTENTS

## Analysis:

Medical sciences utilize from statistics to determine and solve health problems. It’s important to classifying the data correctly to use them efficiently. Doctors and scientists get help from an algorithm as “clinical decision support system (CDSS)”. This system should be able to find the information of patient. Our problem, as computer scientists, is that writing the algorithm properly for usage in real life.

CDSS should represent disease of patient when user call the name of patient’s name. Also, this system is able to calculate accuracy of diagnosis and ratio of people which has this disease.

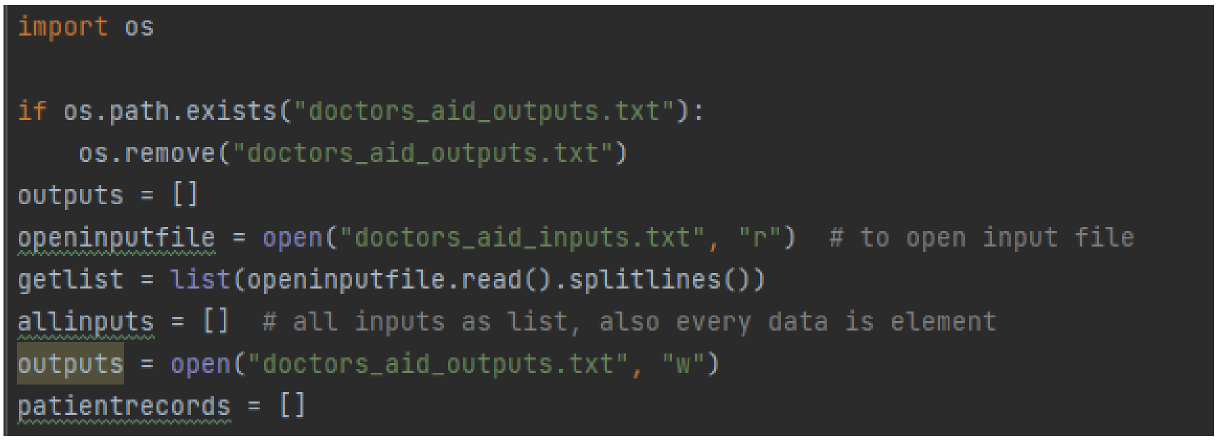
This algorithm has a function that can calculate possibility of having disease thanks to statistics we told below.

CDSS can recommend treatments as well. It is able to recommend the best treatment for the patient and calculate the risk of treatment.

Our problem is writing this program without any error and also making it suitable for working with input files, not with input data which has given by hand.

## Design:

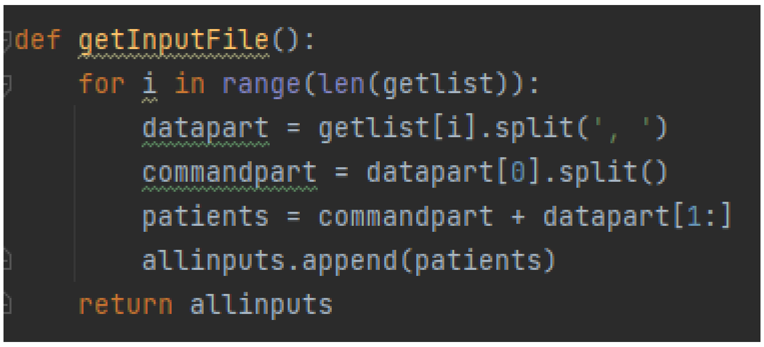
### Solutions and Functions



```python
import os

if os.path.exists("doctors_aid_outputs.txt"):
    os.remove("doctors_aid_outputs.txt")
outputs = []
openinputfile = open("doctors_aid_inputs.txt", "r")  # to open input file
getlist = list(openinputfile.read().splitlines())
allinputs = []  # all inputs as list, also every data is element
outputs = open("doctors_aid_outputs.txt", "w")
patientrecords = []
```

In the first part, I've created outputs list to write all the outputs of algorithm. Also I started to write outputs to my file with giving "open" command in "w" to outputs list. Then I've defined a variable as "openinputfile" to open my input file and get the data from it. I created all list in here, not inside of a function to make it without writing "global x". "allinputs" list will be list in list for all datas in input file. "patientrecords" list will be the data of all patients which is given with "create" command.

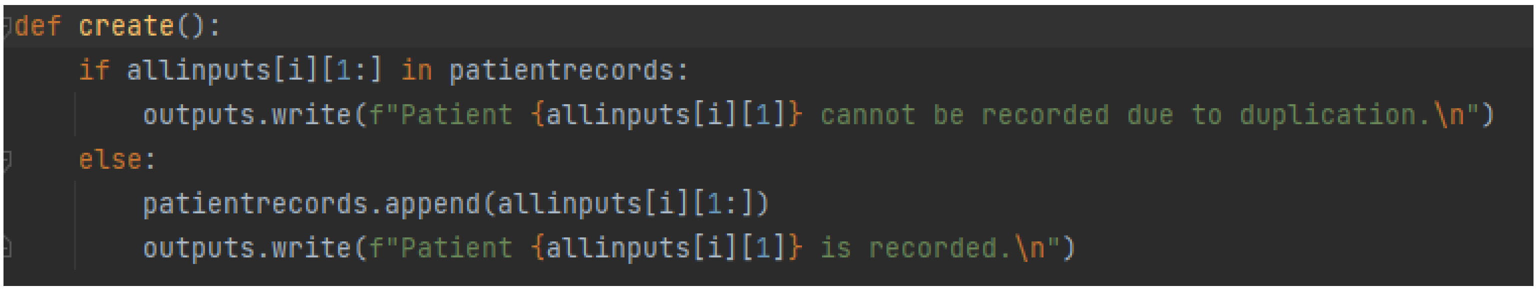
**The first problem: the function "input"**



```python
def getInputFile():
    for i in range(len(getlist)):
        datapart = getlist[i].split(', ')
        commandpart = datapart[0].split()
        patients = commandpart + datapart[1:]
        allinputs.append(patients)
    return allinputs
```

My goal is splitting the command from patient data in "getInputFile" command. I've splitted the command part and the other part as elements and combined them in "allinputs" list. To make this, I used a for loop. Thanks to for loop, function checks every line in input file and splits their first word as element. We will use this when we call our functions to first element of "allinputs".

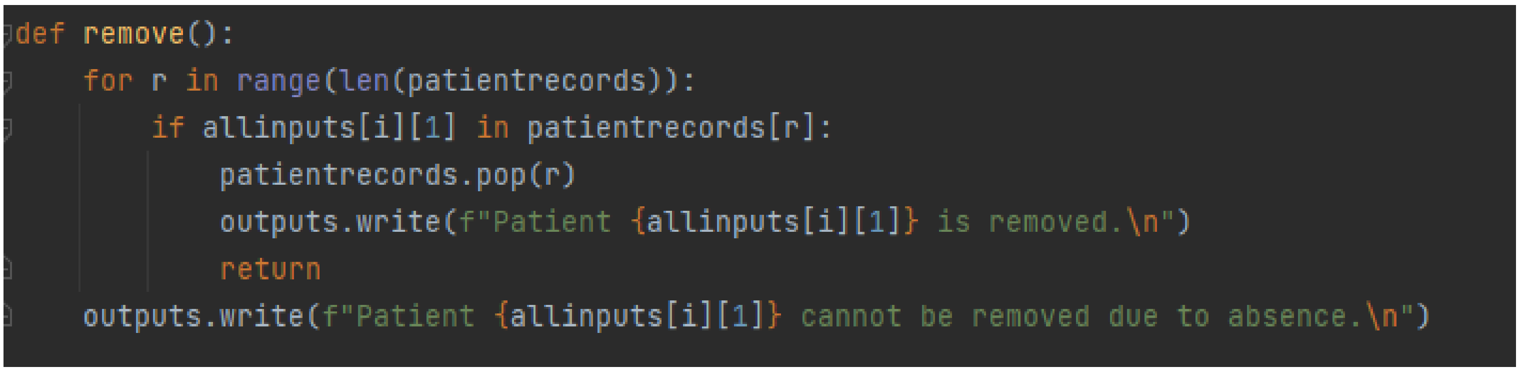
**The second problem: the function "create"**



```python
def create():
    if allinputs[i][1:] in patientrecords:
        outputs.write(f"Patient {allinputs[i][1]} cannot be recorded due to duplication.\n")
    else:
        patientrecords.append(allinputs[i][1:])
        outputs.write(f"Patient {allinputs[i][1]} is recorded.\n")
```

In this function, my goal is taking all data which is given with "create" and making them elements of "patientrecords" list. I used just one if/else condition to check if the patient is in the "patientrecords" list. Function gives success message if patient is not exist in list and record the data. Otherwise, it does nothing and gives a failure message.

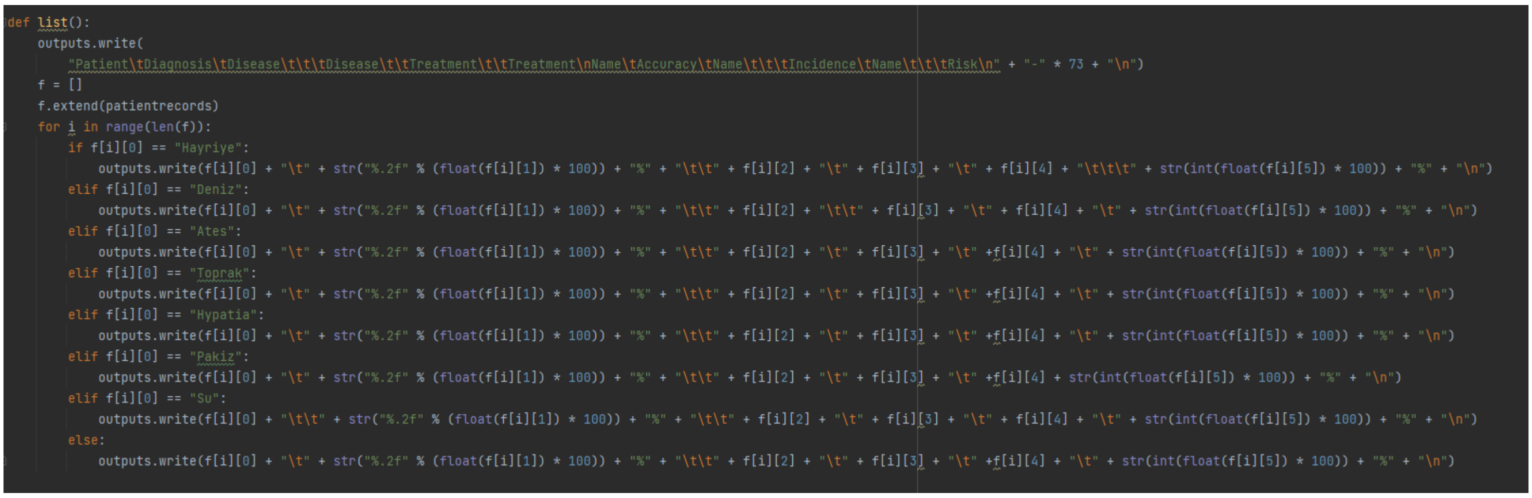
**The third problem: the function "remove"**



```python
def remove():
    for r in range(len(patientrecords)):
        if allinputs[i][1] in patientrecords[r]:
            patientrecords.pop(r)
            outputs.write(f"Patient {allinputs[i][1]} is removed.\n")
            return
    outputs.write(f"Patient {allinputs[i][1]} cannot be removed due to absence.\n")
```

This function works very similar to "create" function. In basic meaning, this command takes the patient name from "allinputs" and checks the situation if it's in the list or not. It pops the patient the the data belongs to it if the name in input is in our "patientrecords". I use one for loop to check all elements of patientrecords and one if/else condition to pop it.

## The fourth problem: the function “list”

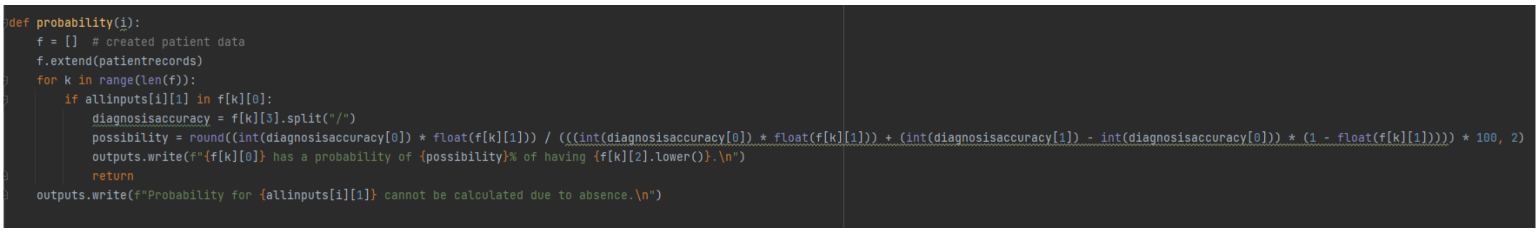


```python
def list():
    outputs.write(
        "Patient\tDiagnosis\tDisease\t\t\tDisease\t\tTreatment\t\tTreatment\nName\tAccuracy\tName\t\t\tIncidence\tName\t\t\tRisk\n" + "-" * 73 + "\n")
    f = []
    f.extend(patientrecords)
    for i in range(len(f)):
        if f[i][0] == "Hayriye":
            outputs.write(f[i][0] + "\t" + str("%.2f" % (float(f[i][1]) * 100)) + "%" + "\t\t" + f[i][2] + "\t" + f[i][3] + "\t" + f[i][4] + "\t\t\t" + str(int(float(f[i][5]) * 100)) + "%" + "\n")
        elif f[i][0] == "Deniz":
            outputs.write(f[i][0] + "\t" + str("%.2f" % (float(f[i][1]) * 100)) + "%" + "\t\t" + f[i][2] + "\t\t" + f[i][3] + "\t" + f[i][4] + "\t" + str(int(float(f[i][5]) * 100)) + "%" + "\n")
        elif f[i][0] == "Ates":
            outputs.write(f[i][0] + "\t" + str("%.2f" % (float(f[i][1]) * 100)) + "%" + "\t\t" + f[i][2] + "\t" + f[i][3] + "\t" +f[i][4] + "\t" + str(int(float(f[i][5]) * 100)) + "%" + "\n")
        elif f[i][0] == "Toprak":
            outputs.write(f[i][0] + "\t" + str("%.2f" % (float(f[i][1]) * 100)) + "%" + "\t\t" + f[i][2] + "\t" + f[i][3] + "\t" +f[i][4] + "\t" + str(int(float(f[i][5]) * 100)) + "%" + "\n")
        elif f[i][0] == "Hypatia":
            outputs.write(f[i][0] + "\t" + str("%.2f" % (float(f[i][1]) * 100)) + "%" + "\t\t" + f[i][2] + "\t" + f[i][3] + "\t" +f[i][4] + "\t" + str(int(float(f[i][5]) * 100)) + "%" + "\n")
        elif f[i][0] == "Pakiz":
            outputs.write(f[i][0] + "\t" + str("%.2f" % (float(f[i][1]) * 100)) + "%" + "\t\t" + f[i][2] + "\t" + f[i][3] + "\t" +f[i][4] + str(int(float(f[i][5]) * 100)) + "%" + "\n")
        elif f[i][0] == "Su":
            outputs.write(f[i][0] + "\t\t" + str("%.2f" % (float(f[i][1]) * 100)) + "%" + "\t\t" + f[i][2] + "\t" + f[i][3] + "\t" + f[i][4] + "\t" + str(int(float(f[i][5]) * 100)) + "%" + "\n")
        else:
            outputs.write(f[i][0] + "\t" + str("%.2f" % (float(f[i][1]) * 100)) + "%" + "\t\t" + f[i][2] + "\t" + f[i][3] + "\t" +f[i][4] + "\t" + str(int(float(f[i][5]) * 100)) + "%" + "\n")
```

this command basically write all data in “patientrecords”. It took that long because of the difference of style of output that we have to do. I counted all tabs and set them proper to names. Also I transformed the type of numbers to show them as percency. I usd one for loop to check every item of patientrecords and one if/else condition to print the right line with the patient.

“f “ list just created for writing “patientrecords” shortly.

## The fifth problem: the function “probability”

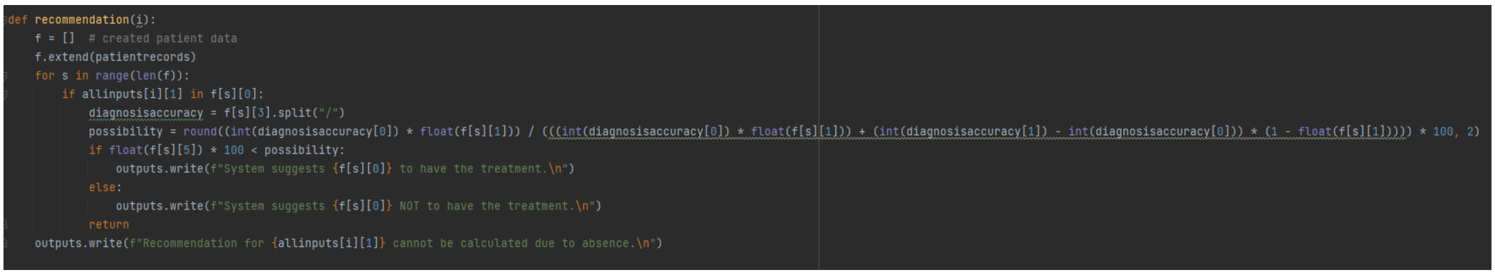


```python
def probability(i):
    f = []  # created patient data
    f.extend(patientrecords)
    for k in range(len(f)):
        if allinputs[i][1] in f[k][0]:
            diagnosisaccuracy = f[k][3].split("/")
            possibility = round((int(diagnosisaccuracy[0]) * float(f[k][1])) / (((int(diagnosisaccuracy[0]) * float(f[k][1])) + (int(diagnosisaccuracy[1]) - int(diagnosisaccuracy[0])) * (1 - float(f[k][1])))) * 100, 2)
            outputs.write(f"{f[k][0]} has a probability of {possibility}% of having {f[k][2].lower()}.\n")
            return
    outputs.write(f"Probability for {allinputs[i][1]} cannot be calculated due to absence.\n")
```

This command calculates the probability of having the disease of patient. First, I used a for loop to return the function inside “f” list. Then I used an if condition to check if the patient name which is given, inside the my “f” list or not. Function calculates the probability and print the result. Otherwise, it gives a failure message.

Possibility variable is the main process to calculate probability. It takes the disease accuracy from “f” and makes the calculations that must have been. I’ve turned all numeric data to integer or float to calculate possibility.

**The sixth problem: the function “recommodation”**

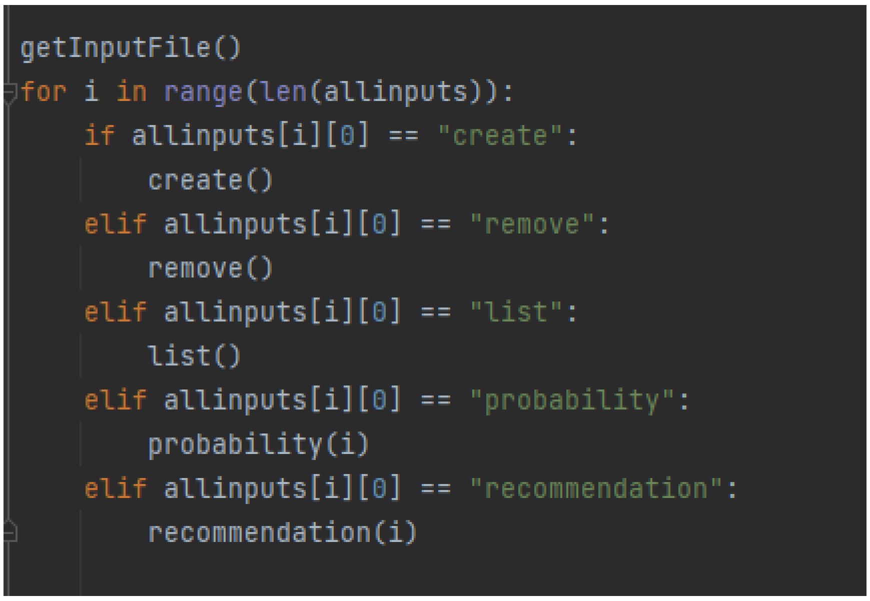


```python
def recommendation(i):
    f = []  # created patient data
    f.extend(patientrecords)
    for s in range(len(f)):
        if allinputs[i][1] in f[s][0]:
            diagnosisaccuracy = f[s][3].split("/")
            possibility = round((int(diagnosisaccuracy[0]) * float(f[s][1])) / (((int(diagnosisaccuracy[0]) * float(f[s][1])) + (int(diagnosisaccuracy[1]) - int(diagnosisaccuracy[0])) * (1 - float(f[s][1])))) * 100, 2)
            if float(f[s][5]) * 100 < possibility:
                outputs.write(f"System suggests {f[s][0]} to have the treatment.\n")
            else:
                outputs.write(f"System suggests {f[s][0]} NOT to have the treatment.\n")
            return
    outputs.write(f"Recommendation for {allinputs[i][1]} cannot be calculated due to absence.\n")
```

This function basically takes the “possibility” of patient and compares with the treatment risk. First I used a for loop to return in all elements of “f”. Then I defined “possibility” again to use in function. I used a if condition to check if the patient name is in the records. If it’s not, function gives a failure message.

I used another if/else condition to compare treatment risk and possibility, it print that treatment is necessary if the possibility of disease greater then treatment risk. Otherwise it prints that treatment is unnecessary.

**The seventh problem: the function “output” and calling the functions**

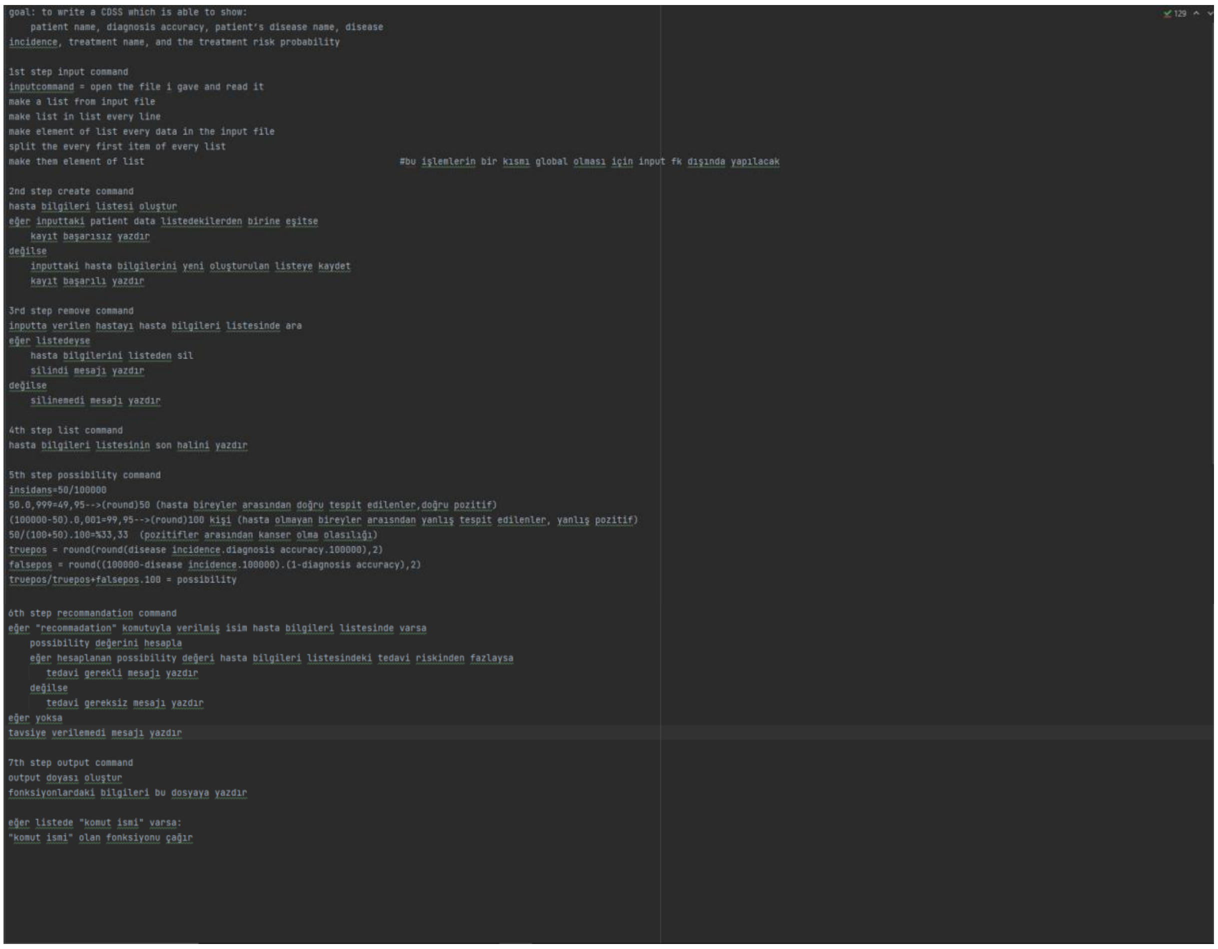


```python
getInputFile()
for i in range(len(allinputs)):
    if allinputs[i][0] == "create":
        create()
    elif allinputs[i][0] == "remove":
        remove()
    elif allinputs[i][0] == "list":
        list()
    elif allinputs[i][0] == "probability":
        probability(i)
    elif allinputs[i][0] == "recommendation":
        recommendation(i)
```

The command up below and “outputs = open(doctors_aid_output.txt, “w”)” should be the output command, but I realised I didn’t add a output function. That’s a mistake and the output function should be like this:

```
def getOutputFile():

        outputs = open("doctors_aid_outputs.txt", "w")

        for i in range(len(allinputs)):

           if allinputs[i][0] == "create":

              create()

           elif allinputs[i][0] == "remove":
                remove()
            elif allinputs[i][0] == "list":
                list()
            elif allinputs[i][0] == "probability":
                probability(i)
            elif allinputs[i][0] == "recommendation":
                recommendation(i)
getInputFile()
getOutputFile()
```

## Pseudocode



```
goal: to write a CDSS which is able to show:
    patient name, diagnosis accuracy, patient's disease name, disease
incidence, treatment name, and the treatment risk probability

1st step input command
inputcommand = open the file i gave and read it
make a list from input file
make list in list every line
make element of list every data in the input file
split the every first item of every list
make them element of list                                   #bu işlemlerin bir kısmı global olması için input fk dışında yapılacak

2nd step create command
hasta bilgileri listesi oluştur
eğer inputtaki patient data listedekilerden birine eşitse
    kayıt başarısız yazdır
değilse
    inputtaki hasta bilgilerini yeni oluşturulan listeye kaydet
    kayıt başarılı yazdır

3rd step remove command
inputta verilen hastayı hasta bilgileri listesinde ara
eğer listedeyse
    hasta bilgilerini listeden sil
    silindi mesajı yazdır
değilse
    silinemedi mesajı yazdır

4th step list command
hasta bilgileri listesinin son halini yazdır

5th step possibility command
insidans=50/100000
50.0,999=49,95-->(round)50 (hasta bireyler arasından doğru tespit edilenler,doğru pozitif)
(100000-50).0,001=99,95-->(round)100 kişi (hasta olmayan bireyler araısndan yanlış tespit edilenler, yanlış pozitif)
50/(100+50).100=%33,33  (pozitifler arasından kanser olma olasılığı)
truepos = round(round(disease incidence.diagnosis accuracy.100000),2)
falsepos = round((100000-disease incidence.100000).(1-diagnosis accuracy),2)
truepos/truepos+falsepos.100 = possibility

6th step recommandation command
eğer "recommadation" komutuyla verilmiş isim hasta bilgileri listesinde varsa
    possibility değerini hesapla
    eğer hesaplanan possibility değeri hasta bilgileri listesindeki tedavi riskinden fazlaysa
       tedavi gerekli mesajı yazdır
    değilse
       tedavi gereksiz mesajı yazdır
eğer yoksa
tavsiye verilemedi mesajı yazdır

7th step output command
output doyası oluştur
fonksiyonlardaki bilgileri bu dosyaya yazdır

eğer listede "komut ismi" varsa:
"komut ismi" olan fonksiyonu çağır
```

I wrote a pseudocode at first to understand what I should write and what is my goal.

## Programmer’s Catalogue:

I tried to understand the main goal and the assignment firstly for probably 1,5-2 days. Writing “create” and “input” commands took my one day for each one of them. First I analysed the assignment and wrote the pseudocode. Then I tried to write algorithm on python file. My first python file was using “print” function instead of using output and input functions. Then I took my code to my main assignment file and created functions.

### Reusability

I think that my code is not reusable %100 percent because I wrote the code suitable with the output that is given to us. Except the command “list”, my algorithm will work properly if the input commands are suitable. I wrote the list command to make the output just the same as the original, so it might make some problems with other programmers.

## User Catalogue:

### Tutorial (User’s Manual)

This program helps you to classify your patient’s data. Features of it:

- Creating new patients
- Removing patients
- Listing patients and data
- Recommending treatment
- Calculating probability of disease

Firstly, you should create a text file name with “doctors_aid_inputs”. Then, you can write commands and data. There are 5 functions you can use.

1. Create

To create new patient, you should write “create” first, then the name of the patient. This command will save the data of patient so, you should data too. You should write the data with this sequence: create + “patient’s name, diagnosis accuracy, disease name, disease incidence, treatment name, treatment risk ”

This command create new patient and add to the list with it’s data.

2. Remove

To remove an existing patient, you should write “remove” command first, then the name of the patient. This command will save the data of patient so, you should data too. You write the data with this sequence: remove + “patient’s name, diagnosis accuracy, disease name, disease incidence, treatment name, treatment risk”

This command remove new patient and add to the list with it’s data.

3. List

To use this command, writing “list” is enough. When you write “list”, all created patients in list until you write list, will be printed.

4. Probability

This command calculates the patient’s probability of disease. To use this, you should write “probability” + patient’s name. If the name is in the list, probability will be calculated, otherwise it will not calculated and a failure message will be printed.

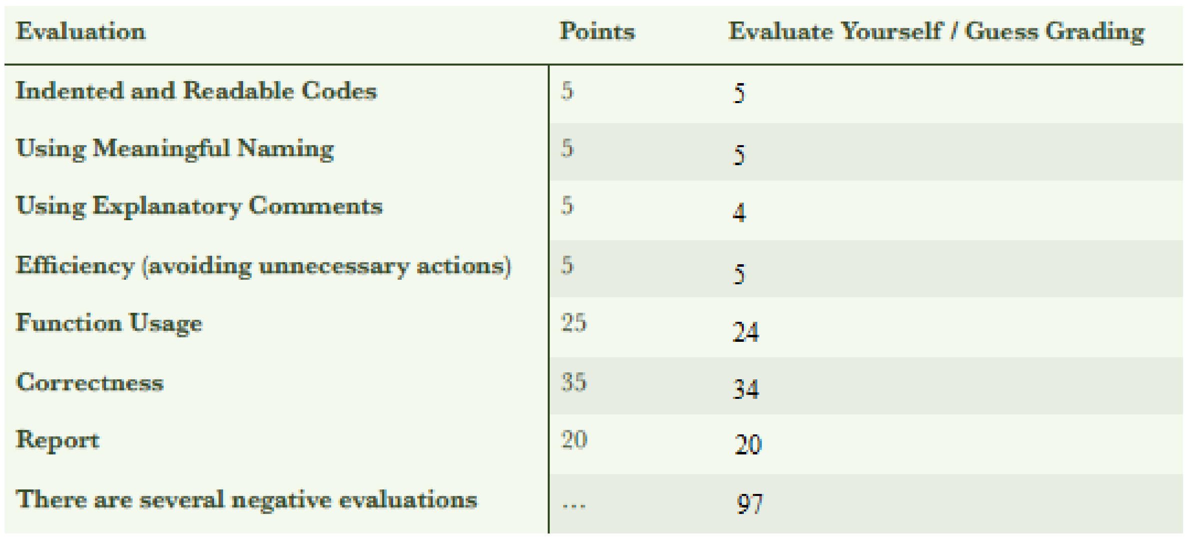
5. Recommendation

This command gives treatment recommendation. To use this command, you should write “recommendation” + patient’s name. If the name is in the list, treatment risk will be checked. If the name is not in the list, program will show a failure message.  If the name is in the list, treatment recommendation will be given to disease probability and treatment risk.

## Restrictions

- This program will not work without an input file which has the commands and data.
- You should write all the command line by line. Otherwise, you cannot reach the true information.

## Grading Table:



| Evaluation | Points | Evaluate Yourself / Guess Grading |
|---|---|---|
| Indented and Readable Codes | 5 | 5 |
| Using Meaningful Naming | 5 | 5 |
| Using Explanatory Comments | 5 | 4 |
| Efficiency (avoiding unnecessary actions) | 5 | 5 |
| Function Usage | 25 | 24 |
| Correctness | 35 | 34 |
| Report | 20 | 20 |
| There are several negative evaluations | ... | 97 |