Dylan W. Morales
October 2021
Artificial Intelligence

Problem Description:

During the Covid-19 pandemic, medical care has become increasingly difficult to receive. Due to overcrowding in hospitals, even those patients who are not suffering Covid-19 have been relocated, denied medical services due to shortages, and even turned away altogether. Our Expert System hopes to alleviate some of the strain on medical professionals by providing an alternate source for a diagnosis for patients.

The domain of the Problem:

Our Expert system will be able to diagnose and recommend treatment for eight heart diseases. These diseases include: Heart attack, Dilated Cardiomyopathy, Congenital Heart Disease, Stroke, Peripheral Artery Disease, Heart Arrhythmia, Coronary Artery Disease, and Pulmonary Stenosis. It will be able to perform these tasks with accuracy and speed comparable to a trained medical professional.

Methodologies:

The implementation of the knowledge gathered from our research was done by constructing an Expert System. An Expert System like ours seeks to imitate a human expert for the purposes of advising a user, determining a solution to a problem, or solving the problem itself. In our case of course, the Expert System was meant to imitate a heart doctor.

Dylan W. Morales
October 2021
Artificial Intelligence

For the system, our first step was to develop a knowledge base, this will consist of two components. Firstly, our rules. A rule in this context can be simply described as an “if-then” statement. For example, one rule in our system is that if the patient has a murmur, an irregular heartbeat, and has recently fainted, then they are suffering from Pulmonary Stenosis.

Secondly, we will need to establish our facts. This part of our knowledge base will be generated during the running of our program. These facts will be pieces of information either directly asked of the user, or inferred from existing data during runtime. Most of our facts for our program either indicated the presence of a symptom, or a disease. Example: The user's skin could be cold to the touch, this fact could have a status of -1 (not known), 0 (symptom not present), or 1 (symptom present).

The relation of these two parts of the knowledge base will be represented within our inference engine, and it will be what drives the system. The question then is, how are they related? The answer to this question is that our Expert System uses backward and forward chaining to relate our rules to our facts within our knowledge base.

forward and backward chaining are two similar but opposite methods of logic. The main distinguishing factor between the two methods is the starting point.

Most people will be more consciously familiar with forward chaining. This method starts with the facts, and infers the conclusion using a relevant rule. For example, if you have Congenital Heart Disease, then the proper treatment will be a medication: Beta Blockers.

Backward chaining, on the other hand, is the opposite. It starts with a conclusion, or rather it asks whether a *specific* conclusion to a situation is true first, and then gathers the relevant facts. For example, one could ask, do I have Congenital Heart Disease? Then they

would need to gather the relevant facts, in this case one may need to ask: Do I have a blue tinge to my skin?

Of course, the intricacies of the methods don't stop there. Because we are implementing this logic into an Expert System with a knowledge base, there will be math involved. The diagnosis of a disease in our system will be handled by a backward chaining algorithm, while the treatments will be left to a forward chaining algorithm.

Starting with the diagnosis then, we developed a variable list consisting of "facts", in this case all of them were symptoms. These "facts" were actually a structure created in our program containing 3 values: A name, a question to confirm or deny the fact, and the status of the fact. Now that we had our list of variables, we developed a clause variable list and a list of our rules simultaneously. The clause variable list is a list of pointers to different elements in our variable list, it is constructed specifically with our rule base in mind. Every rule can be affected by 4 variables maximum, so every rule will get 4 spaces within the clause variable list to store their fact pointers. Both lists were constructed with the below formula in mind.

CLAUSE NUMBER = 4* (RULE NUMBER / 10 - 1) + 1

For our forward chaining, or the treatment portion of the program, it functioned much the same, but opposite. There was only one determining fact for our forward chaining, and only one rule, so in this case there was very little to calculate. Typically, during a forward chaining operation the program would attempt to find the rule number by using the clause number. The clause number would need to be searched out within the clause variable list. However, since our sample size was so small, there was very little searching or calculations to be done, however, we did still apply the formula.

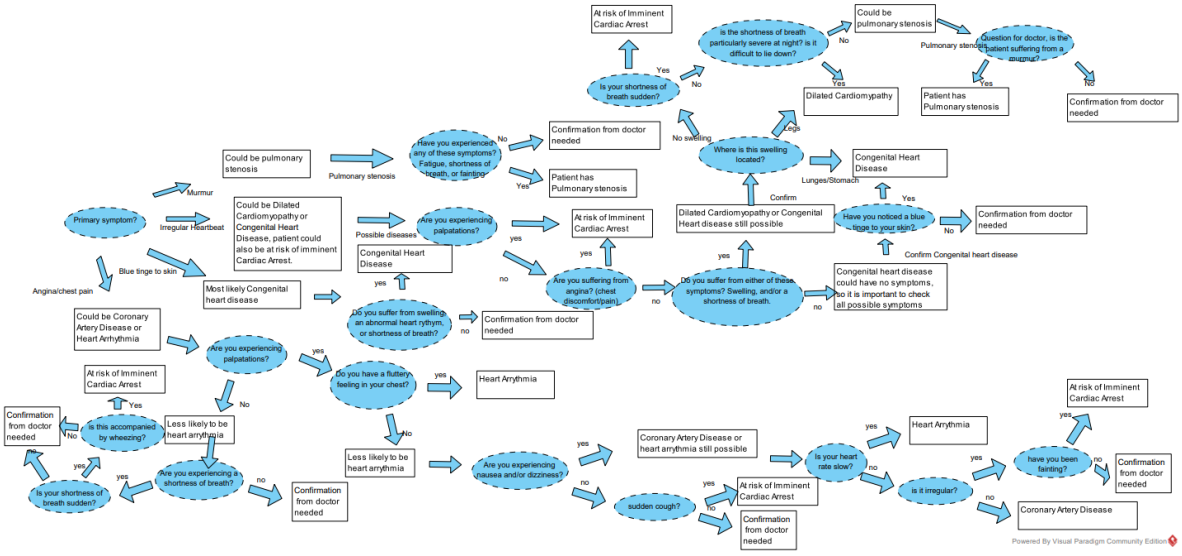
Rule # = [({Quotient (Clause # / 4)} +1) * 10]

Dylan W. Morales
October 2021
Artificial Intelligence

Decision Trees:

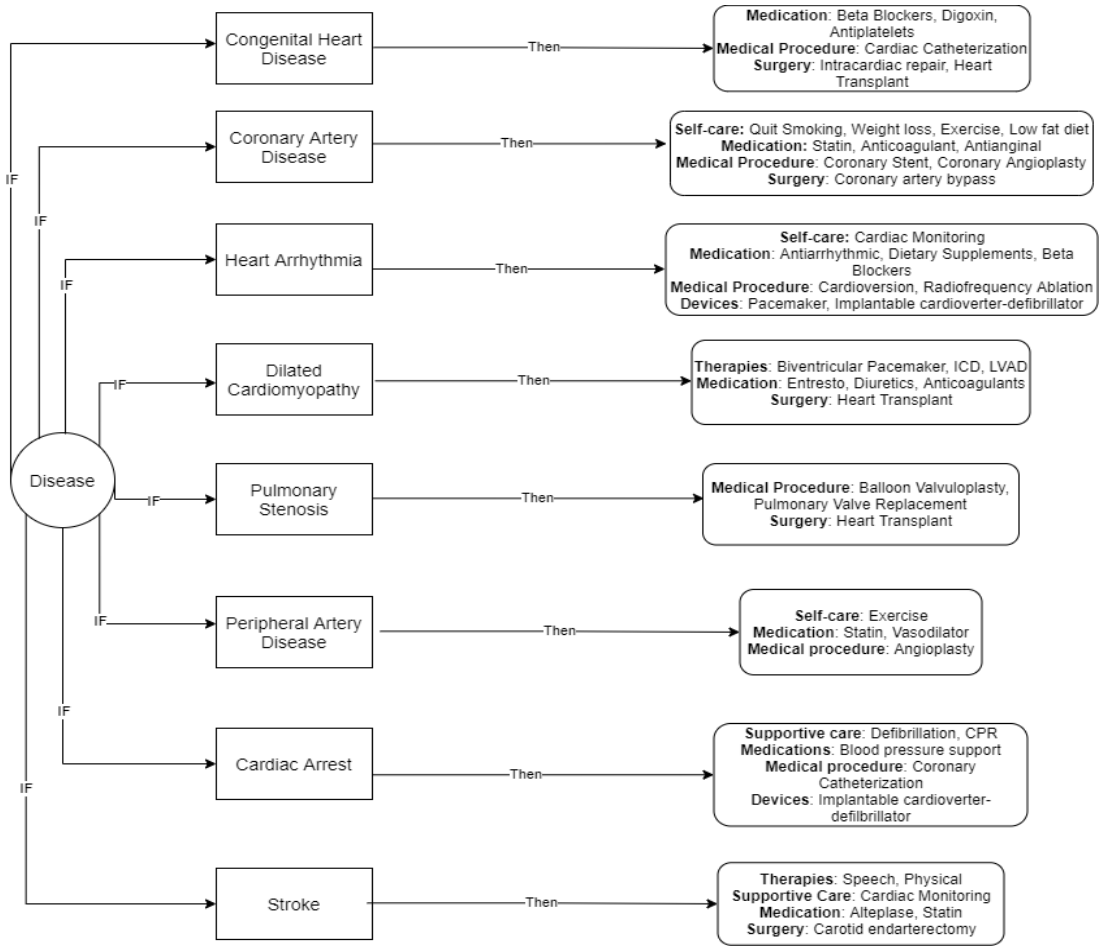
Shown below are our decision trees for our backward and forward chaining. The final programs' decision trees would likely look slightly different, particularly for the diagnosis portion. Ultimately we used these as a guide when determining our rules.

The backward decision tree.



The forward decision tree:

Dylan W. Morales
October 2021
Artificial Intelligence



Dylan W. Morales
October 2021
Artificial Intelligence

Rules:

Below are the rules we formulated originally from the backwards and forwards decision trees above.

1. If patient is suffering from murmur and irregular heartbeat, then they have pulmonary stenosis
2. If patient has murmur and swelling in their lunges or stomach, then they have congenital heart disease.
3. If patient has had a recent heart attack, and has swelling in their legs, then they have dilated cardiomyopathy
4. If the patient is experiencing palpatations and irregular heartbeat, then they are at risk of cardiac arrest.
5. If the patient is experience angina and irregular heartbeat, they are at risk of cardiac arrest
6. If the patient has a blue tinge to their skin and one other symptom of congenital heart disease, they have congenital heart disease
7. If the patient has swelling in their lunges and stomach, accompanied by an irregular heartbeat, then they have congenital heart disease.
8. If the patient has a sudden shortness of breath with palpitations, then they are at risk of cardiac arrest
9. If the patient has been fainting, and has an irregular heartbeat, then they are at risk of cardiac arrest.
10. If the patient is suffering from an irregular heartbeat and dizziness, then they are at risk of cardiac arrest.
11. If the patient is suffering from angina, wheezing, and palpatations then they are at risk of cardiac arest.
12. If the patient is suffering from wheezing and palpatations then they are at risk of cardiac arest.
13. If patient is suffering from an irregular heartbeat and swelling in their legs, then they are suffering from dilated cardiomyopathy.
14. If the patient has an irregular heartbeat and a shortness of breath when lying down or trying to sleep, then they have dilated cardiomyopathy.
15. If the patient is suffering from palpatations, angina, and a fluttery feeling in chest, they are suffering from heart arrythymia
16. If the patient is suffering from slow heart rate, angina, and palpitations, they have heart arrhythmia
17. If they are suffering from irregular heart rate, angina, and palpitations, then they have coronary artery disease.
18. If the patient is experiencing painful cramping at their joints and ulcers, then they are suffering from peripheral artery disease.
19. If the patient is experiencing cramping at their joints, and a drop in skin temperature, then they are suffering from peripheral artery disease.
20. If the patient is experiencing Numbness in their limbs and loss of hair at their limbs, then they are suffering from peripheral artery disease.
21. If the patient is experiencing a drop in body temp at their extremities and painful cramping at their joints, then they are suffering from peripheral artery disease.
22. If the patient is experiencing a drop in body temp at their extremities and painful cramping in the muscles of their thighs and calves, then they are suffering from peripheral artery disease.
23. If the patient is experiencing painful cramping at their joints, thighs, and calves, then they are suffering from peripheral artery disease.
24. If patient has slurred/incoherent speech accompanied by confusion, then they are at risk of stroke.
25. If patient is suffering from slurred/incoherent speech accompanied by dizziness, then they are at risk of stroke.
26. If the patient is suffering from slurred/incoherent speech as well as loss of motor control, then they are at risk of a stroke.
27. If the patient is suffering from dizziness and loss of motor control, then they are at risk of a stroke
28. If the patient is suffering from numbness in their limbs and loss of sight, then they are at risk of stroke.
29. If the patient is suffering from dizziness, lack of balance and confusion, then they are at risk of stroke.
30. If the patient is suffering from lack of balance and numbness in limbs, then they are at risk of stroke.

Dylan W. Morales
October 2021
Artificial Intelligence

If Congenital Heart Disease
Then treatment will include

Medications: Beta Blockers, Digoxin, Antiplatelets
Medical procedure: Cardiac Catheterization
Surgery: Intracardiac repair, Heart Transplant

If Coronary Artery Disease
Then treatment will include

Self-Care: Quit Smoking, Weight loss, Exercise, Low fat diet
Medications: Statin, Anticoagulant, Antianginal
Medical procedure: Coronary Stent, Coronary Angioplasty
Surgery: Coronary artery bypass

If Pulmonary Stenosis
Then treatment will include

Medical Procedure: Balloon Valvuloplasty, Pulmonary Valve Replacement
Surgery: Heart Transplant

If Heart Arrhythmia
Then treatment will include

Self-Care: Cardiac Monitoring
Medications: Antiarrhythmic, Dietary Supplements, Beta Blockers
Medical procedure: Cardioversion, Radio-frequency Ablation
Devices: Pacemaker, Implantable cardioverter-defibrillator

If Dilated Cardiomyopathy
Then treatment will include

Therapies: Bi-ventricular Pacemaker, ICD, LVAD
Medication: Entresto, Diuretics, Anticoagulants
Surgery: Heart Transplant

If Peripheral Artery Disease
Then treatment will include

Self-care: Exercise
Medications: Statin, Vasodilator
Medical procedure: Angioplasty

If Cardiac Arrest
Then treatment will include

Supportive care: Defibrillation, CPR
Medications: Blood pressure support
Medical procedure: Coronary Catheterization
Devices: Implantable cardioverter-defilbrillator

If Stroke
Then treatment will include

Therapies: Speech, Physical
Supportive Care: Cardiac Monitoring
Medication: Alteplase, Statin
Surgery: Carotid endarterectomy

Dylan W. Morales
October 2021
Artificial Intelligence

Program Implementation:

During the implementation of our program, we made heavy use of the following data structures, logical operators, and conditional statements: Stacks, Queues, Switch statements, if-else blocks, pointers, and loops.

Stacks

Because we are using c++ (instead of c in the original program) we had access to the existing library for stacks, we used this stack to store our conclusion (the diagnosis) for our backward chain.

Queues

For queues, the above also applies as we included the queue library, and used it to store our conclusions for the forward chaining (the treatments).

Switch statements

The bulk of our source code for the backward chaining is contained within a large switch statement that allows us to check each disease individually.

If-else

For rules like the one used in the forward chaining section, we were unable to use switch statements, as the fact was stored as a string.

Pointers

For all of our clause variable lists, we used pointers to the original variable list.

Loops

Unlike in the original program, our entire system is contained within a loop that allows the user to run the program again after getting a diagnosis and treatment.

Dylan W. Morales
October 2021
Artificial Intelligence

Source Code:

```
#include <iostream>
#include <string>
#include <stack>
#include <queue>

using namespace std;



//below is the basic infrastructure for a backward and forward chaining
program, I originally coded this with forward chaining in mind so we may have
to make some modifications for backward
//chaining but they shouldn't be major.

/* Defining a structure for facts, this structure will contain a flag value
representing
the facts status in the program (-1 for not known, 0 for no, 1 for yes.)
It will also contain a string value for the related question (if relevant)*/
struct fact
{
    int status = -1;
    string name = "null";
    string question = "null";

};

const int VARLITSIZE = 31;
```

```
fact varlt[VARLITSIZE]; //As was in the example, we have a variable list. This
time of course is a 2d array of the "fact" structure we've created.
const int CLVARLITSIZE = 120;
fact* clvarlt[CLVARLITSIZE]; // This is the corresponding clause variable list,
it will contain pointers to our variable list, as they will be reused
throughout the clause variable list.
char contin = 'Y';
stack<string> backChainConclusions; //This is the backward chaining conclusion
stack, any string representing a disease added to it means the patient has the
disease.
int backClauseNum; //index of clause number for backwards chaining
int result; //int value for result of questions/status of symptom.
string dName; //name of disease to be retrieved once we have a conclusion from
the backward chain
string* forwardClVarLt[4]; //clause variable list for forward chain.
queue<string> forwardConclusions;

void initialize();
int askQuestion(fact& input);

int main()
{
    while(contin == 'y'||contin == 'Y'){
    initialize();

    //This next section will be our rule base of our knowledge base.
    //we have 8 diseases to diagnose, so we will run a loop 8 times, once to
check each disease.
```

```
    bool diagnosed = false; //Flag value for whether the disease has been
found.
    int i = 1; //counter
    while(i <= 8 && diagnosed == false){
        //will check each disease according to what the value of the i counter
is.
        //We have 30 rules total (may end up with more) so they will be divided
among these 8 cases.
        switch(i){
            case 1:{
            cout << "Checking Pulmonary Stenosis." << endl;
                /*According to the formula:
                CLAUSE NUMBER = 4* (RULE NUMBER  / 10 - 1) + 1
                We will be dividing up our clause variable list

                CLAUSE NUMBER = 4* (RULE NUMBER  / 10 - 1)
                this formula will be to find the index in the clvarlt array
(because arrays are base 0)
                */

            //************************
            /*rule 10*/
            //************************
            backClauseNum = 4* (10 / 10 - 1);
                        //rule num^
            //ask question for murmur
            result = askQuestion(*clvarlt[backClauseNum]); //returns whether
the symptom is present or not
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
                //if true, ask question for Irregular heartbeat
                if(result == 1){
                    result = askQuestion(*clvarlt[backClauseNum + 1]);
                    //if true, ask question for fainting
                    if(result == 1){
                        result = askQuestion(*clvarlt[backClauseNum + 2]);
                        //if all true, add pulmonary stenosis to conclusion
stack and set diagnosed to true
                        if(result == 1){
                            backChainConclusions.push("Pulmonary Stenosis");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;
                        }
                    }
                }




            }
            break;
            case 2:{
            cout << "Checking Cardiac Arrest." << endl;
            /*rule 20*/
            backClauseNum = 4* (20 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
//ask question for palpatations
    //if true ask question for irregular heartbeat
    if(result == 1){
        result = askQuestion(*clvarlt[backClauseNum + 1]);
        //if both true then they are at risk for heart attack
        if(result == 1){
                backChainConclusions.push("Cardiac Arrest");
                string test = backChainConclusions.top();
                cout << "You are suffering from " << test << endl;
                diagnosed = true;
                break;

        }
    }

/*rule 30*/
backClauseNum = 4* (30 / 10 - 1);
result = askQuestion(*clvarlt[backClauseNum]);
//ask question for angina
    //if true ask question for irregular heartbeat
    if(result == 1){
        result = askQuestion(*clvarlt[backClauseNum + 1]);
        //if both true then at risk of heart attack
        if(result == 1){
                backChainConclusions.push("Cardiac Arrest");
                string test = backChainConclusions.top();
                cout << "You are suffering from " << test << endl;
                diagnosed = true;
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
                        break;


                }
            }
        /*rule 40*/
        backClauseNum = 4* (40 / 10 - 1);
        result = askQuestion(*clvarlt[backClauseNum]);
        //ask question for Shortness of breath
            //if true ask question for Sudden Shortness of breath
            if(result == 1){
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                //if true ask question for palpatations
                if(result == 1){
                    result = askQuestion(*clvarlt[backClauseNum + 2]);
                    //if both true then they are at risk for heart attack
                    if(result == 1){
                            backChainConclusions.push("Cardiac Arrest");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test <<
endl;
                            diagnosed = true;
                            break;

                    }
                }
            }

                    //if both true then at risk of cardiac arrest
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
            /*rule 50*/
            backClauseNum = 4* (50 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question for Fainting
                //if true ask question for irregular heartbeat
                    //if both true then at risk of cardiac arrest.
                if(result == 1){
                    result = askQuestion(*clvarlt[backClauseNum + 1]);
                        if(result == 1){
                                backChainConclusions.push("Cardiac Arrest");
                                string test = backChainConclusions.top();
                                cout << "You are suffering from " << test <<
endl;
                                diagnosed = true;
                                break;

                        }

                }
            /*rule 60*/
            backClauseNum = 4* (60 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question for Irregular heartbeat
                //if true ask question for dizziness
                    //if both true then at risk of cardiac arrest
                if(result == 1){
                    result = askQuestion(*clvarlt[backClauseNum + 1]);
                        if(result == 1){
```

```
                                backChainConclusions.push("Cardiac Arrest");
                                string test = backChainConclusions.top();
                                cout << "You are suffering from " << test <<
endl;
                                diagnosed = true;
                                break;

                            }

                    }
                /*rule 70*/
                backClauseNum = 4* (70 / 10 - 1);
                result = askQuestion(*clvarlt[backClauseNum]);
                //ask question for angina
                    //if true ask question for wheezing
                        //if true ask question for palpatations
                            //if all true then at risk of cardiac arrest
                if(result == 1){
                        result = askQuestion(*clvarlt[backClauseNum + 1]);
                        if(result == 1){
                            result = askQuestion(*clvarlt[backClauseNum + 2]);
                            if(result == 1){
                                backChainConclusions.push("Cardiac Arrest");
                                string test = backChainConclusions.top();
                                cout << "You are suffering from " << test <<
endl;
                                diagnosed = true;
                                break;
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
            }
        }
    }
/*rule 80*/
backClauseNum = 4* (80 / 10 - 1);
result = askQuestion(*clvarlt[backClauseNum]);
//ask question for wheezing
    //if true ask question for palpations
        //if both true then at risk of cardiac arrest
if(result == 1){
    result = askQuestion(*clvarlt[backClauseNum + 1]);
        if(result == 1){
                backChainConclusions.push("Cardiac Arrest");
                string test = backChainConclusions.top();
                cout << "You are suffering from " << test << endl;
                diagnosed = true;
                break;

        }

}




}
break;
case 3:{
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
            cout << "Checking Congenital Heart disease." << endl;
            /*rule 90*/
            backClauseNum = 4* (90 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question for murmur
                //if true, ask question for swelling
                    //if true, ask question for swelling in lunges or stomach
                        //if all true, add congenital heart disease to
conclusion stack, set diagnosed to true
            if(result == 1){
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                if(result == 1){
                    result = askQuestion(*clvarlt[backClauseNum + 2]);
                    if(result == 1){
                            backChainConclusions.push("Congenital Heart
Disease");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;

                    }
                }
            }
            /*rule 100*/
            backClauseNum = 4* (100 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question for blue tinge to skin
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
                //if true ask question for murmur
                    //if no, ask question for irregular heartbeat
                    //if yes, then add congenital heart disease to conclusion
stack, set diagnosed to true
                        //if no, ask question for shortness of breath
                        //if yes, then add congenital heart disease to
conclusion stack, set diagnosed to true
            if(result == 1){
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                if(result == 1){
                    backChainConclusions.push("Congenital Heart Disease");
                    string test = backChainConclusions.top();
                    cout << "You are suffering from " << test << endl;
                    diagnosed = true;
                    break;
                }
                else{
                        result = askQuestion(*clvarlt[backClauseNum + 2]);
                        if(result == 1){
                            backChainConclusions.push("Congenital Heart
Disease");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;
                        }
                        else{
                            result = askQuestion(*clvarlt[backClauseNum + 3]);
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
                            if(result == 1){
                                backChainConclusions.push("Congenital Heart
Disease");
                                string test = backChainConclusions.top();
                                cout << "You are suffering from " << test <<
endl;
                                diagnosed = true;
                                break;
                            }
                        }

                }




            }

            /*rule 110*/
            backClauseNum = 4* (110 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question for Swelling in Lunges/Stomach
                //if true ask question for irregular heartbeat
                    //if both true then they have Congenital Heart disease
            if(result == 1){
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                    if(result == 1){
                            backChainConclusions.push("Congenital Heart
Disease");
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
                string test = backChainConclusions.top();
                cout << "You are suffering from " << test << endl;
                diagnosed = true;
                break;

        }

}




}
break;
case 4:{
cout << "Checking Dilated Cardiomyopathy." << endl;
/*rule 120*/
backClauseNum = 4* (120 / 10 - 1);
result = askQuestion(*clvarlt[backClauseNum]);
//ask question for recent heart attack
    //if true ask question for swelling
        //if true ask question for swelling in legs
            //if true they have dilated cardiomypathy
if(result == 1){
    result = askQuestion(*clvarlt[backClauseNum + 1]);
    if(result == 1){
        result = askQuestion(*clvarlt[backClauseNum + 2]);
        if(result == 1){
```

```
                            backChainConclusions.push("Dilated
Cardiomyopathy");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;


                        }
                    }
                }
                /*rule 130*/
                backClauseNum = 4* (130 / 10 - 1);
                result = askQuestion(*clvarlt[backClauseNum]);
                //ask question for Irregular heartbeat
                    //if true ask question for Swelling
                        //if true ask question for Swelling in legs
                            //if both true then disease is Dilated Cardiomyopathy
                if(result == 1){
                    result = askQuestion(*clvarlt[backClauseNum + 1]);
                    if(result == 1){
                        result = askQuestion(*clvarlt[backClauseNum + 2]);
                        if(result == 1){
                            backChainConclusions.push("Dilated
Cardiomyopathy");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
                    }
                }
            }
            /*rule 140*/
            backClauseNum = 4* (140 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question for Irregular heartbeat
                //if true ask question for Shortness of breath
                    //if true ask question for Shortness of breath when lying
down or trying to sleep
                        //if both true then they have Dilated Cardiomyopathy
            if(result == 1){
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                if(result == 1){
                    result = askQuestion(*clvarlt[backClauseNum + 2]);
                    if(result == 1){
                            backChainConclusions.push("Dilated
Cardiomyopathy");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;

                    }
                }
            }
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
            }
            break;
            case 5:{
            cout << "Checking Coronary Artery disease." << endl;
            /*rule 150*/
            backClauseNum = 4* (150 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question for Irregular Heartbeat
                //if true ask question for angina
                    //if true ask question for palpitations
                        //if all true then they have Coronary Artery disease
            if(result == 1){
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                if(result == 1){
                    result = askQuestion(*clvarlt[backClauseNum + 2]);
                    if(result == 1){
                            backChainConclusions.push("Coronary Artery
Disease");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;

                    }
                }
            }
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
}
break;
case 6:{
cout << "Checking Heart Arrhythmia." << endl;
/*rule 160*/
backClauseNum = 4* (160 / 10 - 1);
result = askQuestion(*clvarlt[backClauseNum]);
//ask question for palpatations
    //if true ask question for angina
        //if true ask question for Fluttery feeling in chest
            //if all true then they have Heart Arrhythmia
if(result == 1){
    result = askQuestion(*clvarlt[backClauseNum + 1]);
    if(result == 1){
        result = askQuestion(*clvarlt[backClauseNum + 2]);
        if(result == 1){
                backChainConclusions.push("Heart Arrhythmia");
                string test = backChainConclusions.top();
                cout << "You are suffering from " << test << endl;
                diagnosed = true;
                break;

        }
    }
}
/*rule 170*/
```

```
            backClauseNum = 4* (170 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask questions for Slow heartbeat
                //if true ask question for angina
                    //if true ask question for palpitations
                        //if all true, add Heart Arrhythmia to conclusion stack
and set diagnosed to true.
            if(result == 1){
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                if(result == 1){
                    result = askQuestion(*clvarlt[backClauseNum + 2]);
                    if(result == 1){
                            backChainConclusions.push("Heart Arrhythmia");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;

                    }
                }
            }



            }
            break;
            case 7:{
            cout << "Checking Stroke." << endl;
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
/*rule 180*/
backClauseNum = 4* (180 / 10 - 1);
result = askQuestion(*clvarlt[backClauseNum]);
//ask question about slurred/incoherent speech
    //if yes ask if they are confused
        //if both yes, then they are suffering from a stroke
if(result == 1){
    result = askQuestion(*clvarlt[backClauseNum + 1]);
        if(result == 1){
                backChainConclusions.push("Stroke");
                string test = backChainConclusions.top();
                cout << "You are suffering from " << test << endl;
                diagnosed = true;
                break;

        }

}
/*rule 190*/
backClauseNum = 4* (190 / 10 - 1);
result = askQuestion(*clvarlt[backClauseNum]);
//ask question about slurred/incoherent speech
    //if yes, ask about dizziness
        //if both yes, then they are suffering from stroke
if(result == 1){
    result = askQuestion(*clvarlt[backClauseNum + 1]);
        if(result == 1){
                backChainConclusions.push("Stroke");
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
                string test = backChainConclusions.top();
                cout << "You are suffering from " << test << endl;
                diagnosed = true;
                break;

        }

}
/*rule 200*/
backClauseNum = 4* (200 / 10 - 1);
result = askQuestion(*clvarlt[backClauseNum]);
//ask question about slurred/incoherent speech
    //if yes, ask about loss of motor control
        //if both yes, then they are suffering from stroke
if(result == 1){
    result = askQuestion(*clvarlt[backClauseNum + 1]);
        if(result == 1){
                backChainConclusions.push("Stroke");
                string test = backChainConclusions.top();
                cout << "You are suffering from " << test << endl;
                diagnosed = true;
                break;

        }

}
/*rule 210*/
backClauseNum = 4* (210 / 10 - 1);
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
result = askQuestion(*clvarlt[backClauseNum]);
//ask question about dizziness
    //if yes, ask about loss of motor control
        //if both yes, then they are suffering from stroke

if(result == 1){
    result = askQuestion(*clvarlt[backClauseNum + 1]);
        if(result == 1){
                backChainConclusions.push("Stroke");
                string test = backChainConclusions.top();
                cout << "You are suffering from " << test << endl;
                diagnosed = true;
                break;

        }

}



/*rule 220*/
backClauseNum = 4* (210 / 10 - 1);
result = askQuestion(*clvarlt[backClauseNum]);
//ask question about numbness in limbs
    //if yes, ask about loss of sight
        //if both yes, then they are suffering from stroke
if(result == 1){
    result = askQuestion(*clvarlt[backClauseNum + 1]);
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
        if(result == 1){
                backChainConclusions.push("Stroke");
                string test = backChainConclusions.top();
                cout << "You are suffering from " << test << endl;
                diagnosed = true;
                break;

        }

}



/*rule 230*/
backClauseNum = 4* (230 / 10 - 1);
result = askQuestion(*clvarlt[backClauseNum]);
//ask about dizziness
    //if yes, ask about lack of balance
        //if yes, ask about confusion
            //if all yes, then they are suffering from stroke
if(result == 1){
    result = askQuestion(*clvarlt[backClauseNum + 1]);
    if(result == 1){
        result = askQuestion(*clvarlt[backClauseNum + 2]);
        if(result == 1){
                backChainConclusions.push("Stroke");
                string test = backChainConclusions.top();
                cout << "You are suffering from " << test << endl;
                diagnosed = true;
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
                break;


        }
    }
}
/*rule 240*/
backClauseNum = 4* (240 / 10 - 1);
result = askQuestion(*clvarlt[backClauseNum]);
//ask about lack of balance
    //if yes, ask about numbness in limbs
        //if both yes, then suffering from stroke.
if(result == 1){
    result = askQuestion(*clvarlt[backClauseNum + 1]);
        if(result == 1){
                backChainConclusions.push("Stroke");
                string test = backChainConclusions.top();
                cout << "You are suffering from " << test << endl;
                diagnosed = true;
                break;


        }


}



}


break;
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
            case 8:{
            cout << "Checking Peripheral Artery Disease." << endl;
            /*rule 250*/
            backClauseNum = 4* (250 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question for Cramping at joints
                //if true ask question for ulcers
                    //if both true then they have Peripheral artery disease
            if(result == 1){
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                    if(result == 1){
                            backChainConclusions.push("Peripheral Artery
Disease");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;

                    }

            }
            /*rule 260*/
            backClauseNum = 4* (260 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question for Cramping at joints
                //if true ask question for drop in body temp
                    //if both true then they have Peripheral artery disease.
            if(result == 1){
```

```
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                    if(result == 1){
                            backChainConclusions.push("Peripheral Artery
Disease");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;

                    }

            }
            /*rule 270*/
            backClauseNum = 4* (270 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question regarding numbness in limbs
                //if yes, ask question for loss of hair at extremities
                    //if both yes, then they have peripheral artery disease
            if(result == 1){
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                    if(result == 1){
                            backChainConclusions.push("Peripheral Artery
Disease");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;
```

```
                }


            }
            /*rule 280*/
            backClauseNum = 4* (280 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question regarding drop in body temp
                //if yes, ask for ulcers
                    //if both yes, then they have peripheral artery disease
            if(result == 1){
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                    if(result == 1){
                            backChainConclusions.push("Peripheral Artery
Disease");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;


                    }


            }
            /*rule 290*/
            backClauseNum = 4* (290 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question regarding drop in body temp
                //if yes, ask for cramping in thighs and calves
                    //if both yes, then they have peripheral artery disease
```

```
            if(result == 1){
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                    if(result == 1){
                            backChainConclusions.push("Peripheral Artery
Disease");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;

                    }

            }
            /*rule 300*/
            backClauseNum = 4* (300 / 10 - 1);
            result = askQuestion(*clvarlt[backClauseNum]);
            //ask question regarding cramping at joints
                //if yes, ask for cramping in thighs and calves
                    //if both yes, then they have peripheral artery disease
            if(result == 1){
                result = askQuestion(*clvarlt[backClauseNum + 1]);
                    if(result == 1){
                            backChainConclusions.push("Peripheral Artery
Disease");
                            string test = backChainConclusions.top();
                            cout << "You are suffering from " << test << endl;
                            diagnosed = true;
                            break;
```

```
                }

            }



            }
            break;

        }



        i++;
    }



    if(!backChainConclusions.empty()){
            dName = backChainConclusions.top();
            /*8 Disease
            Congenital Heart Disease
            Coronary Artery Disease
            Heart Arrhythmia
            Dilated Cardiomyopathy
            Pulmonary Stenosis
            Peripheral Artery Disease
            Cardiac Arrest
            Stroke
            */
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
          /*theres only one rule for the rule base and clVarLt of the forward
chaining
          however we're still going to calculate the rule number as if theres
more



          this formula
          Rule # = [({Quotient  (Clause # / 4)} +1) * 10]



          modified to this
          Rule # = [({Quotient  (Clause # / 4)} +1) * 10]

          is what we'll use



          theres only one rule, so in this case we know the clause number = 0
          however in any other case we would need to use a linear search to
find the clause num

          */
          int ruleNum = ((0 / 4) +1) * 10;
          //^^ Will always be ten
```

```
          switch(ruleNum){
            case 10:{
              if(*forwardClVarLt[0] == "Congenital Heart Disease")
              {
                  cout << "For " << dName << " Treatment will include. \n" <<
endl;

                  forwardConclusions.push("Medications: Beta Blockers, Digoxin,
Antiplatelets \nMedical procedure: Cardiac Catheterization \nSurgery:
Intracardiac repair, Heart Transplant");

                  cout << forwardConclusions.front() << endl;
              }
              else if (*forwardClVarLt[0] == "Coronary Artery Disease")
              {
                  cout << "For " << dName << " Treatment will include. \n" <<
endl;

                  forwardConclusions.push("Self-Care: Quit Smoking, Weight
loss, Exercise, Low fat diet \nMedications: Statin, Anticoagulant, Antianginal
\nMedical procedure: Coronary Stent, Coronary Angioplasty \nSurgery: Coronary
artery bypass");

                  cout << forwardConclusions.front() << endl;
              }
              else if (*forwardClVarLt[0] == "Heart Arrhythmia")
              {
```

```
                cout << "For " << dName << " Treatment will include. \n" <<
endl;

                forwardConclusions.push("Self-Care: Cardiac Monitoring
\nMedications: Antiarrhythmic, Dietary Supplements, Beta Blockers \nMedical
procedure: Cardioversion, Radio-frequency Ablation \nDevices: Pacemaker,
Implantable cardioverter-defibrillator");

                cout << forwardConclusions.front() << endl;
            }
            else if (*forwardClVarLt[0] == "Dilated Cardiomyopathy")
            {
                cout << "For " << dName << " Treatment will include. \n" <<
endl;

                forwardConclusions.push("Therapies: Bi-ventricular Pacemaker,
ICD, LVAD \nMedication: Entresto, Diuretics, Anticoagulants \nSurgery: Heart
Transplant");

                cout << forwardConclusions.front() << endl;
            }
            else if (*forwardClVarLt[0] == "Pulmonary Stenosis")
            {
                cout << "For " << dName << " Treatment will include. \n" <<
endl;

                forwardConclusions.push("Medical Procedure: Balloon
Valvuloplasty, Pulmonary Valve Replacement \nSurgery: Heart Transplant");
```

```
                cout << forwardConclusions.front() << endl;
            }
            else if (*forwardClVarLt[0] == "Peripheral Artery Disease")
            {
                cout << "For " << dName << " Treatment will include. \n" <<
endl;

                forwardConclusions.push("Self-care: Exercise \nMedications:
Statin, Vasodilator \nMedical procedure: Angioplasty");

                cout << forwardConclusions.front() << endl;
            }
            else if (*forwardClVarLt[0] == "Cardiac Arrest")
            {
                cout << "For " << dName << " Treatment will include. \n" <<
endl;

                forwardConclusions.push("Supportive care: Defibrillation, CPR
\nMedications: Blood pressure support \nMedical procedure: Coronary
Catheterization \nDevices: Implantable cardioverter-defilbrillator");

                cout << forwardConclusions.front() << endl;

            }
            else if (*forwardClVarLt[0] == "Stroke")
            {
```

```
                    cout << "For " << dName << " Treatment will include. \n" <<
endl;

                    forwardConclusions.push("Therapies: Speech, Physical
\nSupportive Care: Cardiac Monitoring \nMedication: Alteplase, Statin
\nSurgery: Carotid endarterectomy");

                    cout << forwardConclusions.front() << endl;
                }
            }
          }
        cout << endl;
    }
    else{
        cout << endl << "No diagnosis found, have a good day." << endl;
    }




    cout << "Would you like to run the program again? Y or y for yes, N or n
for no" << endl;
    cin >> contin;
    while(contin != 'y'&&contin != 'Y'&&contin != 'n'&&contin != 'N'){
        cout << "Incorrect entry. Y or y for yes, N or n for no" << endl;
        cin >> contin;
```

```
    }
    if(contin == 'n'||contin == 'N'){
        cout << "Thank you for using this application, goodbye." << endl;
        return 0;
    }

    }

    return 0;

}


//intializes our rule list, fact list, and clause fact list with the proper
data
void initialize(){

    //************************************************************
    //Building our symptoms
    //************************************************************




    //Murmur
    varlt[0].name = "Murmur";
    varlt[0].question = "Are you suffering from a murmur?";
    varlt[0].status = -1;
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
//Irregular Heartbeat
varlt[1].name = "Irregular Heartbeat";
varlt[1].question = "Are you suffering from an Irregular Heartbeat?";
varlt[1].status = -1;

//Slow heartbeat
varlt[2].name = "Slow heartbeat";
varlt[2].question = "Are you suffering from a Slow heartbeat?";
varlt[2].status = -1;

//Blue tinge on skin
varlt[3].name = "Blue tinge on skin";
varlt[3].question = "Have you noticed a blue tinge to your skin?";
varlt[3].status = -1;

//Angina
varlt[4].name = "Angina";
varlt[4].question = "Have you been suffering from a pain in your chest?";
varlt[4].status = -1;

//Palpatations
varlt[5].name = "Palpatations";
varlt[5].question = "Have you been suffering from palpatations?";
varlt[5].status = -1;

//Fainting
varlt[6].name = "Fainting";
varlt[6].question = "Have you fainted recently?";
```

```
    varlt[6].status = -1;

    //Wheezing
    varlt[7].name = "Wheezing";
    varlt[7].question = "Have you been Wheezing recently?";
    varlt[7].status = -1;

    //Shortness of breath (unspecified)
    varlt[8].name = "Shortness of breath (unspecified)";
    varlt[8].question = "Have you experienced Shortness of breath?";
    varlt[8].status = -1;

    //Sudden shortness of breath
    varlt[9].name = "Sudden shortness of breath";
    varlt[9].question = "Have you experienced a Sudden shortness of breath
recently?";
    varlt[9].status = -1;

    //Shortness of breath when trying to sleep
    varlt[10].name = "Shortness of breath when trying to sleep";
    varlt[10].question = "Do you suffer from Shortness of breath when trying to
sleep?";
    varlt[10].status = -1;

    //Fluttery Feeling in chest
    varlt[11].name = "Fluttery Feeling in chest";
    varlt[11].question = "Do you experience Fluttery Feeling in the chest?";
    varlt[11].status = -1;
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
//Swelling (unspecified)
varlt[12].name = "Swelling (unspecified)";
varlt[12].question = "Have you noticed Swelling?";
varlt[12].status = -1;

//Swelling in legs
varlt[13].name = "Swelling in legs";
varlt[13].question = "Have you noticed Swelling in the legs?";
varlt[13].status = -1;

//Swelling in Lunges/Stomach
varlt[14].name = "Swelling in Lunges/Stomach";
varlt[14].question = "Have you noticed Swelling in the Lunges/Stomach?";
varlt[14].status = -1;

//Cough
varlt[15].name = "Cough";
varlt[15].question = "Have you been Coughing recently?";
varlt[15].status = -1;

//Wheezing (severe/sudden cough)
varlt[16].name = "Wheezing (severe/sudden cough)";
varlt[16].question = "Do you experience Severe/Sudden Wheezing?";
varlt[16].status = -1;

//Nausea
varlt[17].name = "Nausea";
```

```
    varlt[17].question = "Do you feel Nausea?";
    varlt[17].status = -1;

    //Dizziness
    varlt[18].name = "Dizziness";
    varlt[18].question = "Are you suffering from Dizziness?";
    varlt[18].status = -1;

    //Recent heart attack (cardiac arrest)
    varlt[19].name = "Recent heart attack (cardiac arrest)";
    varlt[19].question = "Have you suffered from a Recent Heart Attack?";
    varlt[19].status = -1;

    //Numbness in limbs
    varlt[20].name = "Numbness in limbs";
    varlt[20].question = "Do your limbs feel numb?";
    varlt[20].status = -1;

    //Loss of hair at extremities
    varlt[21].name = "Loss of hair at extremities";
    varlt[21].question = "Have you lost hair at your arms, legs, fingers, toes
etc.?";
    varlt[21].status = -1;

    //Drop in body temp at extremities
    varlt[22].name = "Drop in body temp at extremities";
    varlt[22].question = "Have you experienced a coldness in your arms, legs,
fingers, toes etc.?";
```

```
    varlt[22].status = -1;

    //Cramping at joints
    varlt[23].name = "Cramping at joints";
    varlt[23].question = "Have you experienced cramping at your joints? (Hips,
shoulders, elbows, knees)";
    varlt[23].status = -1;

    //Cramping in thighs or calves
    varlt[24].name = "Cramping in thighs or calves";
    varlt[24].question = "Have you experienced cramping in thighs or calves?";
    varlt[24].status = -1;

    //Slurred or incoherent speech
    varlt[25].name = "Slurred or incoherent speech";
    varlt[25].question = "Is your speech incoherent or slurred?";
    varlt[25].status = -1;

    //Lack of balance
    varlt[26].name = "Lack of balance";
    varlt[26].question = "Have you experienced a lack of balance";
    varlt[26].status = -1;

    //Loss of motor control
    varlt[27].name = "Loss of motor control";
    varlt[27].question = "Have you experienced a loss of motor control?";
    varlt[27].status = -1;
```

```
    //Confusion
    varlt[28].name = "Confusion";
    varlt[28].question = "Have you suffered from a sudden confusion?";
    varlt[28].status = -1;

    //Loss of sight
    varlt[29].name = "Loss of sight";
    varlt[29].question = "Have you suffered from a loss of sight (partial or
full)?";
    varlt[29].status = -1;

    //Ulcers
    varlt[30].name = "Ulcers";
    varlt[30].question = "Have you experienced Ulcers recently?";
    varlt[30].status = -1;




    //*************************************************************
    //Dividing up our clause variable list for the backward chain
    //*************************************************************

    /*According to the formula:
    CLAUSE NUMBER = 4* (RULE NUMBER  / 10 - 1) + 1
    We will be dividing up our clause variable list

    CLAUSE NUMBER = 4* (RULE NUMBER  / 10 - 1)
```

```
    this formula will be to find the index in the clvarlt array (because arrays
are base 0)
    */

    /*rule 10*/
    clvarlt[0] = &varlt[0]; // points to murmur
    clvarlt[1] = &varlt[1]; // points to irregular heartbeat
    clvarlt[2] = &varlt[6]; // points to fainting
    //empty

    /*rule 20*/
    clvarlt[4] = &varlt[5]; // points to palpatations
    clvarlt[5] = &varlt[1]; // points to irregular heartbeat
    //empty
    //empty

    /*rule 30*/
    clvarlt[8] = &varlt[4]; // points to angina
    clvarlt[9] = &varlt[1]; // points to irregular heartbeat
    //empty
    //empty

    /*rule 40*/
    clvarlt[12] = &varlt[8]; // points to shortness of breath
    clvarlt[13] = &varlt[9]; // points to sudden shortness of breath
    clvarlt[14] = &varlt[5]; // points to palpatations
    //empty
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
/*rule 50*/
clvarlt[16] = &varlt[6]; // points to fainting
clvarlt[17] = &varlt[1]; // points to irregular heartbeat
//empty
//empty

/*rule 60*/
clvarlt[20] = &varlt[1]; // points to irregular heartbeat
clvarlt[21] = &varlt[18]; // points to dizziness
//empty
//empty

/*rule 70*/
clvarlt[24] = &varlt[4]; // points to angina
clvarlt[25] = &varlt[7]; // points to wheezing
clvarlt[26] = &varlt[5]; // points to palpatations
//empty

/*rule 80*/
clvarlt[28] = &varlt[7]; // points to wheezing
clvarlt[29] = &varlt[5]; // points to palpatations
//empty
//empty

/*rule 90*/
clvarlt[32] = &varlt[0]; // points to murmur
clvarlt[33] = &varlt[12]; // points to swelling
clvarlt[34] = &varlt[14]; // points to swelling in lunges or stomach
```

Dylan W. Morales
October 2021
Artificial Intelligence

```
//empty

/*rule 100*/
clvarlt[36] = &varlt[3]; // points to blue tinge
clvarlt[37] = &varlt[0]; // points to murmur
clvarlt[38] = &varlt[1]; // points to irregular heartbeat
clvarlt[39] = &varlt[8]; // points to shortness of breath

/*rule 110*/
clvarlt[40] = &varlt[14]; // points to swelling in lunges or stomach
clvarlt[41] = &varlt[1]; // points to irregular heartbeat
//empty
//empty

/*rule 120*/
clvarlt[44] = &varlt[19]; // points to recent heart attack
clvarlt[45] = &varlt[12]; // points to swelling
clvarlt[46] = &varlt[13]; // points to swelling in legs
//empty

/*rule 130*/
clvarlt[47] = &varlt[1]; // points to irregular heartbeat
clvarlt[48] = &varlt[12]; // points to swelling
clvarlt[49] = &varlt[13]; // points to swelling in legs
//empty

/*rule 140*/
clvarlt[52] = &varlt[1]; // points to irregular heartbeat
```

```
clvarlt[53] = &varlt[8]; // points to shortness of breath

clvarlt[54] = &varlt[10]; // points to shortness of breath when lying down

//empty


/*rule 150*/

clvarlt[56] = &varlt[1]; // points to irregular heartbeat

clvarlt[57] = &varlt[4]; // points to angina

clvarlt[58] = &varlt[5]; // points to palpatations

//empty


/*rule 160*/

clvarlt[60] = &varlt[5]; // points to palpatations

clvarlt[61] = &varlt[4]; // points to angina

clvarlt[62] = &varlt[11]; // points to fluttery feeling in chest

//empty


/*rule 170*/

clvarlt[64] = &varlt[2]; // points to slow heartbeat

clvarlt[65] = &varlt[4]; // points to angina

clvarlt[66] = &varlt[5]; // points to palpatations

//empty


/*rule 180*/

clvarlt[68] = &varlt[25]; // points to slurring

clvarlt[69] = &varlt[28]; // points to confusion

//empty

//empty
```

```
/*rule 190*/
clvarlt[72] = &varlt[25]; // points to slurring
clvarlt[73] = &varlt[18]; // points to dizziness
//empty
//empty

/*rule 200*/
clvarlt[76] = &varlt[25]; // points to slurring
clvarlt[77] = &varlt[27]; // points to loss of motor control
//empty
//empty

/*rule 210*/
clvarlt[80] = &varlt[18]; // points to dizziness
clvarlt[81] = &varlt[27]; // points to loss of motor control
//empty
//empty

/*rule 220*/
clvarlt[84] == &varlt[20]; // points to numbness in limbs
clvarlt[85] == &varlt[29]; // points to loss of sight
//empty
//empty

/*rule 230*/
clvarlt[88] = &varlt[18]; // points to dizziness
clvarlt[89] = &varlt[26]; // points to lack of balance
clvarlt[90] = &varlt[28]; // points to confusion
```

```
//empty

/*rule 240*/
clvarlt[92] = &varlt[26]; // points to lack of balance
clvarlt[93] = &varlt[20]; // points to numbness in limbs
//empty
//empty

/*rule 250*/
clvarlt[96] = &varlt[23]; // points to cramping at joints
clvarlt[97] = &varlt[30]; // points to ulcers
//empty
//empty

/*rule 260*/
clvarlt[100] = &varlt[23]; // points to cramping at joints
clvarlt[101] = &varlt[22]; // points to drop in body temp at extremities
//empty
//empty

/*rule 270*/
clvarlt[104] = &varlt[20]; // points to numbness in limbs
clvarlt[105] = &varlt[21]; // points to loss of hair at extremities
//empty
//empty

/*rule 280*/
clvarlt[108] = &varlt[22]; // points to drop in body temp at extremities
```

```
clvarlt[109] = &varlt[30]; // points to ulcers

//empty

//empty


/*rule 290*/

clvarlt[112] = &varlt[22]; // points to drop in body temp at extremities

clvarlt[113] = &varlt[24]; // points to cramping in thighs and calves

//empty

//empty


/*rule 300*/

clvarlt[116] = &varlt[23]; // points to cramping at joints

clvarlt[117] = &varlt[24]; // points to cramping in thighs and calves

//empty

//empty









/*Forward Chaining clause variable list*/


forwardClVarLt[0] = &dName;
```

```
    return;
}


//retrieves the question from an inputted fact structure (passed by reference),
outputs it, and sets the facts status according to the answer. Returns the
facts status as well.
int askQuestion(fact &input){

    if(input.status != -1){
        cout << "Symptom: " << input.name << " status confirmed." << endl;
        if(input.status == 1){
            cout << "Symptom present." << endl;
        }
        if(input.status == 0){
            cout << "Symptom absent." << endl;
        }
        return input.status;
    }

    char answer;
    cout << input.question << endl;
    cout << "Y or y for yes, N or n for no." << endl;
    cin >> answer;
    cout << endl;
    while(answer != 'y'&&answer != 'Y'&&answer != 'n'&&answer != 'N'){
```

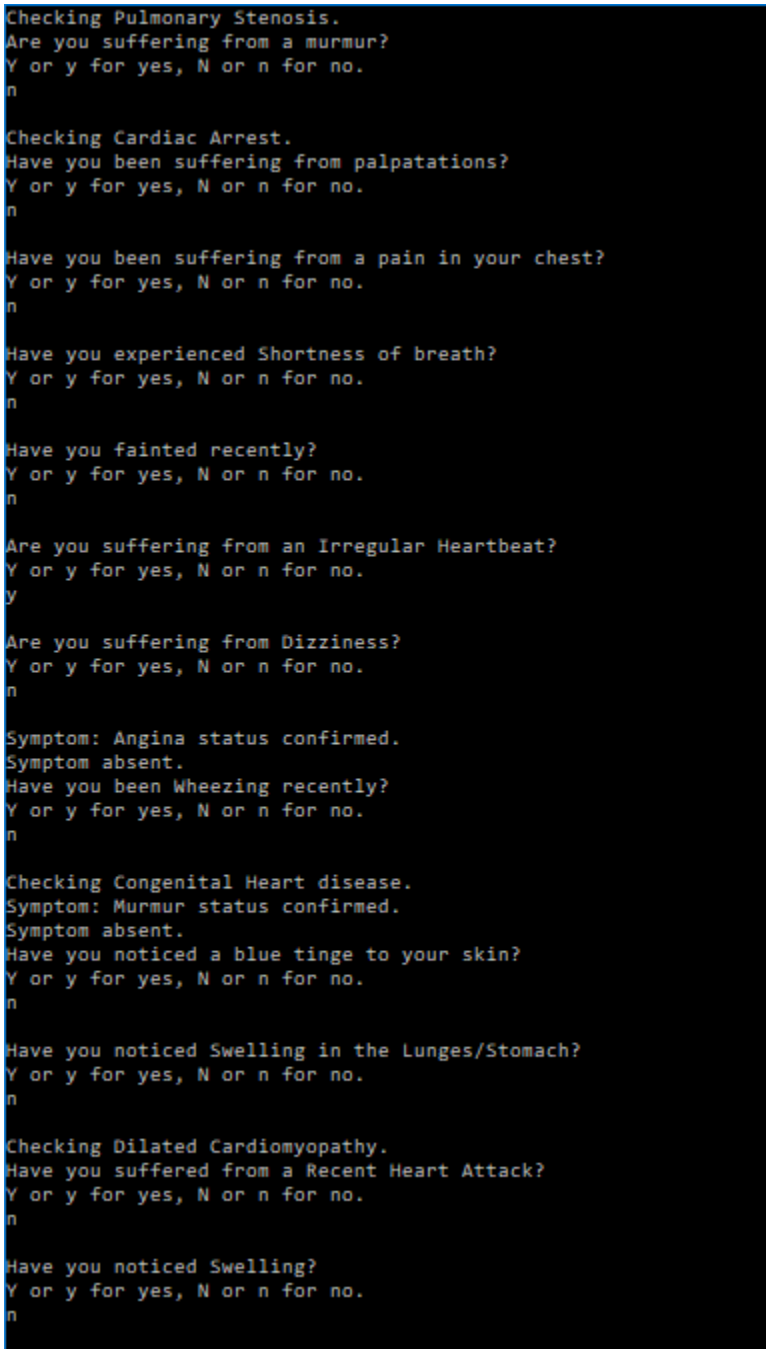
```
        cout << "Incorrect entry. Y or y for yes, N or n for no" << endl;
        cin >> answer;
    }
    if(answer == 'n'||answer == 'N'){
        input.status = 0;
    }
    else if(answer == 'y'||answer == 'Y'){
        input.status = 1;
    }
    return input.status;

}
```

Program Run:

Below is a test run with a test patient with Peripheral Artery Disease

They are suffering from the following symptoms

As well as an Irregular heartbeat



Dylan W. Morales
October 2021
Artificial Intelligence

```
Have you noticed Swelling in the Lunges/Stomach?
Y or y for yes, N or n for no.
n

Checking Dilated Cardiomyopathy.
Have you suffered from a Recent Heart Attack?
Y or y for yes, N or n for no.
n

Have you noticed Swelling?
Y or y for yes, N or n for no.
n

Symptom: Irregular Heartbeat status confirmed.
Symptom present.
Symptom: Shortness of breath (unspecified) status confirmed.
Symptom absent.
Checking Coronary Artery disease.
Symptom: Irregular Heartbeat status confirmed.
Symptom present.
Symptom: Angina status confirmed.
Symptom absent.
Checking Heart Arrhythmia.
Symptom: Palpatations status confirmed.
Symptom absent.
Are you suffering from a Slow heartbeat?
Y or y for yes, N or n for no.
n

Checking Stroke.
Is your speech incoherent or slurred?
Y or y for yes, N or n for no.
n

Symptom: Slurred or incoherent speech status confirmed.
Symptom absent.
Symptom: Slurred or incoherent speech status confirmed.
Symptom absent.
Symptom: Dizziness status confirmed.
Symptom absent.
Symptom: Dizziness status confirmed.
Symptom absent.
Symptom: Dizziness status confirmed.
Symptom absent.
Have you experienced a lack of balance
Y or y for yes, N or n for no.
n

Checking Peripheral Artery Disease.
Have you experienced cramping at your joints? (Hips, shoulders, elbows, knees)
Y or y for yes, N or n for no.
y

Have you experienced Ulcers recently?
Y or y for yes, N or n for no.
y

You are suffering from Peripheral Artery Disease
For Peripheral Artery Disease Treatment will include.

Self-care: Exercise
Medications: Statin, Vasodilator
Medical procedure: Angioplasty

Would you like to run the program again? Y or y for yes, N or n for no
```

Dylan W. Morales
October 2021
Artificial Intelligence

Analysis of the Program:

As can be seen above, during the running of the program, every case and if-else block is compiled and run, which is why you may see multiple “status confirmed” sentences. Due to the nature of backward chaining, where the program must check each conclusion first, we ran the program this way. It may have been useful for the runtime of the program if we used forward chaining more thoroughly. In this sense, backward chaining can be compared closely to something like a linear search algorithm, where the efficiency and runtime of the program is dependent upon how far into the rule list the correct conclusion is. Forward chaining however, uses the rules at its disposal more effectively to calculate which conclusion is correct.

Analysis of the Results:

During the running of our program, it was able to correctly diagnose individual diseases, as long as there were not any more than 2 non-applicable symptoms present to the disease that the patient had.

Additionally, the program is unable to diagnose multiple diseases simultaneously, when confronted with multiple diseases it may end up completely misdiagnosing the diseases. The reason for this is usually because the two diseases intersect in their symptoms to a degree. A more dynamic inference engine and a larger rule base could help to solve this issue.

Dylan W. Morales
October 2021
Artificial Intelligence

Conclusion:

During the development of this project we learned about the use of Expert Systems, their effectiveness in comparison to a real human expert, and the structure of the expert system itself. The knowledge base used in our Expert System used backwards and forwards chaining in its inference engine. The use of backward chaining can lead to slower performance, while forward chaining could potentially be more dynamic, and able to handle multiple conclusions more effectively, (further development needed.)

With further development upon our inference engine and rulebase, I believe we should be able to make an Expert System capable of perfectly and accurately diagnosing any heart disease.

References:

"Better Information. Better Health.," *WebMD*. [Online]. Available: https://www.webmd.com/. [Accessed: 03-Oct-2021].

NHS choices. [Online]. Available: https://www.nhs.uk/. [Accessed: 03-Oct-2021].

Mayo Clinic. [Online]. Available: https://www.mayoclinic.org/. [Accessed: 03-Oct-2021].

"A computer science portal for geeks," *GeeksforGeeks*. [Online]. Available: https://www.geeksforgeeks.org/. [Accessed: 03-Oct-2021].

Dylan W. Morales
October 2021
Artificial Intelligence

Contributions:

Dylan Morales- Backward chaining loop, switch statement, and rules, structure and formulation and writing of source code.
Clause variable list structure and source code
Forward chaining source code.

Micheal Mendez- Backward chaining loop, switch statement, and rules, contributed source code.
Clause variable list source code
Forward chaining structure