CS5346 ADVANCED ARTIFICIAL INTELLIGENCE

PROJECT – 1 SPRING 2024

INTELLIGENT EXPERT SYSTEM

FOR

DIAGNOSING CARDIOVASCULAR (HEART) DISEASES

AND

TREATMENT RECOMMENDATION

Submitted by

Sayali Pathak (A05295714)

Team members: **Gopika Mahadevan and Sri Sudha Kambhampati**

Under the guidance of

Dr. Moonis Ali

CONTENTS

1. PROBLEM DESCRIPTION

The goal of this project is to develop intelligent computer system for the hospitals to diagnose cardiovascular (Heart) diseases and to recommend treatment based on the diagnosis. Cardiovascular diseases include Valvular Heart Disease, Acute Coronary Syndrome, Arrhythmia etc. Diagnosing these diseases accurately is very important to provide appropriate treatment. Heart problems can be really complicated with lots of different symptoms and treatments. So, this intelligent expert system will help doctors making diagnosing and recommending treatments easier using knowledge base and appropriate techniques.

To diagnose the disease, expert system uses backward chaining algorithm and to recommend appropriate treatment system uses forward chaining algorithm.

This expert system is reliable and efficient because it uses decision trees, rules, and efficient algorithms. It's also very easy to use. The system asks questions in simple English and prompts the user for input. Based on the user's answers, it asks relevant follow-up questions and recommends appropriate treatments.

2. DOMAIN

2.1 Importance of diagnosing cardiovascular (Heart) diseases

The domain of this project is cardiovascular (heart) diseases, which encompasses conditions such as Aortic aneurysm, Myocardial infarction, and Pulmonary hypertension, among others. These diseases present with a variety of symptoms including chest pain, fatigue, arrhythmia, and irregular heartbeat. Accurate diagnosis and proper treatment of cardiovascular diseases are critical for healthcare professionals and essential for patients' well-being. This intelligent expert system is designed to focus on diagnosing heart diseases and recommending appropriate treatments to users.

Cardiovascular (Heart) diseases covered in this project are –

1. Ventricular tachycardia
2. Takotsubo cardiomyopathy (broken heart syndrome)
3. Long qt syndrome
4. Paroxysmal atrial fibrillation
5. Pulmonary hypertension
6. Aortic aneurysm
7. Valvular heart disease
8. Acute coronary syndrome
9. Coronary artery disease
10. Rheumatic heart disease
11. Venous thromboembolism
12. Pulmonary embolism
13. Endocarditis
14. Ischemic cardiomyopathy

15. Arrhythmia
16. Heat valve stenosis
17. Brugada syndrome
18. Cardiac tamponade
19. Peripheral artery disease
20. Coronary microvascular disease
21. Chronic total occlusion
22. Atherosclerosis
23. Hypertrophic cardiomyopathy
24. Mitral valve prolapse
25. Hypertrophic obstructive cardio myopathy
26. Eisenmenger syndrome
27. Cardiogenic shock
28. Myocardial infarction
29. Wolff-Parkinson-White syndrome
30. Aortic stenosis

2.2 Intelligent expert system

An intelligent expert system is a software tool that leverages knowledge to solve problems similarly to humans, but with greater speed and accuracy. This system can greatly enhance decision- making, problem-solving, and efficiency in various tasks and domains.

The components of intelligent expert systems are –

2.2.1 Knowledge Base

The knowledge base is a repository for facts, rules, domain- specific information used by the expert system. It has mainly two components

1. Rule Base – During the development of an expert system, the system developer creates the rules that the system will follow. These rules are stored in the rule base, which contains all the necessary instructions for the system to make decisions.

2. Fact Base – During the development of an expert system, the system developer sets up the structure of the fact base. However, the actual information or facts that the system needs are added during the program's execution.

2.2.2 Inference Engine

The Inference Engine is a vital part of intelligent expert systems. It's like the brain of the system, responsible for reasoning and drawing conclusions based on the information given to it. It works similarly to how humans solve problems but does so faster, more efficiently, and accurately.

2.2.3 Interface

Interfaces are what users use to interact with the expert system. Through the interface, users can input information or data, and based on that input, the system can display the corresponding results on the screen. It's like the gateway between the user and the expert system, allowing for communication and exchange of information.

3. METHODOLOGIES

Backward Chaining methodology is used for diagnosing the disease and forward methodology is used for recommending the treatment.

3.1 Backward Chaining

Backward chaining is a goal driven technique where we begin with a goal or conclusion that needs to be achieved. The reasoning process starts from the goal and works backward through a chain of rules to determine the required facts or conditions for the goal to be valid.

The data structures used in the backwards chaining are –

1. **Conclusion List** –
   Variables included in the ‘THEN’ part of rules and corresponding rule numbers are stored in this conclusion list. The backward chaining process involves searching for the goal variable in a list and then retrieving the corresponding rule.

2. **Variable List** –
   The variables from the 'IF' part, which are not present in the 'THEN' part of the rules, along with their corresponding values, are stored in this list. Initially, no variable is initialized, and it is denoted as 'NI' (Not Initialized). Once the variables are initialized, the appropriate values will be updated in the list.

3. **Clause Variable List** –
   All the variables from the ‘IF’ part are stored in the list, and each variable is assigned a clause number in the list. The size of the list is determined by the maximum number of variables present in the ‘IF’ condition of all the rules.
   The clause number for the given rule can be calculated using following formula.
   If the rule numbers are sequenced like 10,20,30,40, … then the formula is –
   CLAUSE NUMBER = 15*((RULE NUMBER/10)-1)+1
   Here for each rule 15 spaces are allocated in the clause variable list.

   If a rule has 7 variables in ‘IF’ part, then the first 7 spaces in the clause variable list are occupied by these variables, leaving the remaining 8 spaces blank.

4. **Derived Global Variable List** –

In the backward chaining, derived global variable list is used to store variable name and its corresponding value. It is used for tracking the recursive calls.

3.2 Forward Chaining

Forward chaining is a data-driven technique because it utilizes the available data to drive the reasoning process. It is commonly used in recommendation systems where the system uses user input data or existing knowledge to generate recommendations based on predefined rules.

Data structures used in forward chaining are –

1. **Clause Variable List**

   All the variables from the 'IF' part are stored in the list, and each variable is assigned a clause number in the list. The size of the list is determined by the maximum number of variables present in the 'IF' condition of all the rules.

   The following formula can be used to find rule number from the given clause number –

   If the rule numbers are sequenced like 10,20,30,40, …then formula is:
   RULE NUMBER = (QUOTIENT(CLAUSE NUMBER/10)+1)*10
   Here for each rule 10 spaces are allocated in the clause variable list.

   If a rule has 6 variables in 'IF' part, then the first 6 spaces in the clause variable list are occupied by these variables, leaving the remaining 4 spaces blank.

2. **Variable List**

   The variables from the 'IF' part, which are not present in the 'THEN' part of the rules, along with their corresponding values, are stored in this list. Initially, no variable is initialized, and it is denoted as 'NI' (Not Initialized). Once the variables are initialized, the appropriate values will be updated in the list.

3. **Global Conclusion Variable Queue**

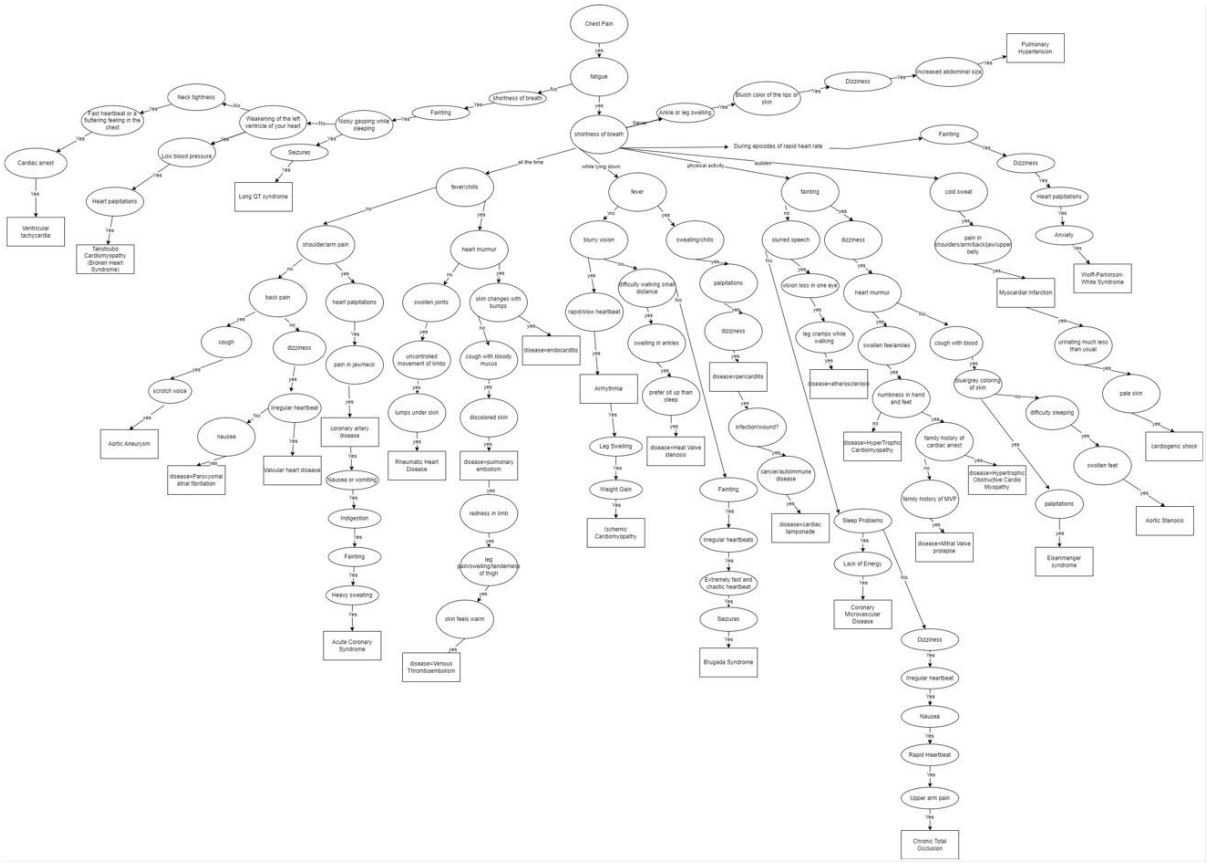
   This data structure is used to store conclusions or derived facts using the execution process. If needed this queue structure helps in triggering further inferences.

4. **Derived Conclusion List**

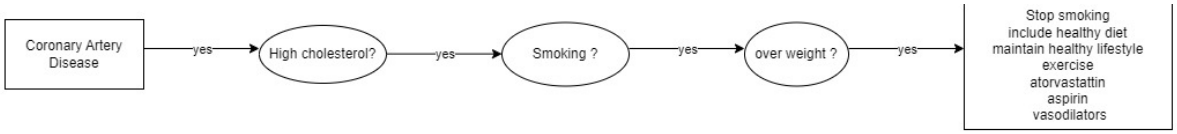
   The derived conclusion list stores the conclusion that have been derived during the reasoning process.

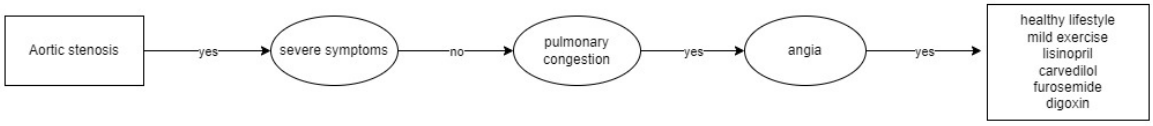
4. DECISION TREE

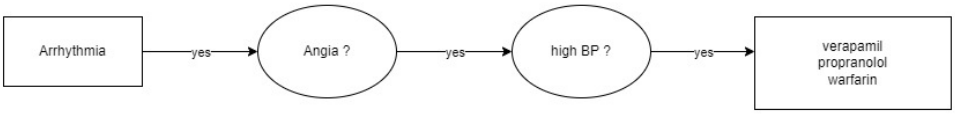
4.1 Backward Chaining

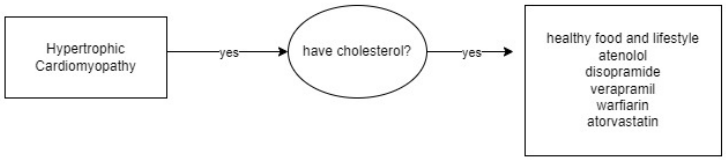


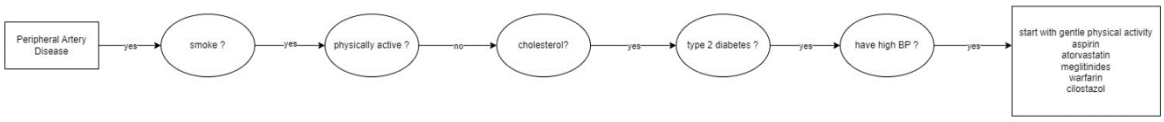
4.2. Forward Chaining

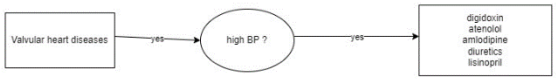


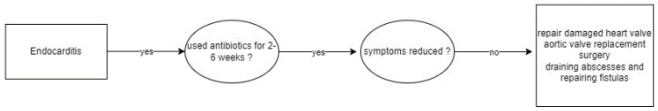


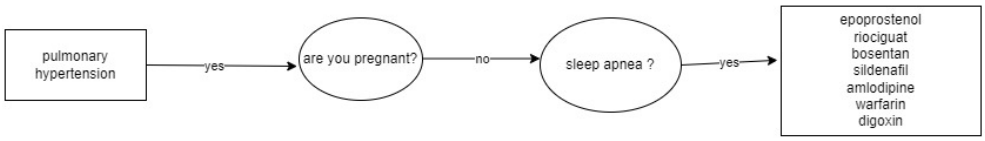




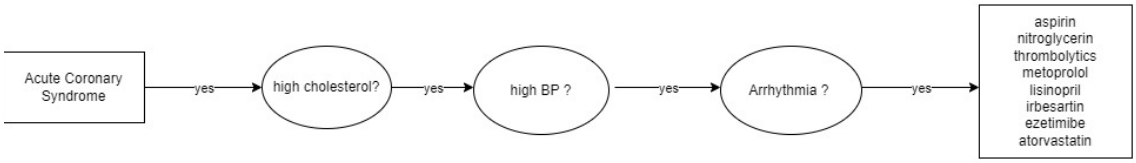


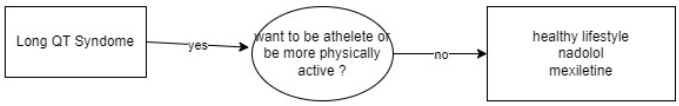


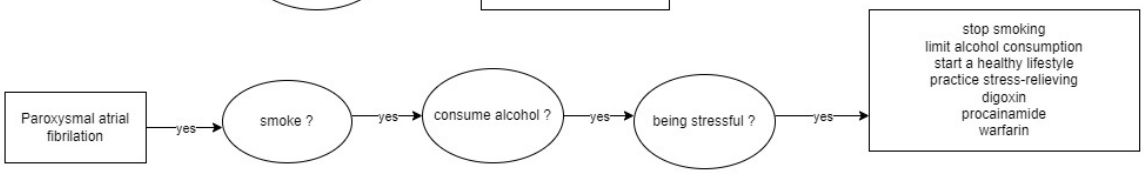


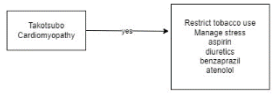

pulmonary hypertension
yes
are you pregnant?
no
sleep apnea ?
yes
epoprostenol
riociguat
bosentan
sildenafil
amlodipine
warfarin
digoxin


wolff-parkinson white syndrome
yes
have symptoms often ?
yes
catherer ablation
cardioversion
procedure
Adinosine

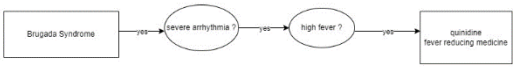

Acute Coronary Syndrome
yes
high cholesterol?
yes
high BP ?
yes
Arrhythmia ?
yes
aspirin
nitroglycerin
thrombolytics
metoprolol
lisinopril
irbesartin
ezetimibe
atorvastatin

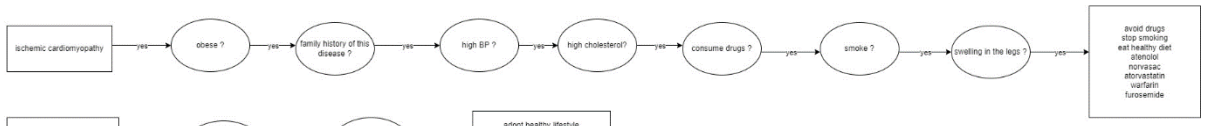

Long QT Syndome
yes
want to be athelete or be more physically active ?
no
healthy lifestyle
nadolol
mexiletine

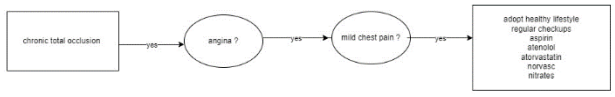

Paroxysmal atrial fibrilation
yes
smoke ?
yes
consume alcohol ?
yes
being stressful ?
yes
stop smoking
limit alcohol consumption
start a healthy lifestyle
practice stress-relieving
digoxin
procainamide
warfarin

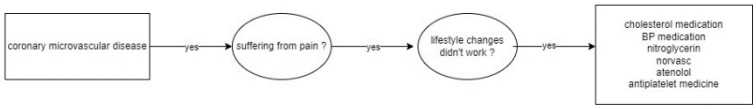

Takotsubo Cardiomyopathy
yes
Restrict tobacco use
Manage stress
aspirin
diuretics
benzazrazil
atenolol


ventricular tachycardia
yes
high BP ?
yes
high cholesterol?
yes
smoke ?
yes
consume alcohol ?
yes
eat low fat, low salt diet
exercise often
stop using tobacco
norvasc
atenolol
amiodarone


Brugada Syndrome
yes
severe arrhythmia ?
yes
high fever ?
yes
quinidine
fever reducing medicine

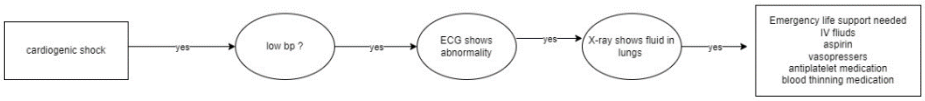

ischemic cardiomyopathy
yes
obese ?
yes
family history of this disease ?
yes
high BP ?
yes
high cholesterol?
yes
consume drugs ?
yes
smoke ?
yes
swelling in the legs ?
yes
avoid drugs
stop smoking
eat healthy diet
atenolol
norvasac
atorvastatin
warfarin
furosemide

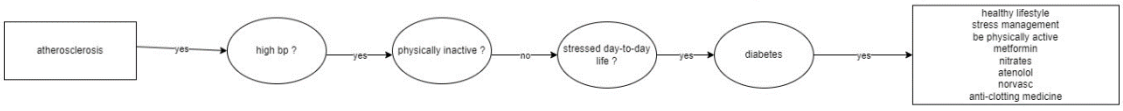

chronic total occlusion
yes
angina ?
yes
mild chest pain ?
yes
adopt healthy lifestyle
regular checkups
aspirin
atenolol
atorvastatin
norvasc
nitrates

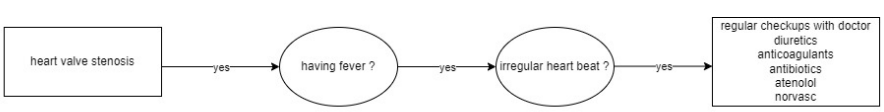

coronary microvascular disease
yes
suffering from pain ?
yes
lifestyle changes didn't work ?
yes
cholesterol medication
BP medication
nitroglycerin
norvasc
atenolol
antiplatelet medicine

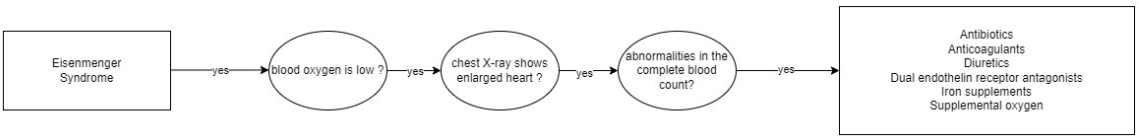

cardiac tamponade
yes
chest X-ray shows enlarged heart ?
yes
decrease in BP while inhaling ?
yes
abnormalities in 2D echo ?
yes
required hospitalization
bed-rest with leg elevation
dobutamine
IV fluids
external oxygen supply
pericardiocentasis

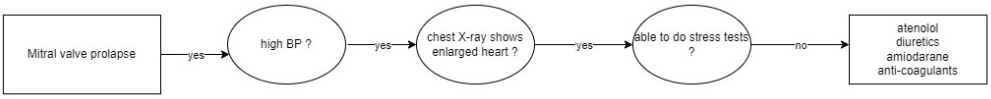

pulmonary embolism
yes
blood oxygen is low ?
yes
D-dimer blood result is high
yes
consume alcohol ?
yes
stop consuming alcohol
heparin
warfarin
regular checkups

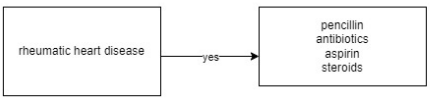

cardiogenic shock
yes
low bp ?
yes
ECG shows abnormality
yes
X-ray shows fluid in lungs
yes
Emergency life support needed
IV fluids
aspirin
vasopressers
antiplatelet medication
blood thinning medication


atherosclerosis
yes
high bp ?
yes
physically inactive ?
no
stressed day-to-day life ?
yes
diabetes
yes
healthy lifestyle
stress management
be physically active
metformin
nitrates
atenolol
norvasc
anti-clotting medicine

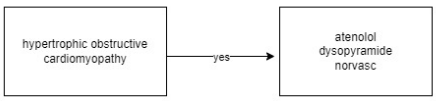

heart valve stenosis
yes
having fever ?
yes
irregular heart beat ?
yes
regular checkups with doctor
diuretics
anticoagulants
antibiotics
atenolol
norvasc

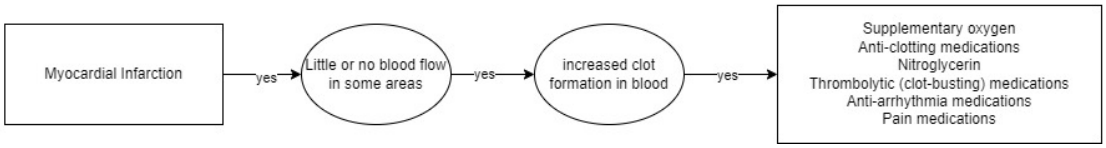

Eisenmenger Syndrome
yes
blood oxygen is low ?
yes
chest X-ray shows enlarged heart ?
yes
abnormalities in the complete blood count?
yes
Antibiotics
Anticoagulants
Diuretics
Dual endothelin receptor antagonists
Iron supplements
Supplemental oxygen


Mitral valve prolapse
yes
high BP ?
yes
chest X-ray shows enlarged heart ?
yes
able to do stress tests ?
no
atenolol
diuretics
amiodarane
anti-coagulants


rheumatic heart disease
yes
pencillin
antibiotics
aspirin
steroids


venous thrombeombolism
yes
age > 65 ?
yes
obesity ?
yes
low oxygen levels using pulse oximetry ?
yes
Apixaban
rivaroxaban


hypertrophic obstructive cardiomyopathy
yes
atenolol
dysopyramide
norvasc


Myocardial Infarction
yes
Little or no blood flow in some areas
yes
increased clot formation in blood
yes
Supplementary oxygen
Anti-clotting medications
Nitroglycerin
Thrombolytic (clot-busting) medications
Anti-arrhythmia medications
Pain medications

5. RULES

The expert system is rule-based system. It operates based on predefined rules. It analyzes questions using these rules and then delivers the solution. Rules are structured as “IF-THEN” statements.

5.1 Backward Chaining Rules:

1. IF CHESTPAIN = YES AND FATIGUE = NO AND SHORTNESSOFBREATH = YES AND FAINTING = YES AND NOISYGASPING = NO AND WEAKENINGINLEFTVENTRICLE = NO AND NECKTIGHTNESS = YES AND HEARTPALPITATION = YES AND CARDIACARST = YES

THEN DIS = VENTRICULAR TACHYCARDIA

2. IF CHESTPAIN = YES AND FATIGUE = NO AND SHORTNESSOFBREATH = YES AND FAINTING = YES AND NOISYGASPING = NO AND WEAKENINGINLEFTVENTRICLE = YES AND LOWBP = YES AND HEARTPALPITATION = YES

THEN DIS = TAKOTSUBO CARDIOMYOPATHY (BROKEN HEART SYNDROME)

3. IF CHESTPAIN = YES AND FATIGUE = NO AND SHORTNESSOFBREATH = YES AND FAINTING = YES AND NOISYGASPING = YES AND SEIZURES = YES

THEN DIS = LONG QT SYNDROME

4. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ALLTIME AND FEVER = NO AND SHOULDERPAIN = NO AND BACKPAIN = NO AND DIZZINESS = YES AND IRREGULARHEARTBEAT = NO AND NAUSEA = YES

THEN DIS = PAROXYSMAL ATRIAL FIBRILLATION

5. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = NEVER AND SWELLING = YES AND BLUESKIN = YES AND DIZZINESS = YES AND INCREASEDABDOMINALSIZE = YES

THEN DIS = PULMONARY HYPERTENSION

6. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ALL TIME AND FEVER = NO AND SHOULDERPAIN = NO AND BACKPAIN = YES AND COUGH = YES AND SCRTHVOICE = YES

THEN DIS = AORTIC ANEURSYM

7. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ALL TIME AND FEVER = NO AND SHOULDERPAIN = NO AND BACKPAIN = NO AND DIZZINESS = YES AND IRREGULARHEARTBEAT = YES

THEN DIS = VALVULAR HEART DISEASE

8. IF CORONARY ARTERY DISEASE = YES AND NAUSEA = YES AND INDIGESTION = YES AND FAINTING = YES AND HEAVYSWEATING = YES

THEN DIS = ACUTE CORONARY SYNDROME

9. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ALL TIME AND FEVER = NO AND SHOULDERPAIN = YES AND HEARTPALPITATION = YES AND NECKJWPAIN = YES

THEN DIS = CORONARY ARTERY DISEASE

10. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ALL TIME AND FEVER = YES AND HEARTMURMUR = NO AND SWOLLENJOINTS = YES AND UNCONTROLLEDMOVEMENT = YES AND SKINLUMPS = YES

THEN DIS = RHEUMATIC HEART DISEASE

11. IF PULMONARY EMBOLISM = YES AND REDLIMB = YES AND TENDERNESSOFTHIGH = YES AND WARMSKIN = YES

THEN DIS = VENOUS THROMBOEMBOLISM

12. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ALL TIME AND FEVER = YES AND HEARTMURMUR = YES AND SKINBUMPS = NO AND BLOODINCOUGH = YES AND DISCOLOREDSKIN = YES

THEN DIS = PULMONARY EMBOLISM

13. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ALL TIME AND FEVER = YES AND HEARTMURMUR = YES AND SKINBUMPS = YES

THEN DIS = ENDOCARDITIS

14. IF ARRHYTHMIA = YES AND SWELLING= YES AND WEIGHTGAIN = YES

THEN DIS = ISCHEMIC CARDIOMYOPATHY

15. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = LYINGDOWN AND FEVER = NO AND BLRYVISION = YES AND HEARTPALPITATION = YES

THEN DIS = ARRHYTHMIA

16. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = LYINGDOWN AND FEVER = NO AND BLRYVISION = NO AND DIFFICULTYINWALKING = YES AND SWELLING = YES AND PREFERSITTING = YES

THEN DIS = HEAT VALVE STENOSIS

17. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = LYINGDOWN AND FEVER = NO AND BLRYVISION = NO AND DIFFICULTYINWALKING = NO AND FAINTING = YES AND IRREGULARHEARTBEAT = YES AND HEARTPALPITATION = YES AND SEIZURES = YES

THEN DIS = BRUGADA SYNDROME

18. IF PERICARDITIS = YES AND INFECTIONORWOUND = YES AND CANCER = YES

THEN DIS = CARDIAC TAMPONADE

19. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = LYINGDOWN AND FEVER = YES AND SWTCHILL = YES AND HEARTPALPITATION = YES AND DIZZINESS = YES

THEN DIS = PERIPHERAL ARTERY DISEASE

20. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ACTIVITY AND FAINTING = NO AND SLURREDSPEECH= NO AND SLEEPPROBLEM = YES AND LACKOFENERGY = YES

THEN DIS = CORONARY MICROVASCULAR DISEASE

21. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ACTIVITY AND FAINTING = NO AND SLURREDSPEECH= NO AND SLEEPPROBLEM = NO AND DIZZINESS = YES AND IRREGULARHEARTBEAT = YES AND NAUSEA = YES AND HEARTPALPITATION = YES AND UPRARMPAIN = YES

THEN DIS = CHRONIC TOTAL OCCLUSION

22. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ACTIVITY AND FAINTING = NO AND SLURREDSPEECH= YES AND VSNLOSSONEI = YES AND LEGCRMP = YES

THEN DIS = ATHEROSCLEROSIS

23. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ACTIVITY AND FAINTING = YES AND DIZZINESS = YES AND HEARTMURMUR = YES AND SWELLING = YES AND NMBNESS = NO

THEN DIS = HYPERTROPHIC CARDIOMYOPATHY

24. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ACTIVITY AND FAINTING = YES AND DIZZINESS = YES AND HEARTMURMUR = YES AND SWELLING = YES AND NMBNESS = YES AND HSTRYCA= NO AND HSTRYMVP = YES

THEN DIS = MITRAL VALVE PROLAPSE

25. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ACTIVITY AND FAINTING = YES AND DIZZINESS = YES AND HEARTMURMUR = YES AND SWELLING = YES AND NMBNESS = YES AND HSTRYCA= YES

THEN DIS = HYPERTROPHIC OBSTRUCTIVE CARDIO MYOPATHY

26. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ACTIVITY AND FAINTING = YES AND DIZZINESS = YES AND HEARTMURMUR = NO AND BLOODINCOUGH = YES AND BLUECOLORSKIN = YES AND HEARTPALPITATION = YES

THEN DIS = EISENMENGER SYNDROME

27. IF MYOCINFCTN = YES AND LESSURIN = YES AND PALESKN= YES

THEN DIS = CARDIOGENIC SHOCK

28. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = SUDDEN AND SWTCHILL = YES AND BDYPAIN = YES

THEN DIS = MYOCARDIAL INFARCTION

29. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = DURINGEPISODESOFRAPIDHEARTRATE AND FAINTING = YES AND DIZZINESS = YES AND HEARTPALPITATION = YES AND ANXIETY = YES

THEN DIS = WOLFF-PARKINSON-WHITE SYNDROME

30. IF CHESTPAIN = YES AND FATIGUE = YES AND SHORTNESSOFBREATH = ACTIVITY AND FAINTING = YES AND DIZZINESS = YES AND HEARTMURMUR = NO AND BLOODINCOUGH = YES AND SLEEPPROBLEM = YES AND SWELLING = YES

THEN DIS = AORTIC STENOSIS

5.2 Forward Chaining Rules:

1. IF CORONARY ARTERY DISEASE=YES AND HIGH CHOLESTEROL?=YES AND SMOKING?= YES AND OVER WEIGHT?= YES
   THEN TREATMENT =
   - STOP SMOKING
   - INCLUDE HEALTHY DIET
   - MAINTAIN HEALTHY LIFESTYLE
   - EXERCISE
   - ATORVASTATTIN
   - ASPIRIN
   - VASODILATORS

2. IF AORTIC STENOSIS=YES AND SEVERE SYMPTOMS=NO AND PULMONARY CONGESTION=YES AND ANGINA?=YES
   THEN TREATMENT =
   HEALTHY LIFESTYLE
   MILD EXERCISE
   LISINOPRIL
   CARVEDILOL
   FUROSEMIDE
   DIGOXIN

3. IF ARRHYTHMIA?=YES AND ANGINA?= YES AND HIGH BP?=YES
   THEN TREATMENT =
   VERAPAMIL
   PROPRANOLOL
   WARFARIN

4. IF HYPERTROPHIC CARDIOMYOPATHY=YES AND HIGH CHOLESTEROL?= YES
   THEN TREATMENT =
   HEALTHY FOOD AND LIFESTYLE
   ATENOLOL
   DISOPRAMIDE
   VERAPRAMIL
   WARFIARIN
   ATORVASTATIN

5. IF PERIPHERAL ARTERY DISEASE=YES AND SMOKING?=YES AND PHYSICALLY ACTIVE?=NO AND HIGH CHOLESTEROL?=YES AND TYPE 2 DIABETES?=YES AND HAVE HIGH BP?=YES
   THEN TREATMENT =
   START WITH GENTLE PHYSICAL ACTIVITY
   ASPIRIN
   ATORVASTATIN
   MEGLITINIDES
   WARFARIN
   CILOSTAZOL

6. IF AORTIC ANEURYSM=YES AND HIGH BP?=YES AND HIGH CHOLESTEROL?=YES
   THEN TREATMENT =

ATENOLOL
LOSARTAN
ATORVASTATIN

7. IF VALVULAR HEART DISEASES=YES AND HIGH BP?=YES
   THEN TREATMENT =
   DIGIDOXIN
   ATENOLOL
   AMLODIPINE
   DIURETICS
   LISINOPRIL

8. IF ENDOCARDITIS=YES AND USED ANTIBIOTICS FOR 2-6 WEEKS ?=YES AND SYMPTOMS REDUCED ?=NO
   THEN TREATMENT =
   REPAIR DAMAGED HEART VALVE
   AORTIC VALVE REPLACEMENT SURGERY
   DRAINING ABSCESSES AND REPAIRING FISTULAS

9. IF PULMONARY HYPERTENSION=YES AND ARE YOU PREGNANT?=NO AND SLEEP APNEA ?=YES
   THEN TREATMENT =
   EPOPROSTENOL
   RIOCIGUAT
   BOSENTAN
   SILDENAFIL
   AMLODIPINE
   WARFARIN
   DIGOXIN

10. IF WOLFF-PARKINSON WHITE SYNDROME=YES AND HAVE SYMPTOMS OFTEN?=YES
    THEN TREATMENT =
    CATHERER ABLATION
    CARDIOVERSION PROCEDURE
    ADINOSINE

11. IF ACUTE CORONARY SYNDROME=YES AND HIGH CHOLESTEROL?= YES AND HIGH BP?= YES AND ARRHYTHMIA?= YES
    THEN TREATMENT =
    ASPIRIN
    NITROGLYCERIN
    THROMBOLYTICS

METOPROLOL
LISINOPRIL
IRBESARTIN
EZETIMIBE
ATORVASTATIN

12. IF LONG QT SYNDROME=YES AND WANT TO BE ATHLETE OR BE MORE PHYSICALLY ACTIVE ?=NO
THEN TREATMENT =
HEALTHY LIFESTYLE
NADOLOL
MEXILETINE

13. IF PAROXYSMAL ATRIAL FIBRILLATION=YES AND SMOKING?= YES AND CONSUME ALCOHOL ?=YES AND BEING STRESSFUL ? = YES
THEN TREATMENT =
STOP SMOKING
LIMIT ALCOHOL CONSUMPTION
START A HEALTHY LIFESTYLE
PRACTICE STRESS-RELIEVING
DIGOXIN
PROCAINAMIDE
WARFARIN

14. IF TAKOTSUBO CARDIOMYOPATHY=YES
THEN TREATMENT =
RESTRICT TOBACCO USE
MANAGE STRESS
ASPIRIN
DIURETICS
BENZAPRAZIL
ATENOLOL

15. IF VENTRICULAR TACHYCARDIA=YES AND HIGH BP?= YES AND HIGH CHOLESTEROL?=YES AND SMOKING?= YES AND CONSUME ALCOHOL ?=YES
THEN TREATMENT =
EAT LOW FAT, LOW SALT DIET
EXERCISE OFTEN
STOP USING TOBACCO
NORVASC
ATENOLOL
AMIODARONE

16. IF BRUGADA SYNDROME=YES AND ARRHYTHMIA?=YES AND HIGH FEVER ? = YES
    THEN TREATMENT =
        QUINIDINE
        FEVER REDUCING MEDICINE

17. IF ISCHEMIC CARDIOMYOPATHY=YES AND OBESE ? = YES AND FAMILY HISTORY OF THIS DISEASE ? = YES AND HIGH BP?= YES AND HIGH CHOLESTEROL?= YES AND CONSUME DRUGS ? = YES AND SMOKING?= YES AND SWELLING IN THE LEGS ? = YES
    THEN TREATMENT =
        AVOID DRUGS
        STOP SMOKING
        EAT HEALTHY DIET
        ATENOLOL
        NORVASAC
        ATORVASTATIN
        WARFARIN
        FUROSEMIDE

18. IF CHRONIC TOTAL OCCLUSION=YES AND ANGINA?= YES AND MILD CHEST PAIN ? = YES
    THEN TREATMENT =
        ADOPT HEALTHY LIFESTYLE
        REGULAR CHECKUPS
        ASPIRIN
        ATENOLOL
        ATORVASTATIN
        NORVASC
        NITRATES

19. IF CORONARY MICROVASCULAR DISEASE=YES AND SUFFERING FROM PAIN ? = YES AND LIFESTYLE CHANGES DIDN'T WORK ? = YES
    THEN TREATMENT =
        CHOLESTEROL MEDICATION
        BP MEDICATION
        NITROGLYCERIN
        NORVASC
        ATENOLOL
        ANTIPLATELET MEDICINE

20. IF CARDIAC TAMPONADE=YES AND CHEST X-RAY SHOWS ENLARGED HEART ? = YES AND DECREASE IN BP WHILE INHALING ? = YES AND ABNORMALITIES IN 2D ECHO ? = YES
    THEN TREATMENT =
        REQUIRED HOSPITALIZATION
        BED-REST WITH LEG ELEVATION
        DOBUTAMINE
        IV FLUIDS
        EXTERNAL OXYGEN SUPPLY
        PERICARDIOCENTESIS

21. IF PULMONARY EMBOLISM=YES AND BLOOD OXYGEN IS LOW ? = YES AND D-DIMER BLOOD RESULT IS HIGH? = YES AND CONSUME ALCOHOL ?=YES
    THEN TREATMENT =
        STOP CONSUMING ALCOHOL
        HEPARIN
        WARFARIN
        REGULAR CHECKUPS

22. IF CARDIOGENIC SHOCK=YES AND DECREASE IN BP WHILE INHALING ? = YES AND ABNORMALITIES IN ECG ? = YES AND X-RAY SHOWS FLUID IN LUNGS = YES
    THEN TREATMENT =
        EMERGENCY LIFE SUPPORT NEEDED
        IV FLIUDS
        ASPIRIN
        VASOPRESSERS
        ANTIPLATELET MEDICATION
        BLOOD THINNING MEDICATION

23. IF ATHEROSCLEROSIS=YES AND HIGH BP?= YES AND PHYSICALLY ACTIVE?=YES AND BEING STRESSFUL ? = YES AND TYPE 2 DIABETES?=YES
    THEN TREATMENT =
        HEALTHY LIFESTYLE
        STRESS MANAGEMENT
        BE PHYSICALLY ACTIVE
        METFORMIN
        NITRATES
        ATENOLOL

NORVASC
ANTI-CLOTTING MEDICINE

24. IF HEART VALVE STENOSIS=YES AND HIGH FEVER ? = YES AND IRREGULAR HEART BEAT ? = YES
THEN TREATMENT =
    REGULAR CHECKUPS WITH DOCTOR
    DIURETICS
    ANTICOAGULANTS
    ANTIBIOTICS
    ATENOLOL
    NORVASC

25. IF EISENMENGER SYNDROME=YES AND BLOOD OXYGEN IS LOW ?=YES AND CHEST X-RAY SHOWS ENLARGED HEART ? =YES AND ABNORMALITIES IN THE COMPLETE BLOOD COUNT?=YES
THEN TREATMENT =

ANTIBIOTICS
ANTICOAGULANTS
DIURETICS
DUAL ENDOTHELIN RECEPTOR ANTAGONISTS
IRON SUPPLEMENTS
SUPPLEMENTAL OXYGEN

26. IF MITRAL VALVE PROLAPSE=YES AND HIGH BP?= YES AND CHEST X-RAY SHOWS ENLARGED HEART ? = YES AND ABLE TO DO STRESS TESTS ? = YES
THEN TREATMENT =
    ATENOLOL
    DIURETICS
    AMIODARANE
    ANTI-COAGULANTS

27. IF RHEUMATIC HEART DISEASE=YES
THEN TREATMENT =
    PENCILLIN
    ANTIBIOTICS
    ASPIRIN
    STEROIDS

28. IF VENOUS THROMBOEMBOLISM=YES AND AGE? >65 AND OBESE ? = YES AND LOW OXYGEN LEVELS USING PULSE OXIMETRY ? = YES
    THEN TREATMENT =
        APIXABAN
        RIVAROXABAN

29. IF HYPERTROPHIC OBSTRUCTIVE CARDIO MYOPATHY=YES
    THEN TREATMENT =
        ATENOLOL
        DYSOPYRAMIDE
        NORVASC

30. IF MYOCARDIAL INFARCTION=YES AND LITTLE OR NO BLOOD FLOW IN SOME AREAS=YES AND INCREASED CLOT FORMATION IN BLOOD=YES
    THEN TREATMENT =
        SUPPLEMENTARY OXYGEN
        ANTI-CLOTTING MEDICATIONS
        NITROGLYCERIN
        THROMBOLYTIC (CLOT-BUSTING) MEDICATIONS
        ANTI-ARRHYTHMIA MEDICATIONS
        PAIN MEDICATIONS

6. PROGRAM IMPLEMENTATION

This program is implemented using cpp language. “Project1-A05295714.cpp” file includes Backward chaining, Forward chaining algorithms and main function.

6.1 Backward chaining algorithm

The backward chaining algorithm is used to diagnose the disease. The main function will call Attacks_BW(), which uses backward chaining to diagnose heart disease.

1. **Attacks_BW() –**
   This function calls the initializeDS() function to initialize data structures used in the backward chaining algorithm. After that, it calls the Process(goal) function with a goal as ‘disease’. The Process(goal) function returns a conclusion, and using this conclusion, Attacks_BW() calls the Prevention_FW(disease name) function for treatment recommendation. Additionally, Attacks_BW() is used to calculate the total running time required for both the backward chaining algorithm and the forward chaining algorithm.

2. **initializeDS() –**
   This function is used to initialize data structures used in backward chaining algorithm. It initializes conclusion list, variable list and clause variable list. In the variable list, each variable stores 'NI' values, indicating that it is not initialized.
3. **Process(goal) –**
   This function iterates through each conclusion variable in the conclusion variable list and find the value for goal and calls search_con(goal) to find the rule number, converts the rule number to a clause number using rule_to_clause(rule number), updates the variable list using update_VL(clause number), and validates the rule using validate_Ri(rule number, conclusion). If the conditions are met, it returns the conclusion.
4. **search_con(goal) –**
   This function is used to find rule number. It will find the matching variable i.e. goal in the conclusion list and gives the rule number.
5. **rule_to_clause(rule number) –**
   This function is used to calculate clause number from given rule number. For this program following formula is used
   CLAUSE NUMBER = 15*((RULE NUMBER/10)-1)+1
   As for each rule 15 spaces are allocated in the clause variable list.
6. **update_VL(clause number) –**
   For all the variables starting from clause number (value of function argument) to clause number + 14 in the clause variable list, this function will check whether all variables are instantiated. If not, it will ask the user multiple questions regarding symptoms and based on user's response it will update the variable list and derived global variable list. If a variable, we are trying to evaluate is the conclusion of any other rule then we will recursively call Process(variable) function.
7. **validate_Ri(rule number, conclusion) –**
   This function is used to validate the rule number. It means that it will check if the values of variables in the 'IF' part of the rule match the values in both the variable list and the derived global variable list. If they match, then the conclusion is returned otherwise, no value is returned.

6.2 Forward chaining algorithm

The forward chaining algorithm is used to recommend treatments to users based on diseases returned by the backward chaining process. Once the backward chaining algorithm concludes and identifies a disease, the program invokes the Prevention_FW(disease name) function to provide treatment recommendations specific to the identified disease.

1. **Prevention_FW(disease name) –**
   This function calls fw_initializeDS() to initialize data structures used by the forward chaining algorithms. After the initialization, it calls fw_Process(disease name).
2. **fw_initializeDS() –**

This function is used to initialize a variable list so that at the start, each variable stores 'NI' values, indicating that it is not initialized. It also initializes the clause variable list.

3. **fw_Process(disease name) –**
   This function instantiates the value of variable in the variable list. For example, it stores the value of that disease in the variable list and calls search_cvl(variable) function.
4. **search_cvl(disease name) –**
   This function is used to find clause number in the clause variable list. After that it will call fw_update_VL(clause number) and clause_to_rule(clause number) functions.
5. **fw_update_VL(clause number) –**
   For all the variables starting from clause number (value of function argument) to clause number + 9, this function will check whether all variables are instantiated. If not, it will prompt the user to provide the values of the variables and instantiate them in the variable list.
6. **clause_to_rule(clause number)** –
   This function is used to calculate rule number from clause number.
   For this program following formula is used –
   RULE NUMBER = (QUOTIENT(CLAUSE NUMBER/10)+1)*10
   As for each rule 10 spaces are allocated in the clause variable list.
   After rule calculation, this function will call validate_Ri(rule number) function.
7. **validate_Ri(rule number) –**
   This function is used to validate the rule number. It means that it will check if the values of variables in the 'IF' part of the rule match the values in the variable list. If they match, then the list of treatments to be recommended for the disease can be pushed into the derived conclusion list and global conclusion variable queue.

6.3 main function

This function is a starting point of program execution. It initiates the Attacks_BW() function, which uses backward chaining to diagnose heart disease. After the backward chaining algorithm finishes, it provides a conclusion. If the conclusion indicates the presence of a disease, the forward chaining algorithm is invoked to recommend treatment. However, if the user doesn't have any disease, the forward chaining algorithm won't be called for treatment recommendation.

7. SOURCE CODE

Program is implemented in cpp language.

Project1-A05295714.cpp

```cpp
//Have used Option 1, The algorithm

#include <iostream>
#include <vector>
#include <string>
#include<list>
#include<map>
#include <unordered_map>
#include <stack>
#include <algorithm>
#include <string.h>
#include <queue>
#include <string>

#include <chrono>
using namespace std;
using namespace std::chrono;



// Defining a structure to represent a rule
struct Rule_type {
    int ruleIndex;  // Index of the rule
    vector<string> conditions;
    string conclusion;
};

// Define a structure to represent a conclusion
struct Conclusion_list {
    int ruleno;
    string varname;
};



struct stack_type {
    int ruleno;
    int clauseno;
};

struct Derived_global{
    string variable;
```

```
    string instantiated;
    string value;
};

//forward chaining structures

using namespace std;
struct forward_rule_type {
    int ruleIndex;  // Index of the rule
    vector<string> conditions;
    vector<string> treatment;
};


// Initialize the Backward chaining rule list
vector<Rule_type> rule_list = {
{10,    {"ARRHYTHMIA=YES" , "SWELLING=YES" , "WEIGHTGAIN=YES"},
"ISCHEMIC CARDIOMYOPATHY"},

{20,    {"CHESTPAIN=YES", "FATIGUE=YES" , "SHORTNESSOFBREATH=WLD" , "FEVER=NO"
, "BLRYVISION=YES", "HEARTPALPITATION=YES"},
"ARRHYTHMIA"},

//change after stack test
//{10, {"CHESTPAIN=YES" , "FATIGUE=NO",  "SHORTNESSOFBREATH=YES"
,"FAINTING=YES", "NOISYGASPING=NO", "WEAKENINGINLEFTVENTRICLE=NO",
"NECKTIGHTNESS=YES", "HEARTPALPITATION=YES", "CARDIACARST=YES"}, "VENTRICULAR
TACHYCARDIA"},

//{20,  {"CHESTPAIN=YES" , "FATIGUE=NO" ,
"SHORTNESSOFBREATH=YES",  "FAINTING=YES",  "NOISYGASPING=NO",
"WEAKENINGINLEFTVENTRICLE=YES",  "LOWBP=YES", "HEARTPALPITATION=YES"},
"TAKOTSUBO CARDIOMYOPATHY (BROKEN HEART SYNDROME)"},

{30,    {"CHESTPAIN=YES", "FATIGUE=NO", "SHORTNESSOFBREATH=YES" ,
"FAINTING=YES", "NOISYGASPING=YES", "SEIZURES=YES"},"LONG QT SYNDROME"},

{40,    {"CHESTPAIN=YES", "FATIGUE=YES", "SHORTNESSOFBREATH=ALLTIME",
"FEVER=NO", "SHOULDERPAIN=NO", "BACKPAIN=NO", "DIZZINESS=YES",
"IRREGULARHEARTBEAT=NO", "NAUSEA=YES"}, "PAROXYSMAL ATRIAL FIBRILLATION"},

{50,    {"CHESTPAIN=YES", "FATIGUE=YES", "SHORTNESSOFBREATH=NEVER",
"SWELLING=YES", "BLUESKIN=YES",
"DIZZINESS=YES",  "INCREASEDABDOMINALSIZE=YES"},
"PULMONARY HYPERTENSION"},

{60,    {"CHESTPAIN=YES",
"FATIGUE=YES",  "SHORTNESSOFBREATH=ALLTIME",  "FEVER=NO",  "SHOULDERPAIN=NO",
 "BACKPAIN=YES", "COUGH=YES", "SCRTHVOICE=YES"}
```

```
,"AORTIC ANEURSYM"},

{70,    {"CHESTPAIN=YES", "FATIGUE=YES", "SHORTNESSOFBREATH=ALLTIME",
"FEVER=NO", "SHOULDERPAIN=NO", "BACKPAIN=NO", "DIZZINESS=YES",
"IRREGULARHEARTBEAT=YES"},
"VALVULAR HEART DISEASE"},

{80,    {"CORONARY ARTERY DISEASE=YES", "NAUSEA=YES", "INDIGESTION=YES",
"FAINTING=YES" ,"HEAVYSWEATING=YES"},
"ACUTE CORONARY SYNDROME"},

{90,    {"CHESTPAIN=YES", "FATIGUE=YES" , "SHORTNESSOFBREATH=ALLTIME ",
"FEVER=NO" , "SHOULDERPAIN=YES" , "HEARTPALPITATION=YES" , "NECKJWPAIN=YES"},
"CORONARY ARTERY DISEASE"},

{100,   {"CHESTPAIN=YES", "FATIGUE=YES",  "SHORTNESSOFBREATH=ALLTIME",
"FEVER=YES", "HEARTMURMUR=NO", "SWOLLENJOINTS=YES" ,
"UNCONTROLLEDMOVEMENT=YES", "SKINLUMPS=YES"},
"RHEUMATIC HEART DISEASE"},

{110,   {"PULMONARY EMBOLISM=YES" , "REDLIMB=YES", "TENDERNESSOFTHIGH=YES",
"WARMSKIN=YES"},
 "VENOUS THROMBOEMBOLISM"},

{120,   {"CHESTPAIN=YES", "FATIGUE=YES", "SHORTNESSOFBREATH=ALLTIME",
"FEVER=YES", "HEARTMURMUR=YES", "SKINBUMPS=NO", "BLOODINCOUGH=YES",
"DISCOLOREDSKIN=YES"},
"PULMONARY EMBOLISM"},

{130,   {"CHESTPAIN=YES" , "FATIGUE=YES" , "SHORTNESSOFBREATH=ALLTIME",
"FEVER=YES", "HEARTMURMUR=YES", "SKINBUMPS=YES" },
"ENDOCARDITIS"},

/*{140, {"ARRHYTHMIA=YES" , "SWELLING= YES" , "WEIGHTGAIN=YES"},
"ISCHEMIC CARDIOMYOPATHY"},

{150,   {"CHESTPAIN=YES", "FATIGUE=YES" , "SHORTNESSOFBREATH=LYINGDOWN" ,
"FEVER=NO" , "BLRYVISION=YES", "HEARTPALPITATION=YES"},
"ARRHYTHMIA"},*/
{140, {"CHESTPAIN=YES" , "FATIGUE=NO",  "SHORTNESSOFBREATH=YES"
,"FAINTING=YES", "NOISYGASPING=NO", "WEAKENINGINLEFTVENTRICLE=NO",
"NECKTIGHTNESS=YES", "HEARTPALPITATION=YES", "CARDIACARST=YES"}, "VENTRICULAR
TACHYCARDIA"},

{150,   {"CHESTPAIN=YES" , "FATIGUE=NO" ,
"SHORTNESSOFBREATH=YES",  "FAINTING=YES",  "NOISYGASPING=NO",
"WEAKENINGINLEFTVENTRICLE=YES",  "LOWBP=YES", "HEARTPALPITATION=YES"},
"TAKOTSUBO CARDIOMYOPATHY"},
```

```
{160,   {"CHESTPAIN=YES","FATIGUE=YES", "SHORTNESSOFBREATH=LYINGDOWN",
"FEVER=NO", "BLRYVISION=NO", "DIFFICULTYINWALKING=YES" , "SWELLING=YES" ,
"PREFERSITTING=YES"},
 "HEART VALVE STENOSIS"},

{170,   {"CHESTPAIN=YES" , "FATIGUE=YES", "SHORTNESSOFBREATH=LYINGDOWN" ,
"FEVER=NO" , "BLRYVISION=NO" , "DIFFICULTYINWALKING=NO", "FAINTING=YES" ,
"IRREGULARHEARTBEAT=YES" , "HEARTPALPITATION=YES" , "SEIZURES=YES" },
 "BRUGADA SYNDROME"},

{180,   {"PERICARDITIS=YES","INFECTIONORWOUND=YES", "CANCER=YES"},
"CARDIAC TAMPONADE"},

{190,   {"CHESTPAIN=YES" , "FATIGUE=YES" , "SHORTNESSOFBREATH=LYINGDOWN" ,
"FEVER=YES" , "SWEATCHILLS=YES" , "HEARTPALPITATION=YES", "DIZZINESS=YES"},
 "PERIPHERAL ARTERY DISEASE"},

{200,  {"CHESTPAIN=YES", "FATIGUE=YES", "SHORTNESSOFBREATH=ACTIVITY",
"FAINTING=NO", "SLURREDSPEECH=NO","SLEEPPROBLEM=YES", "LACKOFENERGY=YES"},
"CORONARY MICROVASCULAR DISEASE"},

{210,   {"CHESTPAIN=YES", "FATIGUE=YES" , "SHORTNESSOFBREATH=ACTIVITY" ,
"FAINTING=NO" , "SLURREDSPEECH=NO" , "SLEEPPROBLEM=NO" , "DIZZINESS=YES" ,
"IRREGULARHEARTBEAT=YES" , "NAUSEA=YES" , "HEARTPALPITATION=YES" ,
"UPRARMPAIN=YES"},
"CHRONIC TOTAL OCCLUSION"},

{220,   {"CHESTPAIN=YES" , "FATIGUE=YES",  "SHORTNESSOFBREATH=ACTIVITY" ,
"FAINTING=NO" , "SLURREDSPEECH=YES" , "ONEEYEVISIONLOSS=YES" , "LEGCRMP=YES"},
"ATHEROSCLEROSIS"},

{230,   {"CHESTPAIN=YES", "FATIGUE=YES" , "SHORTNESSOFBREATH=ACTIVITY" ,
"FAINTING=YES" , "DIZZINESS=YES" , "HEARTMURMUR=YES" , "SWELLING=YES" ,
"NMBNESS=NO"},
 "HYPERTROPHIC CARDIOMYOPATHY"},

{240, {"CHESTPAIN=YES", "FATIGUE=YES", "SHORTNESSOFBREATH=ACTIVITY ",
"FAINTING=YES", "DIZZINESS=YES", "HEARTMURMUR=YES", "SWELLING=YES" ,
"NMBNESS=YES" , "FAMILYHISTORYOFCARDIACARREST=NO" ,
"FAMILYHISTORYOFMITRALVALVEPROLAPSE=YES"},
"MITRAL VALVE PROLAPSE"},

{250,   {"CHESTPAIN=YES", "FATIGUE=YES" , "SHORTNESSOFBREATH=ACTIVITY",
"FAINTING=YES"," DIZZINESS=YES"," HEARTMURMUR=YES", "SWELLING=YES",
"NMBNESS=YES" , "FAMILYHISTORYOFCARDIACARREST=YES"},
"HYPERTROPHIC OBSTRUCTIVE CARDIO MYOPATHY"},
```

```
{260,   {"CHESTPAIN=YES", "FATIGUE=YES", "SHORTNESSOFBREATH=ACTIVITY" ,
"FAINTING=YES", "DIZZINESS=YES", "HEARTMURMUR=NO" , "BLOODINCOUGH=YES",
"BLUECOLORSKIN=YES" , "HEARTPALPITATION=YES"},
"EISENMENGER SYNDROME"},

{270,   {"MYOCINFCTN=YES" , "LESSURIN=YES", "PALESKN=YES"},
"CARDIOGENIC SHOCK"},

{280, {"CHESTPAIN=YES",  "FATIGUE=YES", "SHORTNESSOFBREATH=SUDDEN",
"SWEATCHILLS=YES",  "BDYPAIN=YES"}, "MYOCARDIAL INFARCTION"},

{290, {"CHESTPAIN=YES" ,
"FATIGUE=YES",  "SHORTNESSOFBREATH=DURINGEPISODESOFRAPIDHEARTRATE",  "FAINTING
=YES",  "DIZZINESS=YES",  "HEARTPALPITATION=YES" , "ANXIETY=YES"} , "WOLFF-
PARKINSON-WHITE SYNDROME"},

{300, {"CHESTPAIN=YES",  "FATIGUE=YES" , "SHORTNESSOFBREATH=ACTIVITY",
"FAINTING=YES",  "DIZZINESS=YES" , "HEARTMURMUR=NO" , "BLOODINCOUGH=YES" ,
"SLEEPPROBLEM=YES" ,"SWELLING=YES"}, "AORTIC STENOSIS"}

};

//forward chaining rules
vector<forward_rule_type> fw_rule_list = {

{10, {"CORONARY ARTERY DISEASE=yes", "High cholesterol?=yes" , "Smoking?=yes"
, "over weight?=yes"}, {"Stop smoking","Include healthy diet","Maintain
healthy lifestyle","Exercise","Atorvastattin","Aspirin","Vasodilators"}},
{20, {"AORTIC STENOSIS=yes" , "Severe symptoms=no" , "Pulmonary
congestion=yes" , "Angina?=yes"}, {"Healthy lifestyle","Mild
exercise","Lisinopril","Carvedilol","Furosemide","Digoxin"}},
{30, {"ARRHYTHMIA=yes" , "Angina?=yes" , "High BP?=yes"},
{"Verapamil","propranolol","warfarin"}},
{40, {"HYPERTROPHIC CARDIOMYOPATHY=yes" , "High cholesterol?=yes"}, {"Healthy
food and
lifestyle","atenolol","disopramide","verapramil","warfiarin","atorvastatin"}},
{50, {"PERIPHERAL ARTERY DISEASE=yes" , "Smoking?=yes" , "physically
active?=no" , "High cholesterol?=yes" , "type 2 diabetes?=yes" , "have High
BP?=yes"}, {"Start with gentle physical
activity","aspirin","atorvastatin","meglitinides","warfarin","cilostazol"}},
{60, {"AORTIC ANEURSYM=yes" , "High BP?=yes" , "High cholesterol?=yes" },
{"Atenolol","losartan","atorvastatin"}},
{70, {"VALVULAR HEART DISEASE=yes" , "High BP?=yes"},
{"Digidoxin","atenolol","amlodipine","diuretics","lisinopril"}},
{80, {"ENDOCARDITIS=yes" , "used antibiotics for 2-6 weeks ?=yes" , "symptoms
reduced ?=no"}, {"Repair damaged heart valve","aortic valve replacement
surgery","draining abscesses and repairing fistulas"}},
```

```
{90, {"PULMONARY HYPERTENSION=yes" , "are you pregnant?=no" , "sleep apnea
?=yes"},
{"Epoprostenol","riociguat","bosentan","sildenafil","amlodipine","warfarin","d
igoxin"}},
{100, {"WOLFF-PARKINSON-WHITE SYNDROME=yes" , "have symptoms often?=yes"},
{"Catherer ablation","cardioversion procedure","adinosine"}},
{110, {"ACUTE CORONARY SYNDROME=yes" , "High cholesterol?=yes" , "High
BP?=yes" , "Arrhythmia?=yes"},
{"Aspirin","nitroglycerin","thrombolytics","metoprolol","lisinopril","irbesart
in","ezetimibe","atorvastatin"}},
{120, {"LONG QT SYNDROME=yes" , "want to be athlete or be more physically
active ?=no"}, {"Healthy lifestyle","nadolol","mexiletine"}},
{130, {"PAROXYSMAL ATRIAL FIBRILLATION=yes" , "Smoking?=yes" , "consume
alcohol ?=yes" , "being stressful ?=yes"}, {"Stop smoking","limit alcohol
consumption","start a healthy lifestyle","practice stress-
relieving","digoxin","procainamide","warfarin"}},
{140, {"TAKOTSUBO CARDIOMYOPATHY=yes" }, {"Restrict tobacco use","manage
stress","aspirin","diuretics","benzaprazil","atenolol"}},
{150, {"VENTRICULAR TACHYCARDIA=yes" , "High BP?=yes" , "High
cholesterol?=yes" , "Smoking?=yes" , "consume alcohol ?=yes"}, {"Eat low fat,
low salt diet","exercise often","stop using
tobacco","norvasc","atenolol","amiodarone"}},
{160, {"BRUGADA SYNDROME=yes" , "Arrhythmia?=yes" , "high fever ?=yes"},
{"Quinidine","fever reducing medicine"}},
{170, {"ISCHEMIC CARDIOMYOPATHY=yes" , "obese ?=yes" , "family history of this
disease ?=yes" , "High BP?=yes" , "High cholesterol?=yes" , "consume drugs
?=yes" , "Smoking?=yes" , "swelling in the legs ?=yes" }, {"Avoid drugs","stop
smoking","eat healthy
diet","atenolol","norvasac","atorvastatin","warfarin","furosemide"}},
{180, {"CHRONIC TOTAL OCCLUSION=yes"  , "Angina?=yes" , "mild chest pain
?=yes"}, {"Adopt healthy lifestyle","regular
checkups","aspirin","atenolol","atorvastatin","norvasc","nitrates"}},
{190, {"CORONARY MICROVASCULAR DISEASE=yes" , "suffering from pain ?=yes" ,
"lifestyle changes didn't work ?=yes"}, {"Cholesterol medication","bp
medication","nitroglycerin","norvasc","atenolol","antiplatelet medicine"}},
{200, {"CARDIAC TAMPONADE=yes" , "chest X-ray shows enlarged heart ?=yes" ,
"decrease in BP while inhaling ?=yes" , "abnormalities in 2D echo ?=yes"},
{"Required hospitalization","bed-rest with leg elevation","dobutamine","iv
fluids","external oxygen supply","pericardiocentesis"}},
{210, {"PULMONARY EMBOLISM=yes" , "blood oxygen is low ?=yes" , "D-dimer blood
result is high?=yes" , "consume alcohol ?=yes"}, {"Stop consuming
alcohol","heparin","warfarin","regular checkups"}},
{220, {"CARDIOGENIC SHOCK=yes" , "decrease in BP while inhaling ?=yes" ,
"abnormalities in ECG ?=yes" , "X-ray shows fluid in lungs=yes"}, {"Emergency
life support needed","iv fliuds","aspirin","vasopressers","antiplatelet
medication","blood thinning medication"}},
{230, {"ATHEROSCLEROSIS=yes" , "High BP?=yes" , "physically active?=yes" ,
"being stressful ?=yes" , "type 2 diabetes?=yes"}, {"Healthy
```

```
lifestyle","stress management","be physically
active","metformin","nitrates","atenolol","norvasc","anti-clotting
medicine"}},
{240, {"HEART VALVE STENOSIS=yes" , "high fever ?=yes" , "irregular heart beat
?=yes"}, {"Regular checkups with
doctor","diuretics","anticoagulants","antibiotics","atenolol","norvasc"}},
{250, {"EISENMENGER SYNDROME=yes" , "blood oxygen is low ?=yes" , "chest X-ray
shows enlarged heart ?=yes" , "abnormalities in the complete blood
count?=yes"}, {"Antibiotics","Anticoagulants","Diuretics","Dual endothelin
receptor antagonists","Iron supplements","Supplemental oxygen"}},
{260, {"MITRAL VALVE PROLAPSE=yes" , "High BP?=yes" , "chest X-ray shows
enlarged heart ?=yes" , "able to do stress tests ?=yes"},
{"Atenolol","diuretics","amiodarane","anti-coagulants"}},
{270, {"RHEUMATIC HEART DISEASE=yes"},
{"Pencillin","antibiotics","aspirin","steroids"}},
{280, {"VENOUS THROMBOEMBOLISM=yes" , "Age greater than 65=yes" , "obese
?=yes" , "low oxygen levels using pulse oximetry ?=yes" }, {
"apixaban","rivaroxaban"}},
{290, {"HYPERTROPHIC OBSTRUCTIVE CARDIO MYOPATHY=yes"},
{"Atenolol","dysopyramide","norvasc"}},
{300, {"MYOCARDIAL INFARCTION=yes" , "Little or no blood flow in some
areas=yes" , "increased clot formation in blood=yes" }, {"Supplementary
oxygen","Anti-clotting medications","Nitroglycerin","Thrombolytic (clot-
busting) medications","Anti-arrhythmia medications","Pain medications"}}

};


vector<Conclusion_list> cncl_var_list(30);
unordered_map<string, string> varhashMap(30);
map<int,string> clausevarlist[400];
stack<stack_type> conclusionStack;
vector<string> conclusionList;
unordered_map<string,string> derivedGlobalVariableList(30);
//forward chaining
queue<string> con_var_q;
unordered_map<string, string> fw_varhashMap(30); // variable list
vector<string>derived_con_list{}; //derived_con_list
std::unordered_map<int,int>cls_var_pointer; //pointer
map<int,string> fw_clausevarlist[400];
queue<string>q;

int Attacks_BW();
string Process(string goal);
void initializeDS();
int search_con(string variable,int index);
int rule_to_clause(int ruleno);
void update_VL(int clauseno);
```

```
void validate_Ri(int ruleNo, string& conclusion);
//forward chaining
int Prevention_FW(string disease);
void fw_Process(string value);
void fw_initializeDS();
void search_cvl(string variable);
void clause_to_rule(int clauseno);
void fw_update_VL(int clauseno);
void validate_Ri(int ruleno);



void initializeDS(){
      //Intializing the conclusionList??
    for(Rule_type rule : rule_list){
        string conclusion = rule.conclusion;
        conclusionList.push_back(conclusion);
    }
     //Intializing the conclusion variable list
    for(int i=0;i<30;i++){
       cncl_var_list[i].ruleno = 10*(i+1);
       cncl_var_list[i].varname = "disease";

    }


    //Initializing the variable list
    for(Rule_type rule : rule_list){
        vector<string> condList = rule.conditions;
        for(string cond : condList){
            size_t pos = cond.find("=");
            string cond_name = cond.substr(0,pos);
            string cond_val = cond.substr(pos+1);
            //if cond_name not there in conclusion list add it to variable
list
            auto it = find(conclusionList.begin(), conclusionList.end(),
cond_name);
            if (it == conclusionList.end()) {
            varhashMap[cond_name] = "NI";
            }
            else {
            std::size_t index = std::distance(conclusionList.begin(), it);
            cncl_var_list[index].varname = cond_name;
            }
        }
    }

    int clausenum=1;
   //Initializing Clause Variable list
```

```
    for(Rule_type rule : rule_list){
         vector<string> condList = rule.conditions;
         array<string,15> temparr;
         for(int i=0;i<15;i++){
            temparr[i]="";
         }
        int ind=0;
        for(string cond : condList){
            size_t pos = cond.find("=");
            string cond_name = cond.substr(0,pos);
            string cond_val = cond.substr(pos+1);
            temparr[ind] = cond_name;
            ind++;
         }
         for(int i=0;i<15;i++){
            clausevarlist->insert(pair<int,string>(clausenum+i,temparr[i]));
         }
         clausenum += 15;
    }

    /*cout << "print variable list" << "\n";
    for (auto i = varhashMap.begin(); i != varhashMap.end(); i++)
          cout << i->first << " \t\t\t" << i->second << endl;

   cout << "printing clause variable list" << "\n";
    for (auto i = clausevarlist->begin(); i != clausevarlist->end(); i++)
          cout << i->first << " \t\t\t" << i->second << endl;

   cout << "printing conclusion variable list" << "\n";
   for(Conclusion_list c:cncl_var_list)  {
    cout<<"\n Rule: \n"<<c.ruleno;
    cout<<"\n conc variable: \n"<<c.varname;
   }*/
}

// finds the given variable name in conclusion list
int search_con(string variable,int index){
  /* for(int i=0; i < cncl_var_list.size(); i++){
    int res  = (cncl_var_list[index].varname).compare(variable);
       if(res == 0){
         return cncl_var_list[index].ruleno;
    //   }
        }
    return -1; */

    int res  = (cncl_var_list[index].varname).compare(variable);
    if(res==0){
       return cncl_var_list[index].ruleno;
```

```
    }
    else{
        return -1;
    }


}

// finds the clause no for given rule no
int rule_to_clause(int ruleno){
    stack_type item;
    item.ruleno = ruleno;
    item.clauseno = 15 * ((ruleno / 10) - 1) + 1;

    // Push the item onto the conclusionStack
    conclusionStack.push(item);

    return item.clauseno;

}

void update_VL(int clauseno){


    string userinput;

    //for each variable in clausevarlist we have to find its value in
varhashMap and cncl_var_list
    for(int clnum=clauseno;clnum<clauseno+15;clnum++){

        auto questionvar =  clausevarlist->find(clnum);
        string questoask = questionvar->second; // //for example cp in clause
variable list
    if(!(questoask == "")){

        bool inconclusionvarList=false;
        for(int i=0;i<30;i++){
        if(cncl_var_list[i].varname == questoask){
            inconclusionvarList=true;
            break;
        }}

        if(inconclusionvarList){

            Process(questoask);

        }
        else{
            auto it = varhashMap.find(questoask);
```

```
            string val = it->second;
            int res = val.compare("NI");
            if(res == 0){

                 cout << "\nDo you have " << questoask << " ? " << "\n";
                  if(questoask=="SHORTNESSOFBREATH")   {
                    cout << "\nPlease enter your symptom from anyone
below:\nWLD : If you have symptom while lying down\n";
                    cout << "\nNEVER: If you never have this symptom \n";
                    cout << "\nYES : If you are unsure when you are short of
breath \n";
                    cout << "\nALLTIME : If you are experiencing shortness of
breath all the time \n";
                    cout << "\nLYINGDOWN : If you experience shortness of
breath while lying down occasionally \n";
                    cout << "\nACTIVITY : If you experience shortness of
breath during physical activity \n";
                    cout << "\nSUDDEN : If you experience shortness of breath
suddenly \n";
                    cout << "\nDURINGEPISODESOFRAPIDHEARTRATE : If you
experience shortness of breath whenever heart rate increases\n";
                 }
                 else{
                    cout<< "\nEnter YES or NO as answer \n";
                 }
                 cin >> userinput;
                 varhashMap[questoask] = userinput;

            }
       }
    }
    else{
            break;
        }
    }

}

void validate_Ri(int ruleNo, string& conclusion) {
    bool conditionSatisfied;
    bool allconditionsSatisfied=true;
    Rule_type checkedRule;

    for (const Rule_type& rule : rule_list) {
        if (rule.ruleIndex == ruleNo) {
            checkedRule=rule;
            break;}
    }
```

```
    for(string cond : checkedRule.conditions){
        conditionSatisfied=true;
        size_t pos = cond.find("=");
        string cond_name = cond.substr(0,pos);
        string cond_val = cond.substr(pos+1);

        if (varhashMap[cond_name].compare(cond_val)!=0 &&
derivedGlobalVariableList[cond_name].compare(cond_val)!=0) {//check case
lower,upper
            conditionSatisfied = false;
            break;
            }

        }
    if(!conditionSatisfied){
        allconditionsSatisfied=false;
        }

    if(allconditionsSatisfied){

        conclusion=checkedRule.conclusion;
        derivedGlobalVariableList[conclusion]="YES";
        stack_type top;
        top=conclusionStack.top();
        if(top.ruleno==checkedRule.ruleIndex){
            conclusionStack.pop();
            }
        //cout<< "all conditions satisfied: " <<conclusion;
    }
    //cout<<"after setting derived
global"<<derivedGlobalVariableList[conclusion];

}


string Process(string goal) {
    //loop through each conclusion variable to find value for goal which is
disease initially
    int i=0;
    string conclusion="";

    while(i<cncl_var_list.size()){

        int rulenum = search_con(goal,i);

        if(rulenum==-1){
```

```
            i=i+1;
            continue;
        }

        int clauseno = rule_to_clause(rulenum);


        update_VL(clauseno);

        /*cout << "print variable list" << "\n";
        for (auto i = varhashMap.begin(); i != varhashMap.end(); i++)
            cout << i->first << " \t\t\t" << i->second << endl;*/

        conclusion="";
        validate_Ri(rulenum,conclusion);

        if (!conclusion.empty()) {
                    // A conclusion was found, you can save it or use it as
needed
                    break;  // End the program after updating the Variable
List
                }
        else{
            i=i+1;
        }
    }
return conclusion;
}

int Attacks_BW(){
    vector<int> values(10000);
    auto start = high_resolution_clock::now();
    initializeDS();
    cout<< "Welcome to Intelligent Cardiac Diagnosis System \n";
    cout<< "Please answer the following questions so we can diagnose your
disease \n";
    string conclusion = Process("disease");
    cout<< "\nYou have been diagnosed with: \n"<<conclusion;


 // Call the function, here sort()
 sort(values.begin(), values.end());

 // Get ending timepoint
 auto stop = high_resolution_clock::now();

 // Get duration. Substart timepoints to
 // get duration. To cast it to proper unit
```

```
 // use duration cast method
    auto duration = duration_cast<microseconds>(stop - start);
    cout<<"\n";
    cout<<"\nTime taken by Backward chaining:\n"<<duration.count() <<"
microseconds";

    start = high_resolution_clock::now();

    Prevention_FW(conclusion);
    // Call the function, here sort()
    sort(values.begin(), values.end());

    // Get ending timepoint
    stop = high_resolution_clock::now();

 // Get duration. Substart timepoints to
 // get duration. To cast it to proper unit
 // use duration cast method
    duration = duration_cast<microseconds>(stop - start);
    cout<<"\nTime taken by Forward chaining:\n"<<duration.count() <<"
microseconds";
    return 0;
}

//forward chaining
void fw_initializeDS(){

    //Initializing the variable list
    for(forward_rule_type rule : fw_rule_list){
        vector<string> condList = rule.conditions;
        for(string cond : condList){
            size_t pos = cond.find("=");
            string cond_name = cond.substr(0,pos);
            string cond_val = cond.substr(pos+1);

        fw_varhashMap[cond_name] = "NI";
        }
    }

    int clausenum=1;
   //Initializing Clause Variable list
    for(forward_rule_type rule : fw_rule_list){
         vector<string> condList = rule.conditions;
         array<string,10> temparr;
         for(int i=0;i<10;i++){
            temparr[i]="";
         }
       int ind=0;
```

```
       for(string cond : condList){
            size_t pos = cond.find("=");
            string cond_name = cond.substr(0,pos);
            string cond_val = cond.substr(pos+1);
            temparr[ind] = cond_name;
            ind++;
        }
        for(int i=0;i<10;i++){
            fw_clausevarlist-
>insert(pair<int,string>(clausenum+i,temparr[i]));
        }
        clausenum += 10;
    }

    /*cout << "print variable list" << "\n" ;
    for (auto i = fw_varhashMap.begin(); i != fw_varhashMap.end(); i++)
          cout << i->first << " \t\t\t" << i->second << endl;

   cout << "printing clause variable list" << "\n";
    for (auto i = fw_clausevarlist->begin(); i != fw_clausevarlist->end();
i++)
          cout << i->first << " \t\t\t" << i->second << endl;
*/
}

void search_cvl(string variable){
    //cout << "search_cvl variable name " << " " << variable <<
endl;//ruleno=-1 case??
    for(int i= 0;i<400;i++)
    {
        for(auto it=fw_clausevarlist[i].begin();
it!=fw_clausevarlist[i].end(); ++it)
        {
            if(it->second == variable)
            {
                int clauseno = it->first;
                //cout << "clauseno" << " " <<clauseno << endl;
                fw_update_VL(clauseno);
                clause_to_rule(clauseno);
                break;
            }
        }
    }

}

void fw_update_VL(int clauseno)
{
```

```
    string userinput;
    for(int clnum=clauseno;clnum<clauseno+10;clnum++)
    {

        auto questionvar =  fw_clausevarlist->find(clnum);

            string questoask = questionvar->second; // DIS HICOL

            if(!(questoask == "")){


                //cout<< "qn from clause varlist:"<<questoask << "\n";//handle
questoask==""here
                auto it = fw_varhashMap.find(questoask);
                string val = it->second;
                int res = val.compare("NI");//result if variable Instantiated
in variable list or not
                if(res ==0)
            {
                cout<<questoask <<""<<endl;
                cin>>userinput;
                fw_varhashMap[questoask] = userinput;

                        //cout << "value intialized in variable list";

            }

        }
        else
        {
            break;
        }
    }


}

void clause_to_rule(int clauseno){

    int ruleno= ((clauseno/10)+1)*10;
    //cout << "rule no " << " " << ruleno << endl;//ruleno=-1 case??
    validate_Ri(ruleno);


}

void validate_Ri(int ruleno)
{
```

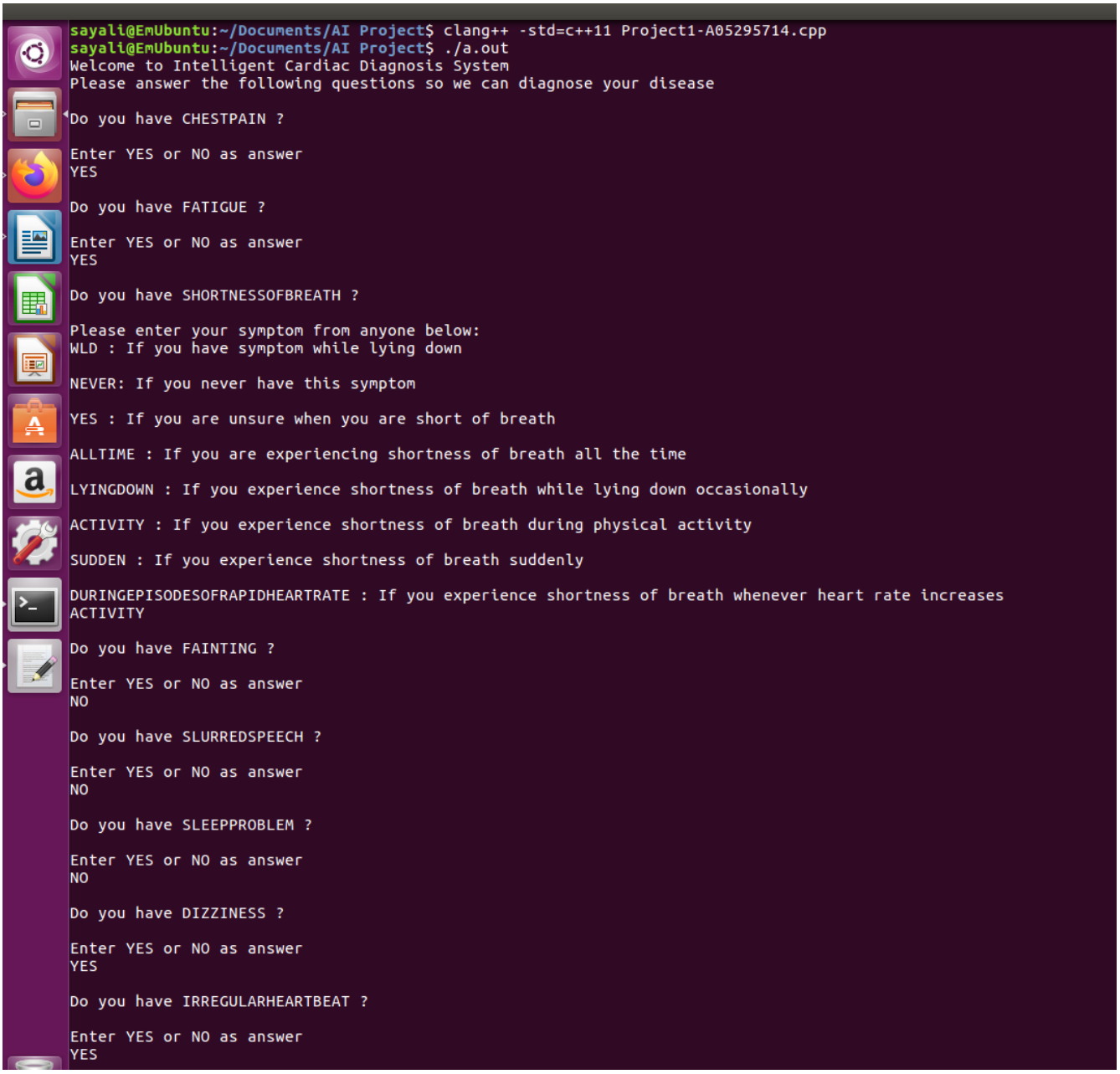
```
    bool conditionSatisfied;
    bool allconditionsSatisfied=true;
    forward_rule_type checkedRule;

    for (const forward_rule_type& rule : fw_rule_list) {
        if (rule.ruleIndex == ruleno) {
            checkedRule=rule;
            break;
    }
    }
    //cout<< "Rule's validity checking : "<<checkedRule.ruleIndex;

    for(string cond : checkedRule.conditions){
        conditionSatisfied=true;
        size_t pos = cond.find("=");
        string cond_name = cond.substr(0,pos);
        string cond_val = cond.substr(pos+1);

    //cout<<"cond_val "<<cond_val<<endl;
    std::string VLvalue = fw_varhashMap[cond_name];
    //cout<<"VLvalue "<<VLvalue<<endl;
    std::transform(VLvalue.begin(), VLvalue.end(), VLvalue.begin(),
::toupper);
    //cout<<"UPPER VLvalue "<<VLvalue<<endl;
    std::string cond_val_uppr = cond_val;
    //cout<<"cond_val "<<cond_val<<endl;
    std::transform(cond_val_uppr.begin(), cond_val_uppr.end(),
cond_val_uppr.begin(), ::toupper);
    //cout<<"UPPER cond_val_uppr "<<cond_val_uppr<<endl;


        if (VLvalue!=cond_val_uppr) {//check case lower,upper
            conditionSatisfied = false;
            //cout<< "breaks at : "<< cond_name;
            break;
            }
        }
    if(!conditionSatisfied){
        allconditionsSatisfied=false;
    cout<<"Exercise regularly"<<endl;
        }
    //cout<< "all conditions satisfied: " <<allconditionsSatisfied;

    if(allconditionsSatisfied){
       string con_var="Treatment";
       // cout<< "all conditions satisfied: " <<endl;
    vector<string> trtmntList=checkedRule.treatment;
    cout<< "\nTreatment we recommend is: \n " <<endl;
```

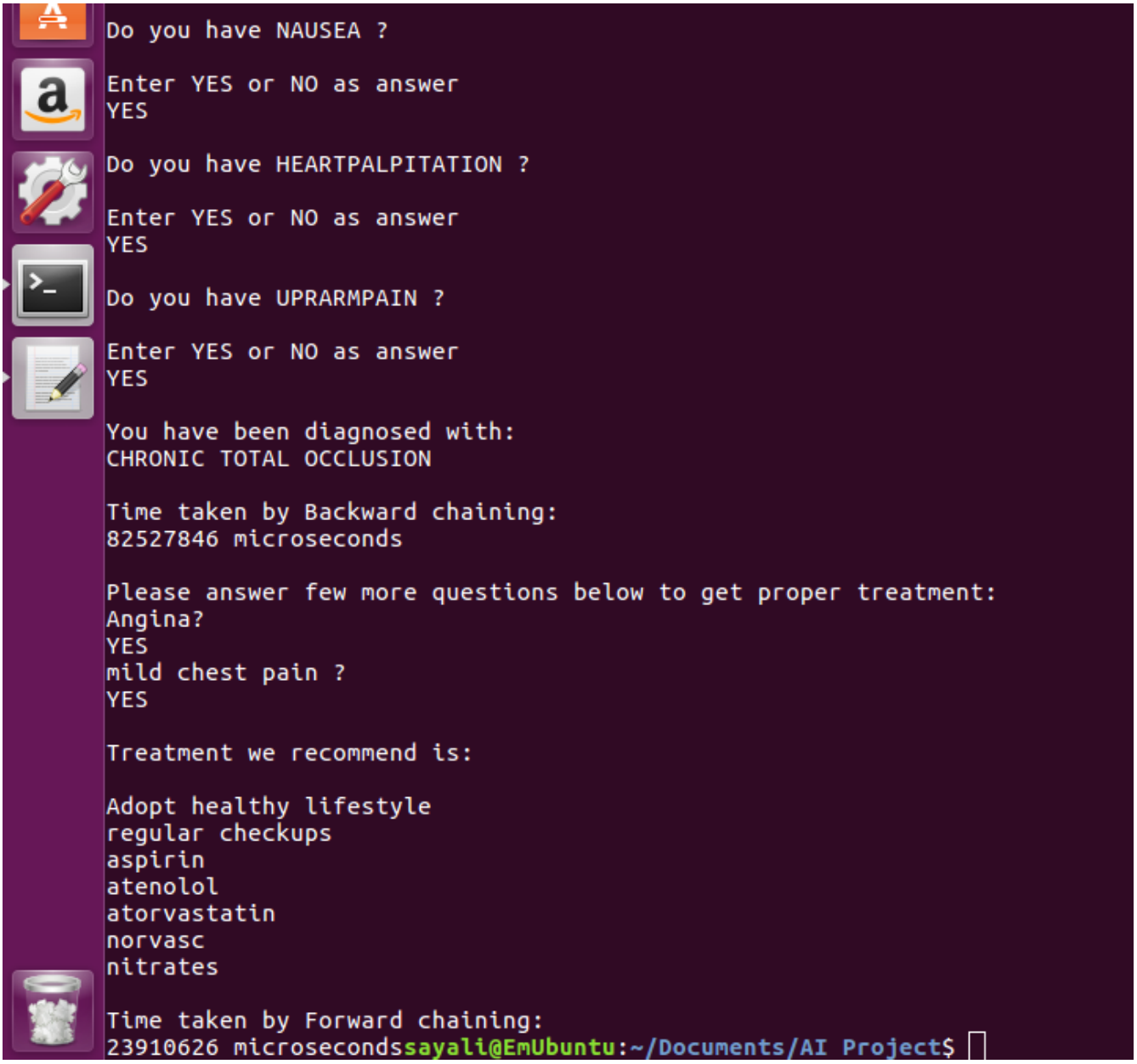
```
    for(string trtmnt : trtmntList)
    {

        cout<<trtmnt <<endl;
        derived_con_list.push_back(trtmnt);

    }
    q.push(con_var);

    }


}

void fw_Process(string value) {
    for (auto i = fw_varhashMap.begin(); i != fw_varhashMap.end(); i++)
    {
        //cout <<"variable and value -----"<< value <<" " << i->first<< endl;

        if(value == i->first)
        {
            fw_varhashMap[value] = "YES";
            //cout <<"value of var list after update"<< fw_varhashMap[value]
<< endl;

        }

        //cout<<"After update :"<<endl;
            //cout << i->first << " \t\t\t" << i->second << endl;

    }

    search_cvl(value);
}

int Prevention_FW(string dis){

    fw_initializeDS();
    cout<<"\n";
    cout<<"\nPlease answer few more questions below to get proper treatment:
\n";
    fw_Process(dis);// value of disease
    return 0;
}

int main(){
    Attacks_BW();
    return 0;
}
```

8. PROGRAM EXECUCATION RESULTS

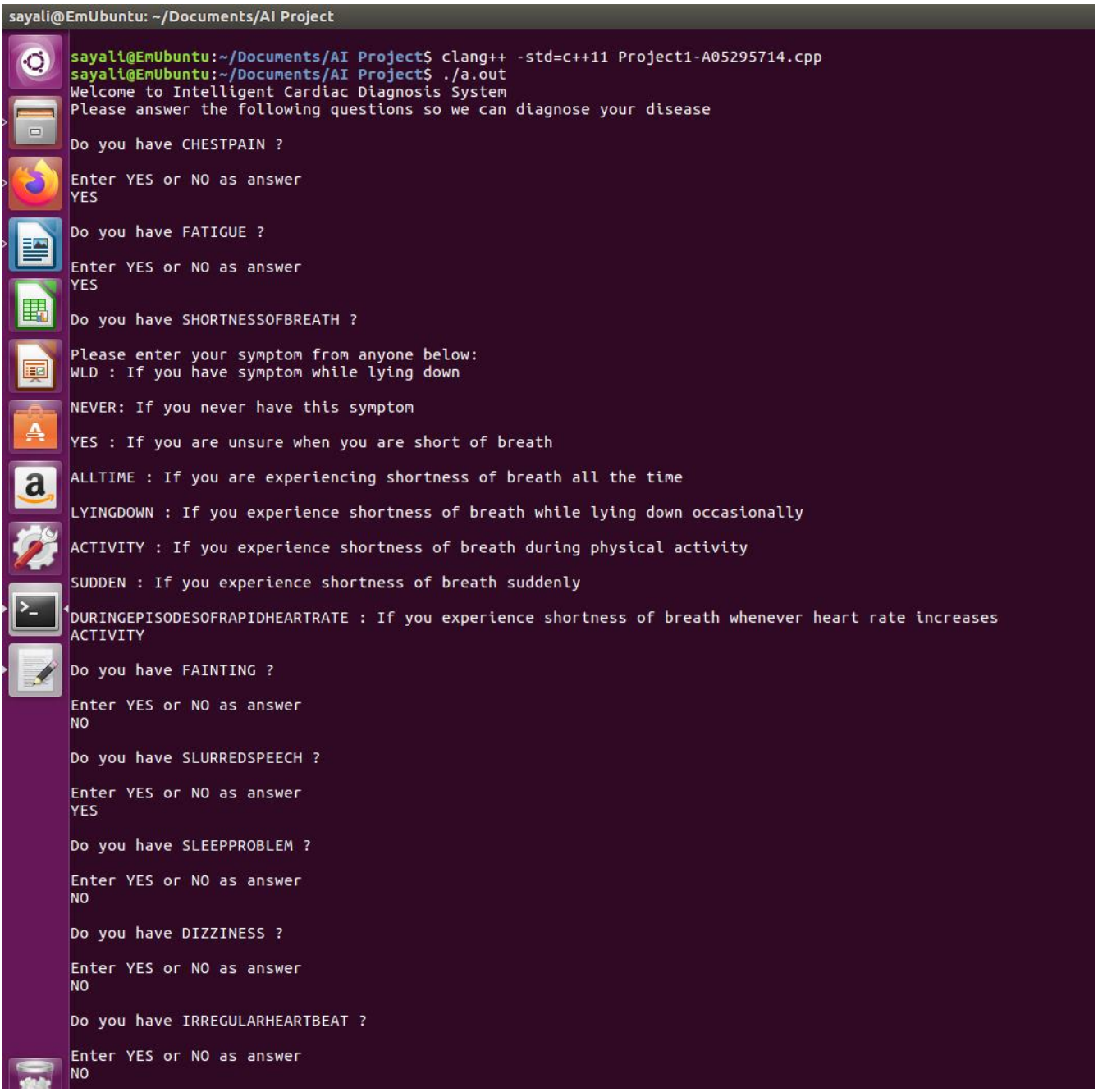
Result of program execution for 5 patients –

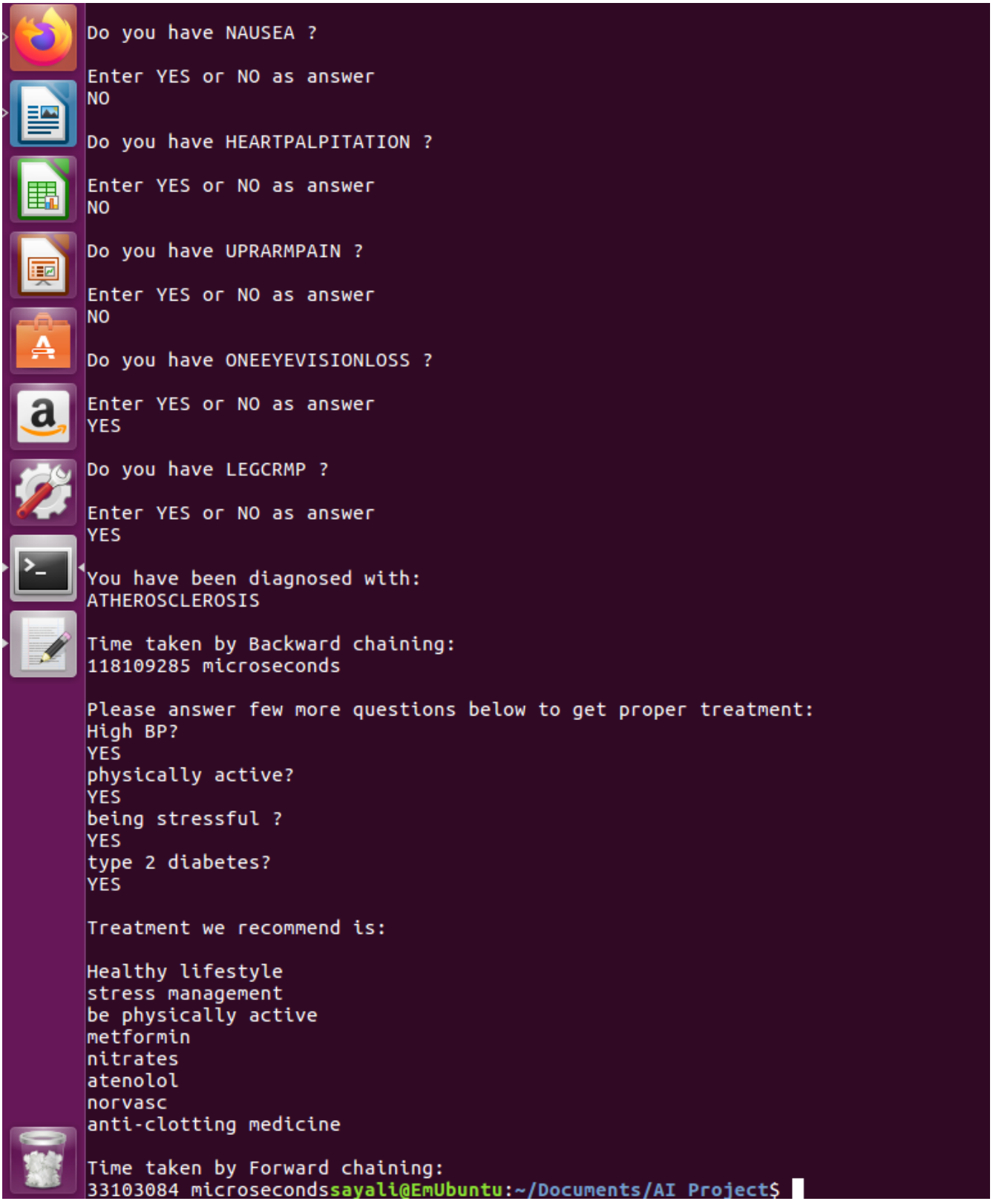
Patient 1 - CHRONIC TOTAL OCCLUSION



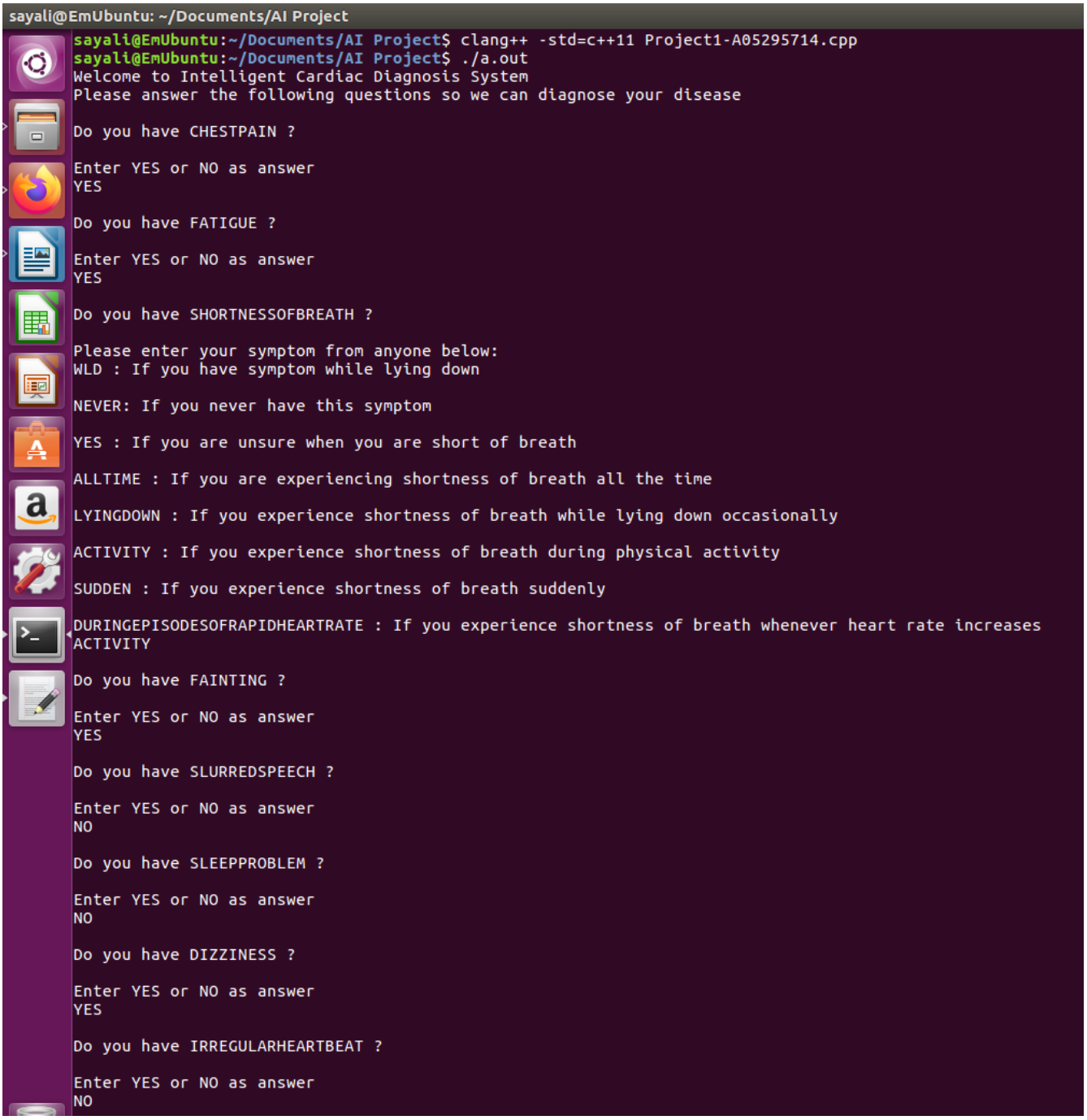

Do you have NAUSEA ?
Enter YES or NO as answer
YES
Do you have HEARTPALPITATION ?
Enter YES or NO as answer
YES
Do you have UPRARMPAIN ?
Enter YES or NO as answer
YES
You have been diagnosed with:
CHRONIC TOTAL OCCLUSION
Time taken by Backward chaining:
82527846 microseconds
Please answer few more questions below to get proper treatment:
Angina?
YES
mild chest pain ?
YES
Treatment we recommend is:
Adopt healthy lifestyle
regular checkups
aspirin
atenolol
atorvastatin
norvasc
nitrates
Time taken by Forward chaining:
23910626 microsecondssayali@EmUbuntu:~/Documents/AI Project$

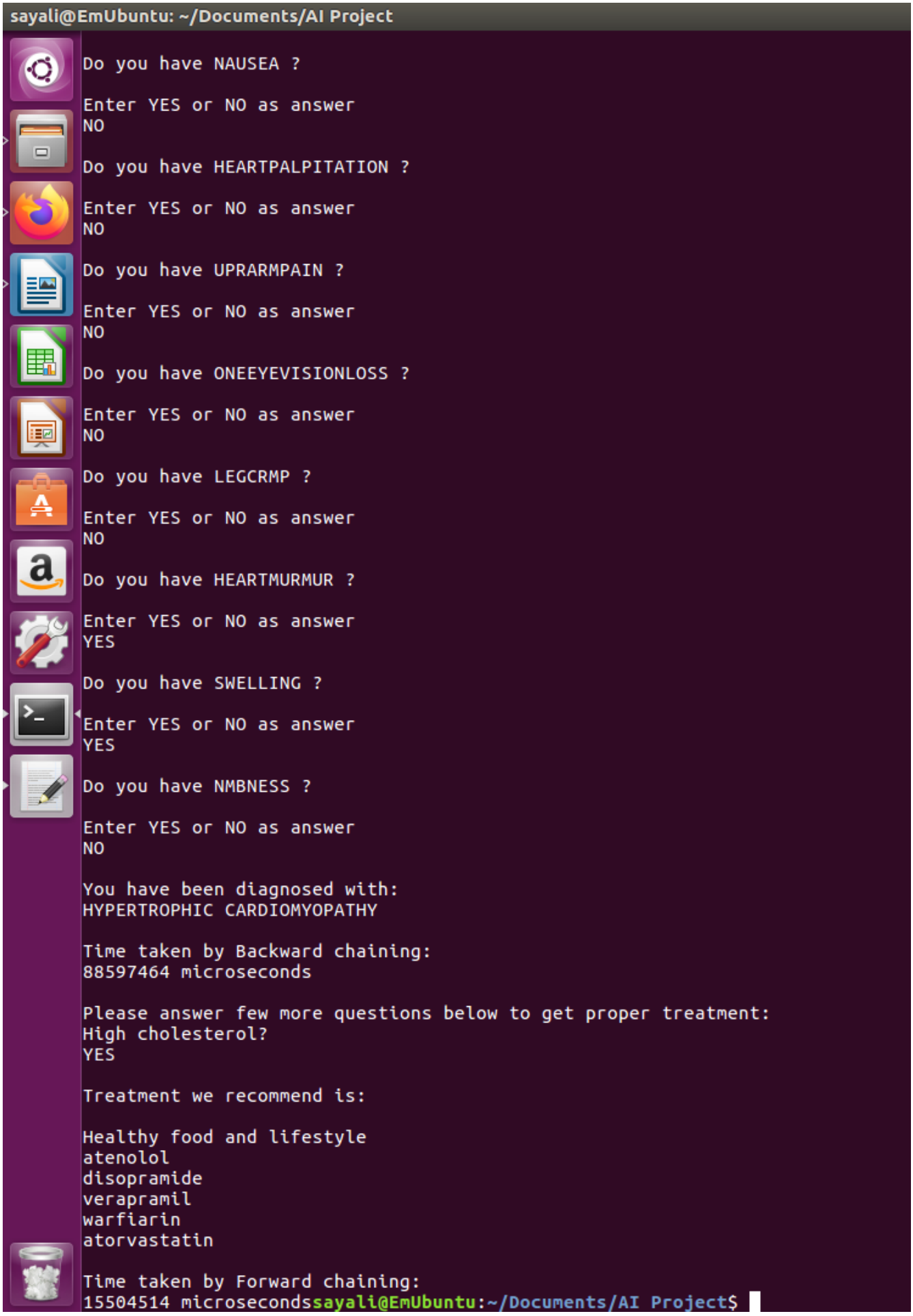
Patient 2 – ATHEROSCLEROSIS



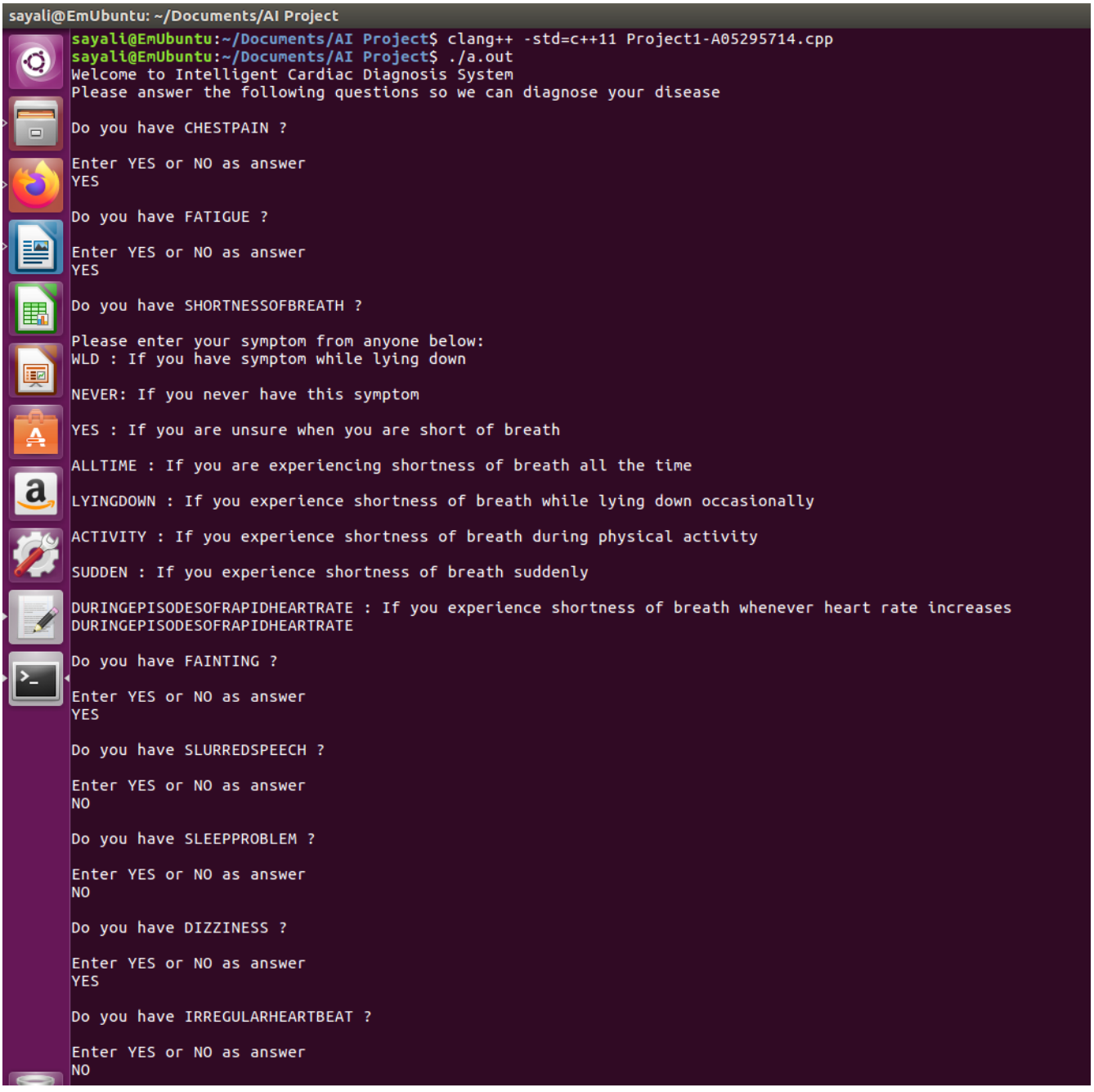

Do you have NAUSEA ?
Enter YES or NO as answer
NO
Do you have HEARTPALPITATION ?
Enter YES or NO as answer
NO
Do you have UPRARMPAIN ?
Enter YES or NO as answer
NO
Do you have ONEEYEVISIONLOSS ?
Enter YES or NO as answer
YES
Do you have LEGCRMP ?
Enter YES or NO as answer
YES
You have been diagnosed with:
ATHEROSCLEROSIS
Time taken by Backward chaining:
118109285 microseconds
Please answer few more questions below to get proper treatment:
High BP?
YES
physically active?
YES
being stressful ?
YES
type 2 diabetes?
YES
Treatment we recommend is:
Healthy lifestyle
stress management
be physically active
metformin
nitrates
atenolol
norvasc
anti-clotting medicine
Time taken by Forward chaining:
33103084 microsecondssayali@EmUbuntu:~/Documents/AI Project$

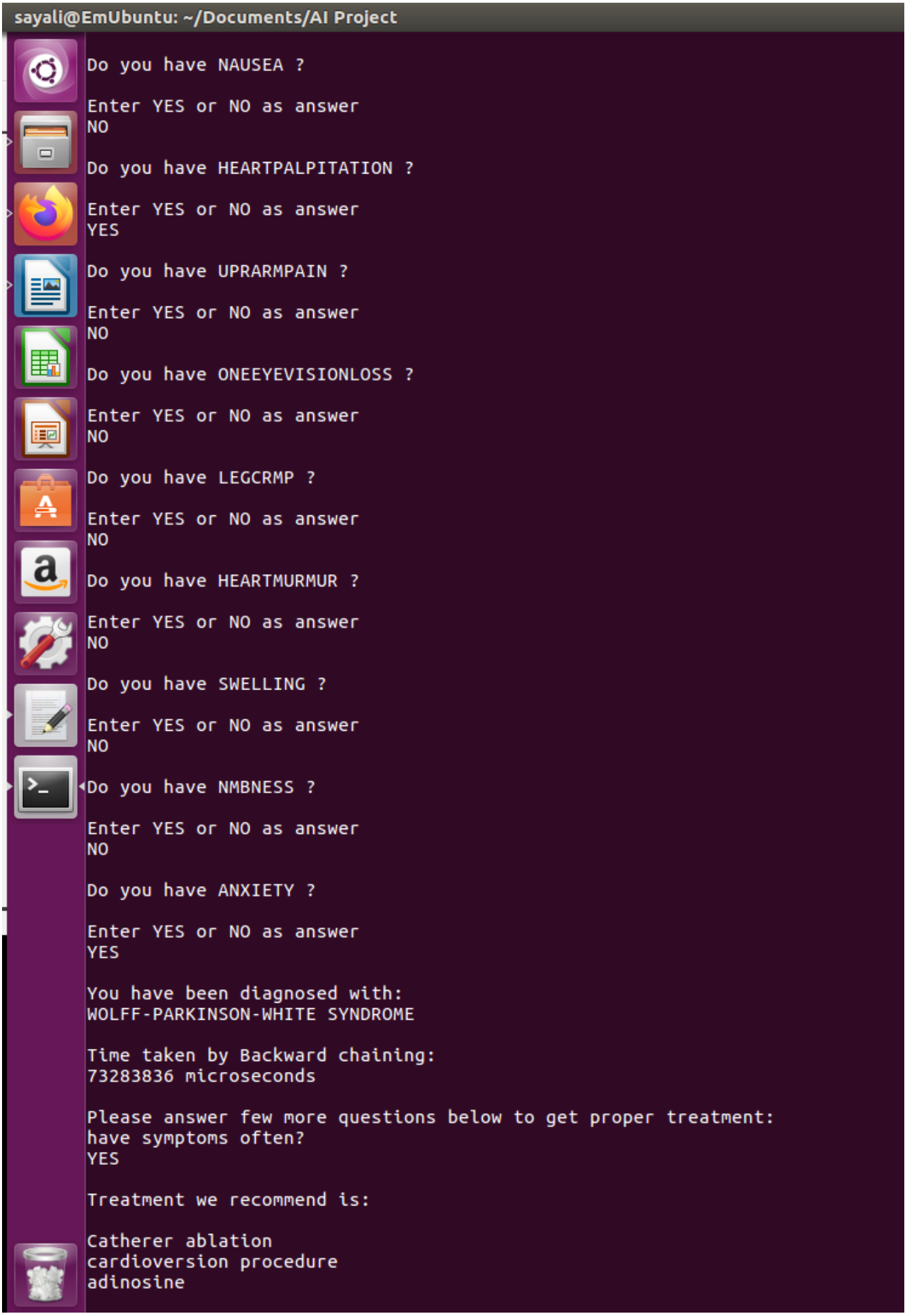
Patient 3 - HYPERTROPHIC CARDIOMYOPATHY

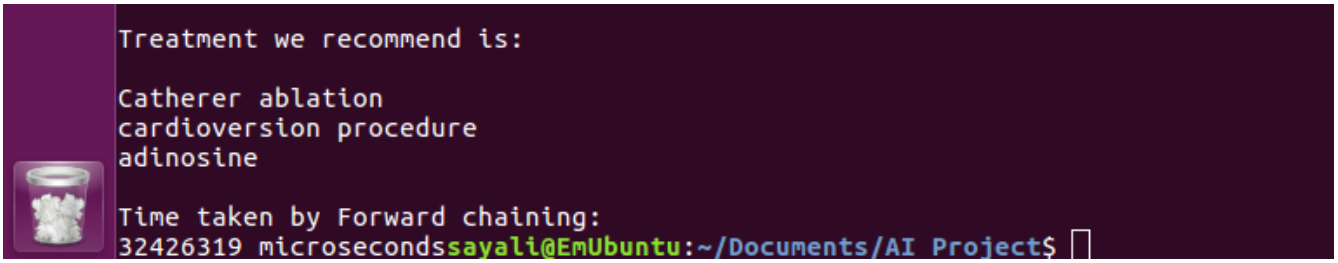



sayali@EmUbuntu: ~/Documents/AI Project
Do you have NAUSEA ?
Enter YES or NO as answer
NO
Do you have HEARTPALPITATION ?
Enter YES or NO as answer
NO
Do you have UPRARMPAIN ?
Enter YES or NO as answer
NO
Do you have ONEEYEVISIONLOSS ?
Enter YES or NO as answer
NO
Do you have LEGCRMP ?
Enter YES or NO as answer
NO
Do you have HEARTMURMUR ?
Enter YES or NO as answer
YES
Do you have SWELLING ?
Enter YES or NO as answer
YES
Do you have NMBNESS ?
Enter YES or NO as answer
NO
You have been diagnosed with:
HYPERTROPHIC CARDIOMYOPATHY
Time taken by Backward chaining:
88597464 microseconds
Please answer few more questions below to get proper treatment:
High cholesterol?
YES
Treatment we recommend is:
Healthy food and lifestyle
atenolol
disopramide
verapramil
warfiarin
atorvastatin
Time taken by Forward chaining:
15504514 microsecondssayali@EmUbuntu:~/Documents/AI Project$

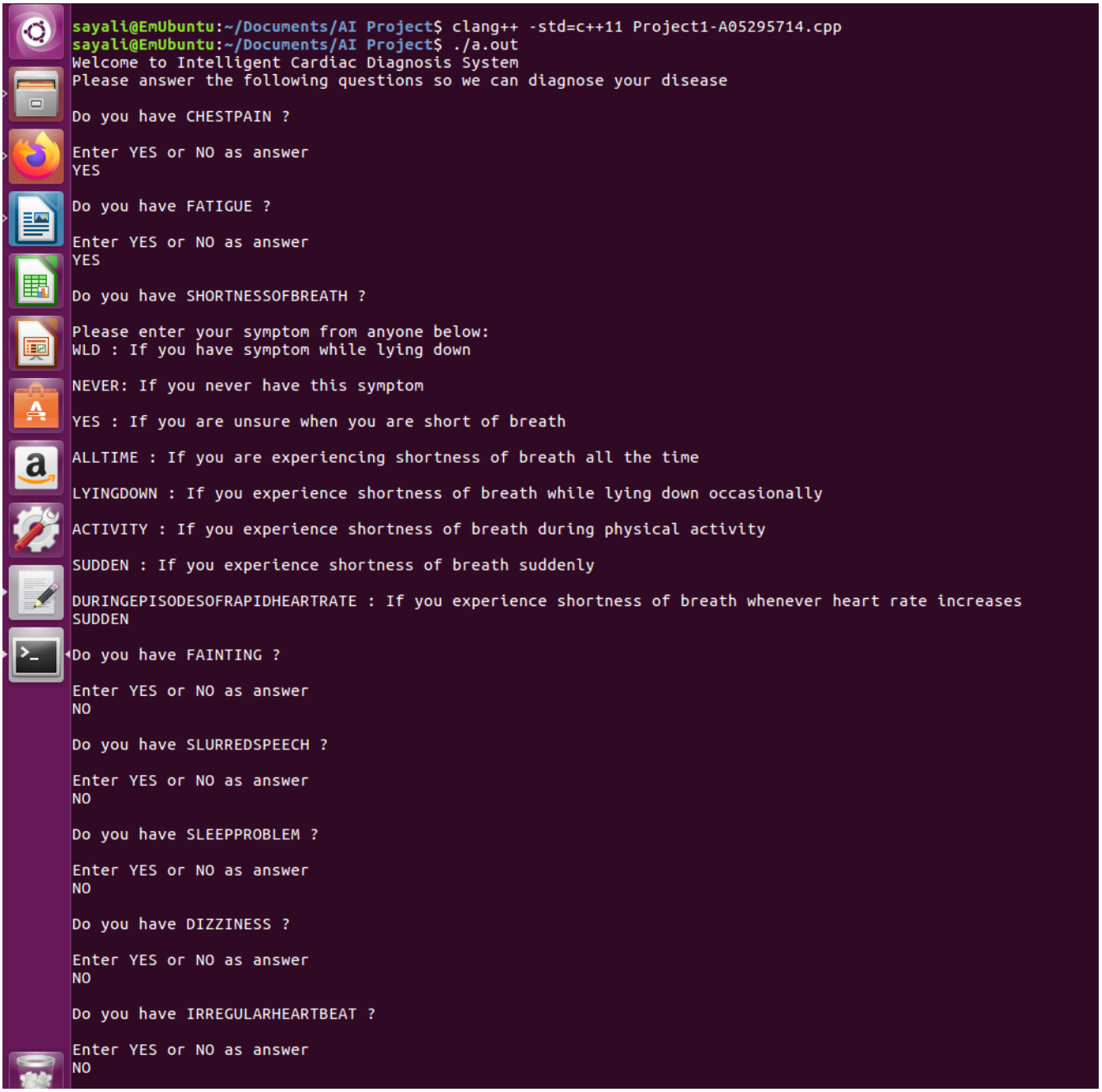
Patient 4 – WOLFF-PARKINSON-WHITE

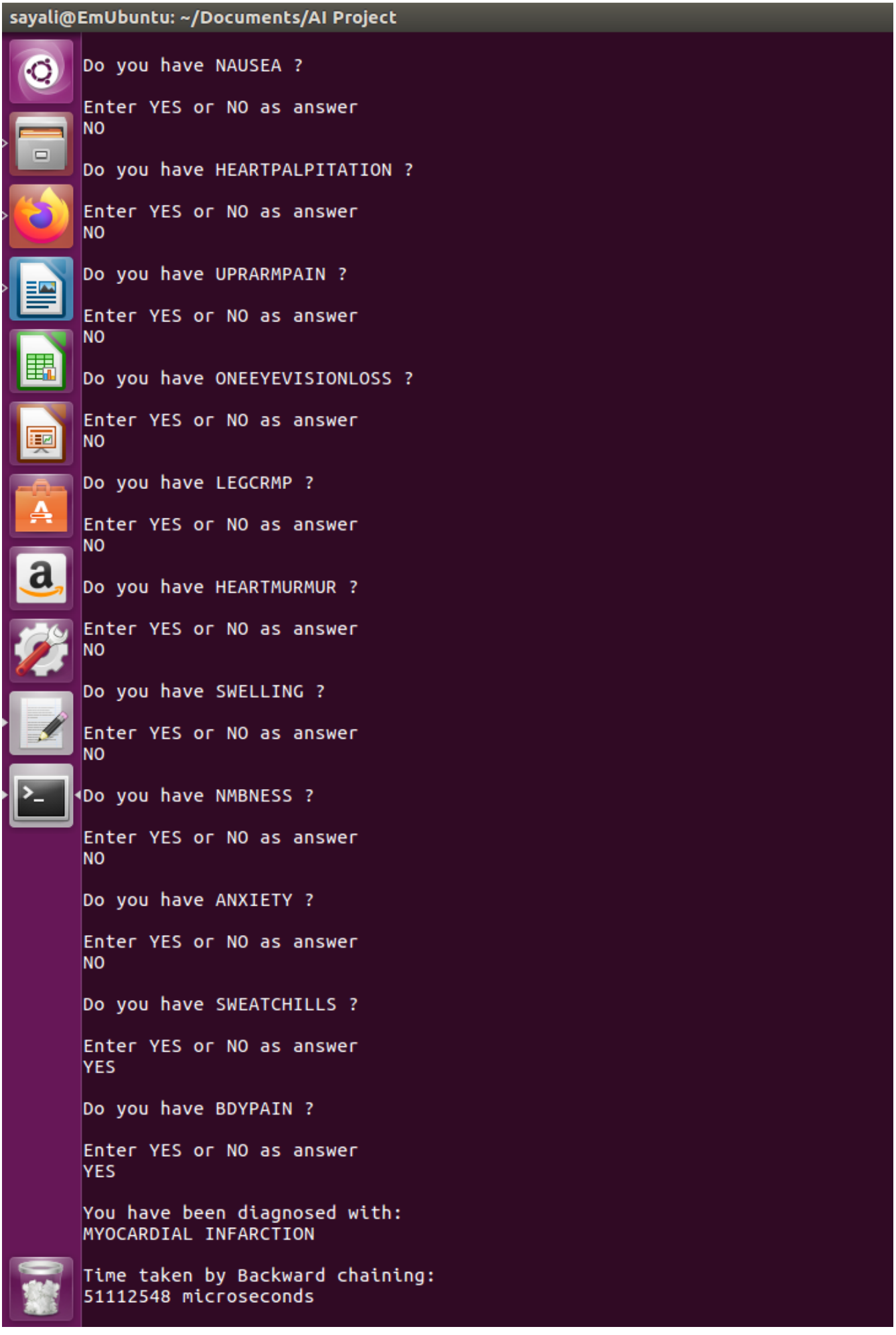



sayali@EmUbuntu: ~/Documents/AI Project
Do you have NAUSEA ?
Enter YES or NO as answer
NO
Do you have HEARTPALPITATION ?
Enter YES or NO as answer
YES
Do you have UPRARMPAIN ?
Enter YES or NO as answer
NO
Do you have ONEEYEVISIONLOSS ?
Enter YES or NO as answer
NO
Do you have LEGCRMP ?
Enter YES or NO as answer
NO
Do you have HEARTMURMUR ?
Enter YES or NO as answer
NO
Do you have SWELLING ?
Enter YES or NO as answer
NO
Do you have NMBNESS ?
Enter YES or NO as answer
NO
Do you have ANXIETY ?
Enter YES or NO as answer
YES
You have been diagnosed with:
WOLFF-PARKINSON-WHITE SYNDROME
Time taken by Backward chaining:
73283836 microseconds
Please answer few more questions below to get proper treatment:
have symptoms often?
YES
Treatment we recommend is:
Catherer ablation
cardioversion procedure
adinosine



Patient 5 - MYOCARDIAL INFARCTION




sayali@EmUbuntu: ~/Documents/AI Project
Do you have NAUSEA ?
Enter YES or NO as answer
NO
Do you have HEARTPALPITATION ?
Enter YES or NO as answer
NO
Do you have UPRARMPAIN ?
Enter YES or NO as answer
NO
Do you have ONEEYEVISIONLOSS ?
Enter YES or NO as answer
NO
Do you have LEGCRMP ?
Enter YES or NO as answer
NO
Do you have HEARTMURMUR ?
Enter YES or NO as answer
NO
Do you have SWELLING ?
Enter YES or NO as answer
NO
Do you have NMBNESS ?
Enter YES or NO as answer
NO
Do you have ANXIETY ?
Enter YES or NO as answer
NO
Do you have SWEATCHILLS ?
Enter YES or NO as answer
YES
Do you have BDYPAIN ?
Enter YES or NO as answer
YES
You have been diagnosed with:
MYOCARDIAL INFARCTION
Time taken by Backward chaining:
51112548 microseconds

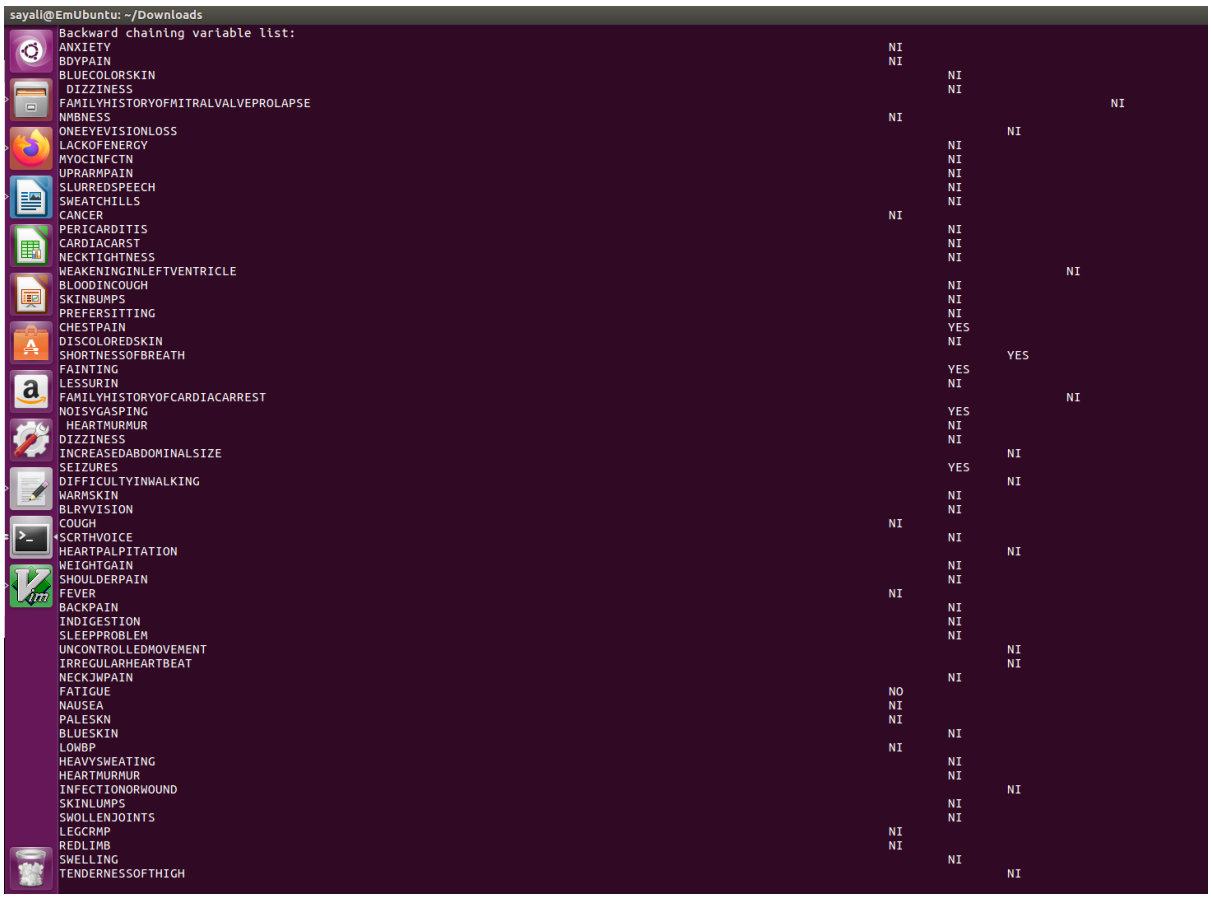
```
Please answer few more questions below to get proper treatment:
Little or no blood flow in some areas
YES
increased clot formation in blood
YES

Treatment we recommend is:

Supplementary oxygen
Anti-clotting medications
Nitroglycerin
Thrombolytic (clot-busting) medications
Anti-arrhythmia medications
Pain medications

Time taken by Forward chaining:
49444177 microsecondssayali@EmUbuntu:~/Documents/AI Project$
```

Note : To take the screenshots for 5 patient diagnoses, The backward chaining rules in the program have been shuffled (temporarily) . Otherwise, the number of screenshots and the length of the report would become too lengthy.

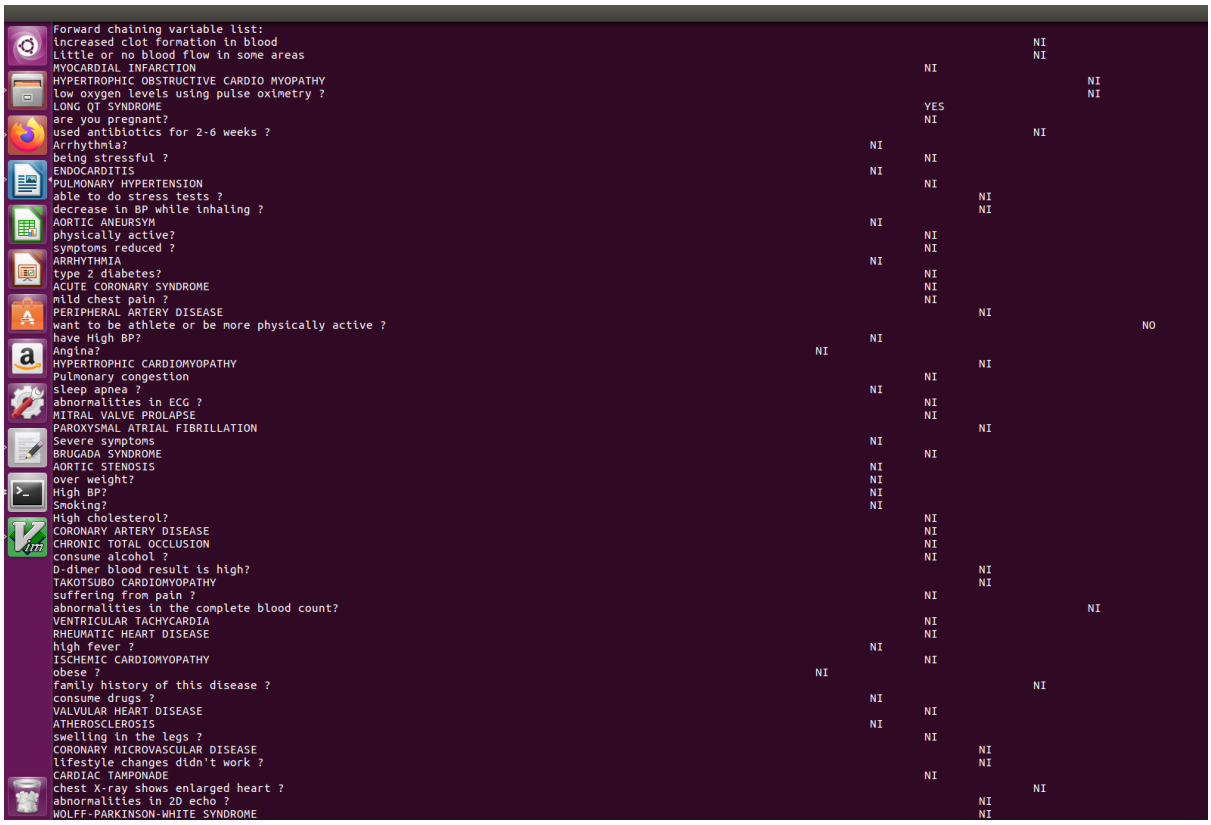
Printing the intermediate results to trace the program.

Consider diagnosis for 'LONG QT SYNDROME' –

Variable list Initialization for backward chaining-



Variable list Initialization for forward chaining-



9. ANALYSIS OF THE PROGRAM

1. The 'SHORTNESSOFBREATH' variable has 8 different types of answers. We've utilized a categorical variable to simplify answering questions related to shortness of breath, making it easier for users.
2. We've employed the map data structure for the variable list, derived global variable list, and clause variable list. The use of maps has eliminated the need for additional for loops, reducing complexity during iteration.

10. ANALYSIS OF THE RESULTS

10.1 Execution time

As we are using the map data structure for the variable list, derived global variable list, and clause variable list, we may have improved the efficiency of the program.

Regarding the execution times, when the decision tree branch is smaller or shallower, the backward chaining algorithm approximately takes 6-10 seconds, while the forward chaining algorithm takes approximately 19 seconds. However, if the decision tree branch is deeper or longer, the backward chaining algorithm may take approximately 47 seconds, while the forward chaining algorithm still takes approximately 19 seconds.

It is important to note that the forward chaining algorithm seems to take a consistent amount of time regardless of the decision tree branch's depth or length, while the backward chaining algorithm's execution time varies based on the complexity of the branch.

10.2 Memory usage

```cpp
#include <iostream>
#include <fstream>
#include <unistd.h>
void process_mem_usage(double& vm_usage, double& resident_set)
{
    vm_usage     = 0.0;
    resident_set = 0.0;

    unsigned long vsize;
    long rss;
    {
        std::string ignore;
        std::ifstream ifs("/proc/self/stat", std::ios_base::in);
        ifs >> ignore >> ignore >> ignore >> ignore >> ignore >> ignore >> ignore >> ignore >> ignore >> ignore
                >> ignore >> ignore >> ignore >> ignore >> ignore >> ignore >> ignore >> ignore >> ignore >> ignore
                >> ignore >> ignore >> vsize >> rss;
    }

    long page_size_kb = sysconf(_SC_PAGE_SIZE) / 1024;
    vm_usage = vsize / 1024.0;
    resident_set = rss * page_size_kb;
}
int main()
{
   using std::cout;
   using std::endl;

   double vm, rss;
   process_mem_usage(vm, rss);
   cout << "Virtual Memory Size: " << vm << "; Resident Set Size: " << rss << endl;
}
```

By executing this snippet code provided above, our expert system program approximately uses Virtual Memory: 6472 KB and Resident Set: 3364 KB.

11. CONCLUSION

We have developed an intelligent expert system for diagnosing Cardiovascular (Heart) diseases and recommending treatment for hospitals. The process involved various steps including requirement gathering, decision tree development, rule conversion, and algorithm implementation.

We diagnosed 30 different heart diseases and recommended treatment for each. Information on symptoms and treatment was gathered from reputable healthcare websites. We used the Backward chaining algorithm for diagnosing diseases and the Forward chaining algorithm for treatment recommendation.

After experimenting with around 15 patients, our intelligent expert system has proven to provide accurate results and is efficient. It is also user-friendly, requiring patients to simply answer questions asked by the system. The system then diagnoses the disease and recommends appropriate treatment. Given the complexity of diagnosing heart diseases due to the variety of symptoms, this expert system is extremely useful in healthcare.

By using advanced technology, we have contributed to solving real-world and critical problems, making it easier for healthcare professionals to obtain accurate results. This is important for giving patients the best care and improving healthcare.

12. REFERENCES

[1] https://www.nhlbi.nih.gov/health/long-qt-syndrome

[2] https://www.pennmedicine.org/for-patients-and-visitors/patient-information/conditions-treated-a-to-z/pulmonary-hypertension

[3] https://my.clevelandclinic.org/health/diseases/17145-ischemic-cardiomyopathy

[4] https://my.clevelandclinic.org/health/diseases/16813-brugada-syndrome

[5] https://my.clevelandclinic.org/health/diseases/22910-acute-coronary-syndrome

[6] https://my.clevelandclinic.org/health/diseases/17857-broken-heart-syndrome
[7] https://www.ncbi.nlm.nih.gov/books/NBK554437/
[8] https://my.clevelandclinic.org/health/diseases/17567-cad-total-coronary-occlusions

[9] https://my.clevelandclinic.org/health/diseases/21052-microvascular-coronary-disease

[10] https://www.cedars-sinai.org/health-library/diseases-and-conditions/v/ventricular-tachycardia-1.html

[11] https://www.baeldung.com/linux/resident-set-vs-virtual-memory-size

13. TEAM MEMBER'S CONTRIBUTION

Sayali Pathak

1. Gathered information(including symptoms and treatments) for 10 cardiovascular diseases.
2. Built 10 branches of the trees required for the program.
3. Constructed 10 rules corresponding to 10 diseases.
4. Actively participated and contributed to the meetings for integration of rules and trees.
5. Implemented 4 functions in the code and worked together with the group for integration.
6. Worked to debug code identified and resolved issues. And contributed towards analysing program efficiency.

Gopika Mahadevan

1. Gathered information(including symptoms and treatments) for 10 cardiovascular diseases.
2. Built 10 branches of the trees required for the program.
3. Constructed 10 rules corresponding to 10 diseases.
4. Actively participated and contributed to the meetings for integration of rules and trees.
5. Implemented 4 functions in the code and worked together with the group for integration.
6. Worked to debug code identified and resolved issues. And contributed towards analysing program efficiency.

Sri Sudha Kambhampati

1. Gathered information(including symptoms and treatments) for 10 cardiovascular diseases.
2. Built 10 branches of the trees required for the program.
3. Constructed 10 rules corresponding to 10 diseases.
4. Actively participated and contributed to the meetings for integration of rules and trees.
5. Implemented 4 functions in the code and worked together with the group for integration.
6. Worked to debug code identified and resolved issues. And contributed towards analysing program efficiency.