```
from pyknow import *

diseases_list = []
diseases_symptoms = []
symptom_map = {}
d_desc_map = {}
d_treatment_map = {}

def preprocess():
    global diseases_list,diseases_symptoms,symptom_map,d_desc_map,d_treatment_map
    diseases = open("diseases\diseases.txt","r")
    diseases_t = diseases.read()
    diseases_list = diseases_t.split("\n")
    diseases.close()
    for disease in diseases_list:
        disease_s_file = open("Disease symptoms/" + disease + ".txt")
        disease_s_data = disease_s_file.read()
        s_list = disease_s_data.split("\n")
        diseases_symptoms.append(s_list)
        symptom_map[str(s_list)] = disease
        disease_s_file.close()
        disease_s_file = open("Disease descriptions/" + disease + ".txt")
        disease_s_data = disease_s_file.read()
        d_desc_map[disease] = disease_s_data
        disease_s_file.close()
        disease_s_file = open("Disease treatments/" + disease + ".txt")
        disease_s_data = disease_s_file.read()
        d_treatment_map[disease] = disease_s_data
        disease_s_file.close()
```

```
def identify_disease(*arguments):
        symptom_list = []
        for symptom in arguments:
                symptom_list.append(symptom)
        # Handle key error
        return symptom_map[str(symptom_list)]


def get_details(disease):
        return d_desc_map[disease]


def get_treatments(disease):
        return d_treatment_map[disease]


def if_not_matched(disease):
                print("")
                id_disease = disease
                disease_details = get_details(id_disease)
                treatments = get_treatments(id_disease)
                print("")
                print("The most probable disease that you have is %s\n" %(id_disease))
                print("A short description of the disease is given below :\n")
                print(disease_details+"\n")
                print("The common medications and procedures suggested by other real doctors
are: \n")
                print(treatments+"\n")


# @my_decorator is just a way of saying just_some_function = my_decorator(just_some_function)
#def identify_disease(headache, back_pain, chest_pain, cough, fainting, sore_throat, fatigue,
restlessness,low_body_temp ,fever,sunken_eyes):
class Greetings(KnowledgeEngine):
        @DefFacts()
        def _initial_action(self):
```

```
        print("")
        print("Hi! I am Dr.Yar, I am here to help you make your health better.")
        print("For that you'll have to answer a few questions about your conditions")
        print("Do you feel any of the following symptoms:")
        print("")
        yield Fact(action="find_disease")


@Rule(Fact(action='find_disease'), NOT(Fact(headache=W())),salience = 1)
def symptom_0(self):
        self.declare(Fact(headache=input("headache: ")))

@Rule(Fact(action='find_disease'), NOT(Fact(back_pain=W())),salience = 1)
def symptom_1(self):
        self.declare(Fact(back_pain=input("back pain: ")))

@Rule(Fact(action='find_disease'), NOT(Fact(chest_pain=W())),salience = 1)
def symptom_2(self):
        self.declare(Fact(chest_pain=input("chest pain: ")))

@Rule(Fact(action='find_disease'), NOT(Fact(cough=W())),salience = 1)
def symptom_3(self):
        self.declare(Fact(cough=input("cough: ")))

@Rule(Fact(action='find_disease'), NOT(Fact(fainting=W())),salience = 1)
def symptom_4(self):
        self.declare(Fact(fainting=input("fainting: ")))

@Rule(Fact(action='find_disease'), NOT(Fact(fatigue=W())),salience = 1)
def symptom_5(self):
        self.declare(Fact(fatigue=input("fatigue: ")))
```

```
    @Rule(Fact(action='find_disease'), NOT(Fact(sunken_eyes=W())),salience = 1)
    def symptom_6(self):
        self.declare(Fact(sunken_eyes=input("sunken eyes: ")))

    @Rule(Fact(action='find_disease'), NOT(Fact(low_body_temp=W())),salience = 1)
    def symptom_7(self):
        self.declare(Fact(low_body_temp=input("low body temperature: ")))

    @Rule(Fact(action='find_disease'), NOT(Fact(restlessness=W())),salience = 1)
    def symptom_8(self):
        self.declare(Fact(restlessness=input("restlessness: ")))

    @Rule(Fact(action='find_disease'), NOT(Fact(sore_throat=W())),salience = 1)
    def symptom_9(self):
        self.declare(Fact(sore_throat=input("sore throat: ")))

    @Rule(Fact(action='find_disease'), NOT(Fact(fever=W())),salience = 1)
    def symptom_10(self):
        self.declare(Fact(fever=input("fever: ")))

    @Rule(Fact(action='find_disease'), NOT(Fact(nausea=W())),salience = 1)
    def symptom_11(self):
        self.declare(Fact(nausea=input("Nausea: ")))

    @Rule(Fact(action='find_disease'), NOT(Fact(blurred_vision=W())),salience = 1)
    def symptom_12(self):
        self.declare(Fact(blurred_vision=input("blurred_vision: ")))

    @Rule(Fact(action='find_disease'),Fact(headache="no"),Fact(back_pain="no"),Fact(chest_pa
in="no"),Fact(cough="no"),Fact(fainting="no"),Fact(sore_throat="no"),Fact(fatigue="yes"),Fact(restl
```

```
essness="no"),Fact(low_body_temp="no"),Fact(fever="yes"),Fact(sunken_eyes="no"),Fact(nausea="
yes"),Fact(blurred_vision="no"))

    def disease_0(self):

        self.declare(Fact(disease="Jaundice"))


    @Rule(Fact(action='find_disease'),Fact(headache="no"),Fact(back_pain="no"),Fact(chest_pa
in="no"),Fact(cough="no"),Fact(fainting="no"),Fact(sore_throat="no"),Fact(fatigue="no"),Fact(restle
ssness="yes"),Fact(low_body_temp="no"),Fact(fever="no"),Fact(sunken_eyes="no"),Fact(nausea="n
o"),Fact(blurred_vision="no"))

    def disease_1(self):

        self.declare(Fact(disease="Alzheimers"))


    @Rule(Fact(action='find_disease'),Fact(headache="no"),Fact(back_pain="yes"),Fact(chest_p
ain="no"),Fact(cough="no"),Fact(fainting="no"),Fact(sore_throat="no"),Fact(fatigue="yes"),Fact(rest
lessness="no"),Fact(low_body_temp="no"),Fact(fever="no"),Fact(sunken_eyes="no"),Fact(nausea="
no"),Fact(blurred_vision="no"))

    def disease_2(self):

        self.declare(Fact(disease="Arthritis"))


    @Rule(Fact(action='find_disease'),Fact(headache="no"),Fact(back_pain="no"),Fact(chest_pa
in="yes"),Fact(cough="yes"),Fact(fainting="no"),Fact(sore_throat="no"),Fact(fatigue="no"),Fact(restl
essness="no"),Fact(low_body_temp="no"),Fact(fever="yes"),Fact(sunken_eyes="no"),Fact(nausea="
no"),Fact(blurred_vision="no"))

    def disease_3(self):

        self.declare(Fact(disease="Tuberculosis"))


    @Rule(Fact(action='find_disease'),Fact(headache="no"),Fact(back_pain="no"),Fact(chest_pa
in="yes"),Fact(cough="yes"),Fact(fainting="no"),Fact(sore_throat="no"),Fact(fatigue="no"),Fact(restl
essness="yes"),Fact(low_body_temp="no"),Fact(fever="no"),Fact(sunken_eyes="no"),Fact(nausea="
no"),Fact(blurred_vision="no"))

    def disease_4(self):

        self.declare(Fact(disease="Asthma"))


    @Rule(Fact(action='find_disease'),Fact(headache="yes"),Fact(back_pain="no"),Fact(chest_p
ain="no"),Fact(cough="yes"),Fact(fainting="no"),Fact(sore_throat="yes"),Fact(fatigue="no"),Fact(res
tlessness="no"),Fact(low_body_temp="no"),Fact(fever="yes"),Fact(sunken_eyes="no"),Fact(nausea=
"no"),Fact(blurred_vision="no"))
```

```
    def disease_5(self):
        self.declare(Fact(disease="Sinusitis"))

    @Rule(Fact(action='find_disease'),Fact(headache="no"),Fact(back_pain="no"),Fact(chest_pain="no"),Fact(cough="no"),Fact(fainting="no"),Fact(sore_throat="no"),Fact(fatigue="yes"),Fact(restlessness="no"),Fact(low_body_temp="no"),Fact(fever="no"),Fact(sunken_eyes="no"),Fact(nausea="no"),Fact(blurred_vision="no"))
    def disease_6(self):
        self.declare(Fact(disease="Epilepsy"))

    @Rule(Fact(action='find_disease'),Fact(headache="no"),Fact(back_pain="no"),Fact(chest_pain="yes"),Fact(cough="no"),Fact(fainting="no"),Fact(sore_throat="no"),Fact(fatigue="no"),Fact(restlessness="no"),Fact(low_body_temp="no"),Fact(fever="no"),Fact(sunken_eyes="no"),Fact(nausea="yes"),Fact(blurred_vision="no"))
    def disease_7(self):
        self.declare(Fact(disease="Heart Disease"))

    @Rule(Fact(action='find_disease'),Fact(headache="no"),Fact(back_pain="no"),Fact(chest_pain="no"),Fact(cough="no"),Fact(fainting="no"),Fact(sore_throat="no"),Fact(fatigue="yes"),Fact(restlessness="no"),Fact(low_body_temp="no"),Fact(fever="no"),Fact(sunken_eyes="no"),Fact(nausea="yes"),Fact(blurred_vision="yes"))
    def disease_8(self):
        self.declare(Fact(disease="Diabetes"))

    @Rule(Fact(action='find_disease'),Fact(headache="yes"),Fact(back_pain="no"),Fact(chest_pain="no"),Fact(cough="no"),Fact(fainting="no"),Fact(sore_throat="no"),Fact(fatigue="no"),Fact(restlessness="no"),Fact(low_body_temp="no"),Fact(fever="no"),Fact(sunken_eyes="no"),Fact(nausea="yes"),Fact(blurred_vision="yes"))
    def disease_9(self):
        self.declare(Fact(disease="Glaucoma"))

    @Rule(Fact(action='find_disease'),Fact(headache="no"),Fact(back_pain="no"),Fact(chest_pain="no"),Fact(cough="no"),Fact(fainting="no"),Fact(sore_throat="no"),Fact(fatigue="yes"),Fact(restlessness="no"),Fact(low_body_temp="no"),Fact(fever="no"),Fact(sunken_eyes="no"),Fact(nausea="yes"),Fact(blurred_vision="no"))
    def disease_10(self):
```

```
                self.declare(Fact(disease="Hyperthyroidism"))


        @Rule(Fact(action='find_disease'),Fact(headache="yes"),Fact(back_pain="no"),Fact(chest_p
ain="no"),Fact(cough="no"),Fact(fainting="no"),Fact(sore_throat="no"),Fact(fatigue="no"),Fact(restl
essness="no"),Fact(low_body_temp="no"),Fact(fever="yes"),Fact(sunken_eyes="no"),Fact(nausea="
yes"),Fact(blurred_vision="no"))
        def disease_11(self):
                self.declare(Fact(disease="Heat Stroke"))


        @Rule(Fact(action='find_disease'),Fact(headache="no"),Fact(back_pain="no"),Fact(chest_pa
in="no"),Fact(cough="no"),Fact(fainting="yes"),Fact(sore_throat="no"),Fact(fatigue="no"),Fact(restl
essness="no"),Fact(low_body_temp="yes"),Fact(fever="no"),Fact(sunken_eyes="no"),Fact(nausea="
no"),Fact(blurred_vision="no"))
        def disease_12(self):
                self.declare(Fact(disease="Hypothermia"))


        @Rule(Fact(action='find_disease'),Fact(disease=MATCH.disease),salience = -998)
        def disease(self, disease):
                print("")
                id_disease = disease
                disease_details = get_details(id_disease)
                treatments = get_treatments(id_disease)
                print("")
                print("The most probable disease that you have is %s\n" %(id_disease))
                print("A short description of the disease is given below :\n")
                print(disease_details+"\n")
                print("The common medications and procedures suggested by other real doctors
are: \n")
                print(treatments+"\n")


        @Rule(Fact(action='find_disease'),
                  Fact(headache=MATCH.headache),
                  Fact(back_pain=MATCH.back_pain),
```

```
                Fact(chest_pain=MATCH.chest_pain),
                Fact(cough=MATCH.cough),
                Fact(fainting=MATCH.fainting),
                Fact(sore_throat=MATCH.sore_throat),
                Fact(fatigue=MATCH.fatigue),
                Fact(low_body_temp=MATCH.low_body_temp),
                Fact(restlessness=MATCH.restlessness),
                Fact(fever=MATCH.fever),
                Fact(sunken_eyes=MATCH.sunken_eyes),
                Fact(nausea=MATCH.nausea),

Fact(blurred_vision=MATCH.blurred_vision),NOT(Fact(disease=MATCH.disease)),salience = -999)


        def not_matched(self,headache, back_pain, chest_pain, cough, fainting, sore_throat,
fatigue, restlessness,low_body_temp ,fever ,sunken_eyes ,nausea ,blurred_vision):
                print("\nDid not find any disease that matches your exact symptoms")
                lis = [headache, back_pain, chest_pain, cough, fainting, sore_throat, fatigue,
restlessness,low_body_temp ,fever ,sunken_eyes ,nausea ,blurred_vision]
                max_count = 0
                max_disease = ""
                for key,val in symptom_map.items():
                        count = 0
                        temp_list = eval(key)
                        for j in range(0,len(lis)):
                                if(temp_list[j] == lis[j] and lis[j] == "yes"):
                                        count = count + 1
                        if count > max_count:
                                max_count = count
                                max_disease = val
                if_not_matched(max_disease)
```

```
if __name__ == "__main__":
        preprocess()
        engine = Greetings()
        while(1):
                engine.reset()  # Prepare the engine for the execution.
                engine.run()  # Run it!
                print("Would you like to diagnose some other symptoms?")
                if input() == "no":
                        exit()
                #print(engine.facts)
```

**Output:**

Hi! I am Dr.Yar, I am here to help you make your health better.

For that you'll have to answer a few questions about your conditions

Do you feel any of the following symptoms:

headache: no

chest pain: no

sore throat: yes

cough: no

blurred_vision: no

fatigue: yes

restlessness: yes

sunken eyes: no

Nausea: no

fever: no

back pain: no

low body temperature: no

fainting: no

Did not find any disease that matches your exact symptoms

The most probable disease that you have is Jaundice

A short description of the disease is given below :

Jaundice is a term used to describe a yellowish tinge to the skin and the whites of the eye. Body fluids may also be yellow.

The color of the skin and whites of the eyes will vary depending on levels of bilirubin. Bilirubin is a waste material found in the blood.

Moderate levels lead to a yellow color, while very high levels will appear brown.

An inflamed liver or obstructed bile duct can lead to jaundice, as well as other underlying conditions.

Underlying conditions that may cause jaundice include:

Acute inflammation of the liver: This may impair the ability of the liver to conjugate and secrete bilirubin, resulting in a buildup.

Inflammation of the bile duct: This can prevent the secretion of bile and removal of bilirubin, causing jaundice.

Obstruction of the bile duct: This prevents the liver from disposing of bilirubin.

Hemolytic anemia: The production of bilirubin increases when large quantities of red blood cells are broken down.

Gilbert's syndrome: This is an inherited condition that impairs the ability of enzymes to process the excretion of bile.

Cholestasis: This interrupts the flow of bile from the liver. The bile containing conjugated bilirubin remains in the liver instead of being excreted.

Diagnosis:

The yellowing of skin and eyes are likely to be the main clues a doctor will use before confirming a jaundice diagnosis.

A physical examination will be carried out to look for signs of swelling of the liver and legs, ankles or feet, which might indicate cirrhosis of the liver.

The common medications and procedures suggested by other real doctors are:

Jaundice is treated by managing the underlying cause.

Medication:

Medication or supplements can help jaundice depending on the cause.

Treatment will depend on the underlying cause.

Jaundice treatment targets the cause rather than the jaundice symptoms.

The following treatments are used:

Anemia-induced jaundice may be treated by boosting the amount of iron in the blood by either taking iron supplements or eating more iron-rich foods.

Hepatitis-induced jaundice requires antiviral or steroid medications.

Doctors can treat obstruction-induced jaundice by surgically removing the obstruction.

If the jaundice has been caused by use of a medication, treatment for involves changing to an alternative medication.

Would you like to diagnose some other symptoms?

yes

Hi! I am Dr.Yar, I am here to help you make your health better.

For that you'll have to answer a few questions about your conditions

Do you feel any of the following symptoms:

headache: