BABU BANARASI DAS
INSTITUTE OF TECHNOLOGY & MANAGEMENT
Affiliated to Dr A P J Abdul Kalam Technical University (AKTU College Code- 054)
Approved by All India Council for Technical Education (AICTE)
Website: **www.bbditm.ac.in** / e-mail: **director@bbdnitm.ac.in**
Phone Number: +91 – (522) – 6196222 / 6196223 / 6196305 (VPN No. 723)

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

LAB MANUAL

OF

ARTIFICIAL INTELLIGENCE

[BCS-751]

B.TECH, 4th Year, Semester-VIIth

Dr. A. P. J. Abdul Kalam Technical University,

Uttar Pradesh

2025-26

Department of Information Technology

Faculty Name: Dr. Shailja Pandey
(Associate Professor)

Approved By:
(Head Of Department)

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

MANUAL CONTENTS

This manual is intended for the 4th year students of INFORMATION TECHNOLOGY in the subject of ARTIFICIAL INTELLIGENCE LAB. This manual typically contains practical/lab sessions related to ARTIFICIAL INTELLIGENCE covering various aspects related to the subject to enhanced understanding.

Students are advised to thoroughly go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

PREFACE

This practical manual will be helpful for students of Information Technology for understanding the course from the point of view of applied aspects. Though all the efforts have been made to make this manual error free, some errors might have crept in inadvertently. Suggestions from the readers for the improvement of the manual are most welcome.

Vision of the Institute

"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."

Mission of the Institute

M1.To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence.

M2. To develop technocrats with creative skills and leadership qualities, to solve local and global challenges.

M3. To impart human values and ethics in students, to make them socially and eco-friendly responsible.

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Vision of the Department

"Vision of the department to transform the students to be skilled IT professionals, Innovative leader and environmentally receptive citizens."

Mission of the Department

M1. To develop and deliver quality academic programs in emerging & innovative fields of engineering to empower the students to meet industry standards

M2. To provide students with the ethical and professional tools to become productive & become continuous Learner

M3. To inculcate professional behavior, positive attitude, and communication skills

Program Educational Objectives (PEOs)

An engineering graduate of the Department of Information Technology Program will be able to demonstrate:

PEO I: To develop the ability among students to understand the concept of core Information Technology subjects that will facilitate understanding of new technology.

PEO II: To embed a strong foundation in engineering fundamentals to solve, analyze and design real time engineering products

PEO III: To give exposures to emerging edge technologies, adequate training and opportunities to work as team on multidisciplinary projects with effective communication skills and leadership qualities.

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Program Out Comes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Program Specific Outcomes (PSOs)

An engineering graduate of the Department of Information Technology Program will be able to demonstrate:

PSO1: Use and apply current technical concepts and practices in the core Information Technologies of human Machine Learning, Information Management, Programming, Cloud Computing.

PSO2: Design, develop and test software systems for world-wide network of computers to provide solutions to real world problems

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Course Evaluation Scheme

Sr No	Subject Code	Subject Name	Periods			Evauation Scheme				Total	Credit
			L	T	P	Sessional Assessment			PE		
						CT	TA	PS			
1.	BCS751	Artificial Intelligence Lab	0	0	2			50	50	100	1

Course Objectives:

The teacher will explain:

1.	The Artificial Intelligence (AI) Lab is designed to provide hands-on experience with AI techniques, algorithms, and applications. The lab aims to bridge theoretical knowledge with practical implementations.

Pre- requisite:

1	Mathematics & Stats.
2	Programming Knowledge
3	Machine Learning & Deep Learning
4	NLP Concepts.
5	Natural Language Toolkit (NLTK)

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Course Outcomes (COs)

Course Outcomes: The students should be able to:		**Bloom's Level**
CO1	Understand and apply basic search algorithms and intelligent problem-solving approaches in Python.	K2, K3
CO2	Design and implement logic-based knowledge representations and reasoning using Python and Prolog.	K3, K4
CO3	Use Natural Language Toolkit (NLTK) for basic NLP tasks like stemming, tagging, and classification.	K3, K4
CO4	Analyze and implement basic AI game strategies like Minimax and Alpha-Beta pruning.	K4, K5
CO5	Demonstrate AI techniques in real-world tasks such as text processing, planning, or constraint satisfaction.	K4, K5

CO-PO-PSO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	-	1	1	1	3	1	-	-	2	-	-	1	3	3
CO2	-	1	2	2	2	-	-	-	1	-	1	1	3	3
CO3	-	1	2	2	2	-	-	-	1	-	1	1	3	3
CO4	-	1	2	2	2	-	-	-	1	-	1	1	3	3
CO5	-	1	2	2	2	-	-	-	1	-	1	1	3	3

The extent of mapping is as follows: 1 for low, 2 for moderate, 3 for high &"-" for No correlation between CO & PO.

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Department of Information Technology

Computer Lab DO's and DON'T

Do's

1. Know the location of the fire extinguisher and the first aid box how to use them in case of an emergency.
2. Read and understand how to carry out an activity thoroughly before coming to the laboratory.
3. Report files or accidents to your lecturer/laboratory technician immediately.
4. Report any broken plugs or exposed electrical wires to your lecturer/laboratory technician immediately.

Don'ts

1. Do not eat or drink in the laboratory.
2. Avoid stepping on electrical wires or any other computer cables
3. Do not open the system unit casing or monitor casing particularly when the power is turned on. Some internal components hold electric voltages of up to 240 volts, which can be fatal.
4. Do not insert metal objects such as clips, pins, and needles into the computer casings. They may cause fire.
5. Do not remove anything from computer laboratory without permission.
6. Do not touch, connect, or disconnect any plug or cable without your lecturer/laboratory technician's permission.

Safety Measures

1. Specific Safety Rules like Do's and Don'ts are display and instructed for all students.
2. First aid box and fire extinguishers are kept in each laboratory.
3. Proper earthing of all electrical equipment's.
4. CCTV installed.
5. Well trained technical supporting staff monitors the labs at all times.
6. Damaged system/equipment's are identified and serviced at the earliest on regular basis.
7. Software's are updated regularly.

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Department of Information Technology

INDEX

S. No	Experiment Name	CO	BL
1	Implement Depth First Search (DFS) for a tree or graph structure.	CO1	L3
2	Solve the 8 Puzzle Problem using A* Search Algorithm.	CO1	L3
3	Implement Hill Climbing Algorithm for numerical optimization or path finding.	CO1	L3
4	Solve Water Jug Problem using state-space search (BFS or DFS).	CO1	L3
5	Write Prolog programs to define family relationships using predicates.	CO2	L3
6	Implement 4-Queens Problem in Prolog using backtracking	CO2	L4
7	Implement Unification Algorithm in Python or Prolog.	CO2	L4
8	Implement Forward and Backward Chaining in a rule-based system (manual or code-based).	CO2	L4
9	Demonstrate Resolution in Propositional Logic through a basic example (e.g.; proving a theorem).	CO2	L4
10	Implement Tic-Tac-Toe game with a basic AI opponent.	CO4	L4
11	Implement Min-Max (Minimax) Algorithm for decision making in turn-based games.	CO4	L4
12	Enhance the game with Alpha-Beta Pruning to optimize Min-Max.	CO4	L5
13	Build a simple chatbot using rules or pre-trained logic (can use regex or basic intent matching).	CO5	L5
14	Design a Constraint Satisfaction Problem solver, e.g., Sudoku, or Map Coloring.	CO5	L5
15	Perform simple Bayesian reasoning for a probability-based decision problem (e.g., medical diagnosis).	CO5	L4

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

BEYOND SYLLABUS

16	Build a Sentiment Analysis Classifier using Machine Learning (Naïve Bayes/Logistic Regression)	CO3	L6
17	Implement Genetic Algorithm (GA) to solve an optimization problem (eg; TSP or String Matching).	CO5	L6

Software & Hardware Required

1. Windows 10/11 (64-bit), Linux (Ubuntu 20.04+ preferred for better ML support).
2. SWI-Prolog, Python (3.8+): Main language for NLP, Jupiter Notebook: Interactive coding environment, Anaconda: Python package manager (optional but useful), VS Code / PyCharm: Preferred IDEs for coding
3. PostgreSQL / MySQL – For structured text storage, MongoDB / Elasticsearch – For unstructured text and search applications.

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENT NO–1

OBJECTIVE: Implement Depth First Search (DFS) for a tree or graph structure

DFS Implementation

```
% dfs(Start, VisitedNodes)
dfs(Start) :-
dfs(Start, []).

dfs(Node, Visited) :-
   \+ member(Node, Visited),
write(Node), write(' '),
findall(Next, edge(Node, Next), Neighbors),
dfs_list(Neighbors, [Node | Visited]).

dfs_list([], _).
dfs_list([H|T], Visited) :-
dfs(H, Visited),
dfs_list(T, Visited).
```

Graph Representation

```
% Graph edges
edge(a, b).
edge(a, c).
edge(b, d).
edge(b, e).
edge(c, f).
```

Run Query

```
?-dfs(a).
```

Output

a b d e c f

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENT NO–2

OBJECTIVE:Solve the 8 Puzzle Problem using A* Search Algorithm.

A* Search Framework

Each node is represented as:
node(State, Parent, G, H, F)
Where:

- G = cost from start
- H = heuristic (Manhattan distance)
- F = G + H

SWI-Prolog Program (A* for 8-Puzzle)

```
/* ============================
   8-Puzzle using A* Search
   ============================ */

:-use_module(library(lists)).

/* ---------- Goal State ---------- */
goal([1,2,3,4,5,6,7,8,0]).

/* ---------- Start State ---------- */
start([1,2,3,4,0,6,7,5,8]).

/* ---------- Entry Point ---------- */
solve :-
start(Start),
heuristic(Start, H),
astar([node(Start, nil, 0, H, H)], [], Solution),
print_solution(Solution).

/* ---------- A* Algorithm ---------- */
astar(Open, _, node(State, Parent, G, H, F)) :-
   Open = [node(State, Parent, G, H, F)|_],
goal(State).
astar(Open, Closed, Solution) :-
   Open = [Current|RestOpen],
   Current = node(State, _, G, _, _),
findall(Child,
expand(Current, Closed, Child),
        Children),
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
append(RestOpen, Children, NewOpen),
sort_open(NewOpen, SortedOpen),
astar(SortedOpen, [State|Closed], Solution).

/* ---------- Expand Nodes ---------- */
expand(node(State, _, G, _, _), Closed,
node(NewState, State, NewG, H, F)) :-
move(State, NewState),
   \+ member(NewState, Closed),
NewG is G + 1,


heuristic(NewState, H),
   F is NewG + H.

/* ---------- Moves ---------- */
move(State, NewState) :-
   nth0(Blank, State, 0),
swap(Blank, State, NewState).

swap(Blank, State, NewState) :-
neighbor(Blank, N),
   nth0(N, State, Tile),
replace(State, Blank, Tile, Temp),
replace(Temp, N, 0, NewState).

/* ---------- Valid Blank Neighbors ---------- */
neighbor(I, J) :-
   Row is I // 3, Col is I mod 3,
( J is I - 3, Row > 0
   ; J is I + 3, Row < 2
   ; J is I - 1, Col > 0
   ; J is I + 1, Col <2 ).

/* ---------- Replace Element ---------- */
replace(List, Index, Elem, NewList) :-
same_length(List, NewList),
append(Prefix, [_|Suffix], List),
length(Prefix, Index),
append(Prefix, [Elem|Suffix], NewList).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
/* ---------- Heuristic: Manhattan Distance ---------- */
heuristic(State, H) :-
goal(Goal),
manhattan(State, Goal, 0, H).

manhattan([], [], H, H).
manhattan([0|T1], [_|T2], Acc, H) :-
manhattan(T1, T2, Acc, H).
manhattan([X|T1], Goal, Acc, H) :-
   nth0(GI, Goal, X),
   nth0(CI, [X|T1], X),
rowcol(GI, R1, C1),
rowcol(CI, R2, C2),
   D is abs(R1-R2) + abs(C1-C2),
   Acc1 is Acc + D,
manhattan(T1, Goal, Acc1, H).

rowcol(I, R, C) :-
   R is I // 3,
   C is I mod 3.
/* ---------- Sort Open List by F ---------- */
sort_open(Open, Sorted) :-
predsort(compare_f, Open, Sorted).

compare_f(<, node(_,_,_,F1), node(_,_,_,F2)) :- F1 < F2.
compare_f(>, node(_,_,_,F1), node(_,_,_,F2)) :- F1 >= F2.

/* ---------- Print Solution Path ---------- */
print_solution(Node) :-
build_path(Node, Path),
reverse(Path, Solution),
print_steps(Solution, 0).

build_path(node(State, nil, _, _, _), [State]).
build_path(node(State, Parent, _, _, _), [State|Rest]) :-
   Parent \= nil,
build_path(node(Parent, nil, _, _, _), Rest).
print_steps([], _).
print_steps([State|T], N) :-
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
format('Step ~w: ~n', [N]),
print_state(State),
   N1 is N + 1,
print_steps(T, N1).

print_state([A,B,C,D,E,F,G,H,I]) :-
format('~w ~w ~w~n', [A,B,C]),
format('~w ~w ~w~n', [D,E,F]),
format('~w ~w ~w~n~n', [G,H,I]).
```

4. How to Run in SWI-Prolog

1. Save the file as eight_puzzle.pl
2. Open SWI-Prolog
3. Load the program:

```
?- [eight_puzzle].
```

4. Solve the puzzle:

```
?- solve.
```

5. Output (Example)

```
Step 0:
1 2 3
4 0 6
7 5 8

Step 1:
1 2 3
4 5 6
7 0 8

Step 2:
1 2 3
4 5 6
7 8 0
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENTNO–3

OBJECTIVE: Implement Hill Climbing Algorithm for numerical optimization or path finding.

Hill Climbing for Numerical Optimization (Maximization)

Problem

Maximize a function

$$f(X) = -X^2 + 10$$

Idea

- Start from an initial value
- Move to a neighbor with **higher value**
- Stop when no better neighbor exists

Prolog Program

```
% Objective function
value(X, V) :-
   V is -X*X + 10.

% Generate neighbors
neighbor(X, X1) :-
   X1 is X + 1.
neighbor(X, X1) :-
   X1 is X - 1.

% Hill Climbing algorithm
hill_climb(X, X) :-
value(X, V),
   \+ ( neighbor(X, N),
value(N, VN),
        VN >V ).

hill_climb(X, Result) :-
neighbor(X, N),
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
value(X, V),
value(N, VN),
 VN > V,
hill_climb(N, Result).
```

Query

```
?-hill_climb(5, Result).
```

Output

Result = 0(0 is the maximum of the function)

Hill Climbing for Path Finding (Using Heuristic)

Problem

Find a path from **a** to **goal** using heuristic values.

Graph Definition

```
edge(a, b).
edge(a, c).
edge(b, d).
edge(c, e).
edge(d, goal).
edge(e, goal).
```

Heuristic Values (Lower is Better)

```
h(a, 5).
h(b, 4).
h(c, 3).
h(d, 2).
h(e, 1).
h(goal, 0).
```

Hill Climbing Algorithm

```
hill_climb(Node, Node) :-
   Node == goal.
hill_climb(Node, Path) :-
findall(H-N,
( edge(Node, N),
h(N, H) ),
Neighbors),
sort(Neighbors, [_-Best|_]),
hill_climb(Best, Path).
```

Query

```
?-hill_climb(a, Result).
```

Output

Result = goal

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENTNO–4

OBJECTIVE: Solve Water Jug Problem using state-space search (BFS or DFS).

Problem Statement

- Jug A capacity = **4 liters**
- Jug B capacity = **3 liters**
- Initial state: (0,0)
- Goal state: **exactly 2 liters in Jug A**

State Representation

state(A, B)
A → water in Jug A
B → water in Jug B

Allowed Operations

1. Fill Jug A
2. Fill Jug B
3. Empty Jug A
4. Empty Jug B
5. Pour A → B
6. Pour B → A

SWI-Prolog Code (BFS)

```
% ----------------------------
% Water Jug Problem using BFS
% ----------------------------

% Capacity of jugs
capacity(4, 3).

% Initial and Goal state
initial(state(0,0)).
goal(state(2,_)).

% ----------------------------
% State Transitions
% ----------------------------
% Fill Jug A
move(state(_,B), state(4,B)).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
% Fill Jug B
move(state(A,_), state(A,3)).

% Empty Jug A
move(state(_,B), state(0,B)).

% Empty Jug B
move(state(A,_), state(A,0)).
% Pour A -> B
move(state(A,B), state(A2,B2)) :-
   capacity(_,CapB),
   T is min(A, CapB - B),
   T > 0,
   A2 is A - T,
   B2 is B + T.

% Pour B -> A
move(state(A,B), state(A2,B2)) :-
capacity(CapA,_),
   T is min(B, CapA - A),
   T > 0,
   A2 is A + T,
   B2 is B - T.

% ------------------------
% Breadth First Search
% ------------------------

bfs([[State|Path]|_], [State|Path]) :-
goal(State).

bfs([[State|Path]|Rest], Solution) :-
findall(
      [Next,State|Path],
( move(State,Next),
       \+ member(Next,[State|Path])
      ),
NewPaths
   ),
append(Rest, NewPaths, Queue),
bfs(Queue, Solution).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
% ------------------------
% Solve Predicate
% ------------------------

solve :-
initial(State),
bfs([[State]], Solution),
reverse(Solution, Path),
write('Solution Path:'), nl,
print_path(Path).

% Print path
print_path([]).
print_path([state(A,B)|T]) :-
format('Jug A: ~w  Jug B: ~w~n',[A,B]),
print_path(T).
```

How to Run (SWI-Prolog)

1. Save file as:
water_jug.pl
2. Load and run:
?- [water_jug].
?- solve.

Output

```
Solution Path:
Jug A: 0  Jug B: 0
Jug A: 4  Jug B: 0
Jug A: 1  Jug B: 3
Jug A: 1  Jug B: 0
Jug A: 0  Jug B: 1
Jug A: 4  Jug B: 1
Jug A: 2  Jug B: 3
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENT NO–5

OBJECTIVE:Write Prolog programs to define family relationships using predicates.

Family Relationships in SWI-Prolog

1. Facts (Basic Information)

```
% Gender
male(john).
male(paul).
male(mike).
male(tom).

female(mary).
female(linda).
female(susan).
female(anna).

% Parent relationships
parent(john, paul).
parent(mary, paul).

parent(john, linda).
parent(mary, linda).

parent(paul, mike).
parent(susan, mike).

parent(paul, anna).
parent(susan, anna).

parent(tom, susan).
S
```

2. Rules (Derived Relationships)

```
% Father
father(X, Y) :-
parent(X, Y),
   male(X).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
% Mother
mother(X, Y) :-
parent(X, Y),
   female(X).

% Child
child(X, Y) :-
parent(Y, X).

% Sibling
sibling(X, Y) :-
parent(Z, X),
parent(Z, Y),

   X \= Y.

% Brother
brother(X, Y) :-
sibling(X, Y),
   male(X).

% Sister
sister(X, Y) :-
sibling(X, Y),
   female(X).

% Grandparent
grandparent(X, Y) :-
parent(X, Z),
parent(Z, Y).

% Grandfather
grandfather(X, Y) :-
grandparent(X, Y),
   male(X).

% Grandmother
grandmother(X, Y) :-
grandparent(X, Y),
   female(X).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
% Ancestor (recursive rule)
ancestor(X, Y) :-
parent(X, Y).

ancestor(X, Y) :-
parent(X, Z),
ancestor(Z, Y).
```

3. Example Queries in SWI-Prolog

```
?-father(john, paul).
?-mother(mary, linda).
?-sibling(paul, linda).
?-brother(paul, linda).
?-grandparent(john, mike).
?-ancestor(john, anna).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENT NO–6

OBJECTIVE: Implement 4-Queens Problem in Prolog using backtracking

Problem Approach

- Represent a solution as a list **[Q1, Q2, Q3, Q4]**
- Index = **column number**
- Value = **row number**
- Constraints:
  - No two queens share the same row
  - No two queens share the same diagonal

Prolog Code (Backtracking Solution)

```
% Solve the 4-Queens problem
queens4(Solution) :-
   Solution = [Q1, Q2, Q3, Q4],
permutation([1,2,3,4], Solution),
safe(Solution).

% Check that all queens are safe
safe([]).
safe([Q|Others]) :-
safe(Others),
no_attack(Q, Others, 1).

% Ensure Q does not attack any other queen
no_attack(_, [], _).
no_attack(Q, [Q1|Others], Distance) :-
   Q =\= Q1,                    % not same row
abs(Q - Q1) =\= Distance,       % not same diagonal
NextDistance is Distance + 1,
no_attack(Q, Others, NextDistance).
```

How It Works

- permutation/2 generates all possible row arrangements (backtracking)
- safe/1 recursively checks diagonal conflicts
- Distance tracks column difference for diagonal detection
- Prolog automatically backtracks to find all valid solutions

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Example Query
?- queens4(S).

Output:
S = [2, 4, 1, 3] ;
S = [3, 1, 4, 2] ;
false.

EXPERIMENT NO–7

OBJECTIVE: Implement Unification Algorithm in Python or Prolog.

A clean Python version using recursive unification:

```
from collections import deque

#Helper functions
def is_variable(x):
    return isinstance(x, str) and x[0].islower()

def is_compound(x):
    return isinstance(x, tuple)

def occurs_check(var, term, subst):
    """Check for circular references"""
    if var == term:
        return True
elifis_variable(term) and term in subst:
        return occurs_check(var, subst[term], subst)
elifis_compound(term):
        return any(occurs_check(var, t, subst) for t in term[1:])
    else:
        return False

def unify(x, y, subst=None):
    """Unify terms x and y with existing substitution subst"""
    if subst is None:
subst = {}
    stack = deque([(x, y)])
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
    while stack:
        (a, b) = stack.pop()

        if a == b:
            continue
elifis_variable(a):
            if occurs_check(a, b, subst):
                return None
subst[a] = b
elifis_variable(b):
            if occurs_check(b, a, subst):
                return None
subst[b] = a
elifis_compound(a) and is_compound(b):
            if a[0] != b[0] or len(a) != len(b):
                return None
stack.extend(zip(a[1:], b[1:]))
        else:
            return None
    return subst
```

Example usage

```
if __name__ == "__main__":
    term1 = ('f', 'X', 'a')
    term2 = ('f', 'b', 'Y')
    result = unify(term1, term2)
print("Unification result:", result)
```

How this works:

1. Variables are strings starting with lowercase letters (x, y, z).
2. Compound terms are tuples: ('f', arg1, arg2).
3. Occurs check prevents infinite substitutions (e.g., X = f(X)).
4. The algorithm uses a stack instead of recursion for efficiency.
5. Returns a substitution dictionary if unification succeeds, or None if it fails.

Example Output:

```
Unification result: {'X': 'b', 'Y': 'a'}
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENT NO–8

OBJECTIVE: Implement Forward and Backward chaining in a rule-based system (manual or code-based).

1. Forward Chaining in Prolog

Forward chaining is **data-driven**: it starts with facts and applies rules to infer new facts until a goal is reached.
Here's a simple example:

```
% Facts
fact(sunny).
fact(weekend).

% Rules
rule(go_out, [sunny, weekend]).
rule(stay_home, [rainy]).

% Forward Chaining Algorithm
forward_chain :-
findall(F, fact(F), Facts),
apply_rules(Facts).

apply_rules(Facts) :-
rule(Conclusion, Conditions),
subset(Conditions, Facts),
   \+ member(Conclusion, Facts),
write('Inferred: '), writeln(Conclusion),
   assert(fact(Conclusion)),
forward_chain.  % Continue chaining
apply_rules(_).

% Helper: check if all elements of a list are in another list
subset([], _).
subset([H|T], L) :-
member(H, L),
subset(T, L).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Explanation:

1. fact/1 stores the known facts.
2. rule/2 stores rules as rule(Conclusion, [Conditions]).
3. forward_chain/0 iterates through facts and rules to infer new facts.
4. subset/2 checks if all conditions of a rule are met.

Example Run:

```
?-forward_chain.
Inferred: go_out
true.
```

2. Backward Chaining in Prolog

Backward chaining is **goal-driven**: it starts with a goal and tries to prove it using rules and facts.

```
% Facts
fact(sunny).
fact(weekend).

% Rules
rule(go_out, [sunny, weekend]).
rule(stay_home, [rainy]).

% Backward Chaining Algorithm
backward_chain(Goal) :-
fact(Goal), !.   % Goal already known
backward_chain(Goal) :-
rule(Goal, Conditions),
prove_all(Conditions).

% Helper: prove all conditions
prove_all([]).
prove_all([H|T]) :-
backward_chain(H),
prove_all(T).
```

Example Run:

```
?-backward_chain(go_out).
true.
?-backward_chain(stay_home).
false.
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Explanation:

1. backward_chain(Goal) tries to see if Goal is a known fact or can be derived from rules.
2. prove_all/1 recursively proves all conditions of a rule.

EXPERIMENT NO–9

OBJECTIVE: Demonstrate Resolution in Propositional Logic through a basic example (e.g.; proving a theorem).

Problem

We want to prove:
If it rains, the ground gets wet. It rains. Therefore, the ground gets wet.

Step 1: Represent in Propositional Logic

Let:

- r = "It rains"
- w = "The ground is wet"

We have:

1. $r \rightarrow w$(If it rains, the ground is wet)
2. r(It rains)
   We want to prove w(The ground is wet).

Step 2: Convert to Clausal Form

Resolution works on clauses in Conjunctive Normal Form (CNF).

1. $r \rightarrow w \equiv \neg r \vee w$
2. r
   Negate the conclusion for proof by contradiction: $\neg w$

So, our clauses are:

- $C1: \neg r \vee w$
- $C2: r$
- $C3: \neg w$(negation of what we want to prove)

Step 3: Apply Resolution

- Resolve $C1(\neg r \vee w)$and $C2(r)$on $r \rightarrow$ we get w
- Resolve $w$and $\neg w \rightarrow$ empty clause $\perp$ (contradiction)

Hence, the theorem is proved.

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Step 4: Implement in Prolog

In Prolog, we can represent clauses as facts and use a simple resolution mechanism:

```
% Knowledge base
clause([not(r), w]).  % r -> w
clause([r]).          % r

% Negation of the theorem we want to prove
clause([not(w)]).

% Resolution rule
resolve([Literal|Rest1], Clause2, Resolvent) :-
select(Lit, Clause2, Clause2Rest),
   Literal = not(Lit) ;not(Literal) = Lit,
append(Rest1, Clause2Rest, Resolvent).


% Check if empty clause can be derived
prove(Clauses) :-
member([], Clauses), !, write('The theorem is proved by resolution.'), nl.


prove(Clauses) :-
select(C1, Clauses, Rest),
member(C2, Rest),
resolve(C1, C2, Resolvent),
   \+ member(Resolvent, Clauses),   % avoid duplicate clauses
   prove([Resolvent|Clauses]).
```

Step 5: Run the Proof

In Prolog:

```
?-findall(C, clause(C), Clauses), prove(Clauses).
```

Output:

The theorem is proved by resolution.

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENT NO–10

OBJECTIVE: Implement Tic- Tac-Toe game with a basic AI opponent.

Tic-Tac-Toe in SWI-Prolog

Start the game

```
start :-
write('Welcome to Tic-Tac-Toe!'), nl,
write('You are X, AI is O.'), nl,
empty_board(Board),
display_board(Board),
play(Board, x).  % Player 'x' starts
```

Initial empty board

```
empty_board([empty, empty, empty,
             empty, empty, empty,
             empty, empty, empty]).
```

Display the board nicely

```
display_board(Board) :-
nl,
display_row(Board, 0),
display_row(Board, 3),
display_row(Board, 6),
nl.
display_row(Board, N) :-
   nth0(N, Board, A), write_cell(A), write(' | '),
   N1 is N+1, nth0(N1, Board, B), write_cell(B), write(' | '),
   N2 is N+2, nth0(N2, Board, C), write_cell(C), nl.
write_cell(empty) :- write(' ').
write_cell(X)     :- write(X).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Main play loop

```
play(Board, Player) :-
   (winner(Board, Winner) ->
format('Game Over! ~w wins!~n', [Winner])
   ; draw(Board) ->
write('Game Over! It is a draw.'), nl
   ; Player = x ->
player_move(Board, NewBoard),
display_board(NewBoard),
play(NewBoard, o)
   ; Player = o ->
ai_move(Board, NewBoard),
write('AI moves:'), nl,
display_board(NewBoard),
play(NewBoard, x)
   ).
```

Check for draw

```
draw(Board) :-
   \+ member(empty, Board).

% Player move
player_move(Board, NewBoard) :-
write('Enter your move (0-8): '),
read(Pos),
   nth0(Pos, Board, empty), !,
replace(Board, Pos, x, NewBoard).
player_move(Board, NewBoard) :-
write('Invalid move, try again.'), nl,
player_move(Board, NewBoard).
```

AI move (basic: first empty spot)

```
ai_move(Board, NewBoard) :-
   nth0(Pos, Board, empty), !,
replace(Board, Pos, o, NewBoard).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Replace Nth element in list

```
replace([_|T], 0, X, [X|T]).
replace([H|T], N, X, [H|R]) :-
   N > 0,
   N1 is N-1,
replace(T, N1, X, R).
```

Winning conditions

```
winner(Board, Player) :-
   (row_win(Board, Player);
col_win(Board, Player);
diag_win(Board, Player)).

row_win(Board, Player) :-
   (nth0(0, Board, Player), nth0(1, Board, Player), nth0(2, Board, Player));
   (nth0(3, Board, Player), nth0(4, Board, Player), nth0(5, Board, Player));
   (nth0(6, Board, Player), nth0(7, Board, Player), nth0(8, Board, Player)).

col_win(Board, Player) :-
   (nth0(0, Board, Player), nth0(3, Board, Player), nth0(6, Board, Player));
   (nth0(1, Board, Player), nth0(4, Board, Player), nth0(7, Board, Player));
   (nth0(2, Board, Player), nth0(5, Board, Player), nth0(8, Board, Player)).

diag_win(Board, Player) :-
   (nth0(0, Board, Player), nth0(4, Board, Player), nth0(8, Board, Player));
   (nth0(2, Board, Player), nth0(4, Board, Player), nth0(6, Board, Player)).
```

How it works:

Start the game: ?- start.
The board is represented as a list of 9 elements.
Player 'x' moves first and chooses a position 0–8.
AI ('o') moves automatically: it simply picks the first available empty spot.
The board updates after each move and checks for win/draw conditions.

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Game continues until there is a winner or a draw.

Output:

yaml

Copy code

?- start.

Welcome to Tic-Tac-Toe!

You are X, AI is O.

```
  |  |
  |  |
  |  |
```

Enter your move (0-8): 0

```
X |  |
  |  |
  |  |
```

AI moves:

```
X | O |
  |  |
  |  |
```

Enter your move (0-8):

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENT NO–11

OBJECTIVE: Implement Min-Max (Min max) Algorithm for decision making in turn-based games.

SWI-Prolog Mini max Implementation

Game State Representation:
State = Number of stones left in the pile

Player identifiers

```
player(max).  % Maximizing player
player(min).  % Minimizing player
```

Moves: remove 1 or 2 stones

```
move(State, NewState) :-
NewState is State - 1,
NewState>= 0.
move(State, NewState) :-
NewState is State - 2,
NewState>= 0.
```

Terminal state check

```
terminal(State) :- State =:= 0.
```

Utility function
Max player wins = +1
Min player wins = -1

```
utility(State, max, 1) :- State =:= 0. % Max wins
utility(State, min, -1) :- State =:= 0. % Min wins
```

Minimax algorithm

```
minimax(State, Player, BestMove, Value) :-
terminal(State), !,
utility(State, Player, Value).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
minimax(State, Player, BestMove, Value) :-
findall(Move, move(State, Move), Moves),
evaluate_moves(Moves, State, Player, MovesWithValues),
select_best(Player, MovesWithValues, BestMove, Value).
```

Evaluate all moves

```
evaluate_moves([], _, _, []).
evaluate_moves([Move|Moves], State, Player, [mv(Move, Value)|Rest]) :-
next_player(Player, NextPlayer),


minimax(Move, NextPlayer, _, Value),
evaluate_moves(Moves, State, Player, Rest).

% Determine next player
next_player(max, min).
next_player(min, max).

% Select best move based on player type
select_best(max, MovesWithValues, BestMove, BestValue) :-
max_member(mv(BestMove, BestValue), MovesWithValues).

select_best(min, MovesWithValues, BestMove, BestValue) :-
min_member(mv(BestMove, BestValue), MovesWithValues).

% max_member helper
max_member(X, [X]).
max_member(mv(Move, Value), [mv(Move1, Value1)|Rest]) :-
max_member(mv(Move2, Value2), Rest),
   (Value >= Value2 -> X = mv(Move, Value) ; X = mv(Move2, Value2)).

% min_member helper
min_member(X, [X]).
min_member(mv(Move, Value), [mv(Move1, Value1)|Rest]) :-
min_member(mv(Move2, Value2), Rest),
   (Value =< Value2 -> X = mv(Move, Value) ; X = mv(Move2, Value2)).
```

Play game interactive

```
play(State, Player) :-
terminal(State), !,
format("Game Over! ~w wins~n", [Player]).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
play(State, Player) :-
minimax(State, Player, BestMove, Value),
format("Player ~w: Best move = ~w with value = ~w~n", [Player, BestMove, Value]),
NewState = BestMove,
next_player(Player, NextPlayer),
play(NewState, NextPlayer).
```

Example Run
Load the program into SWI-Prolog, then query:
?-play(5, max).

Expected Output:
Player max: Best move = 2 with value = 1
Player min: Best move = 1 with value = -1
Player max: Best move = 1 with value = 1
Player min: Best move = 0 with value = -1
Game Over! min wins
This output shows the Minimax algorithm deciding the optimal moves for both players at each turn.

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENT NO–12

OBJECTIVE: Enhance the game with Alpha-Beta Pruning to optimize Min-Max.

1. Basic Idea (Quick Recap)

- **Minimax** explores all moves → slow
- **Alpha-Beta Pruning** cuts off branches that can't affect the final decision
- Result: **same best move, much faster**

2. Game Assumptions

We assume these predicates already exist (or you'll adapt them):

```
moves(Position, Player, Moves).     % legal moves
make_move(Position, Move, NewPos).
terminal(Position).
utility(Position, Value).           % evaluation score
```

3. Alpha-Beta Minimax Implementation (SWI-Prolog)

```
% Entry point
alphabeta(Position, Player, BestMove, BestValue) :-
alphabeta(Position, Player, -1000, 1000, BestMove, BestValue).
```

Terminal node

```
alphabeta(Position, _, _, _, none, Value) :-
terminal(Position),
utility(Position, Value), !.
```

MAX player

```
alphabeta(Position, max, Alpha, Beta, BestMove, BestValue) :-
moves(Position, max, Moves),
max_value(Moves, Position, Alpha, Beta, none, -1000, BestMove, BestValue).
```

MIN player

```
alphabeta(Position, min, Alpha, Beta, BestMove, BestValue) :-
moves(Position, min, Moves),
min_value(Moves, Position, Alpha, Beta, none, 1000, BestMove, BestValue).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
---------- MAX PLAYER ----------
max_value([], _, _, _, BestMove, BestValue, BestMove, BestValue).

max_value([Move|Moves], Position, Alpha, Beta, _, Value,
BestMove, BestValue) :-

make_move(Position, Move, NewPos),
alphabeta(NewPos, min, Alpha, Beta, _, NewValue),

( NewValue> Value ->
      Alpha1 is max(Alpha, NewValue),
      BestMove1 = Move,
      Value1 = NewValue;


  Alpha1 = Alpha,
      BestMove1 = BestMove,
      Value1 = Value
   ),

( Alpha1 >= Beta ->
BestMove = BestMove1,
BestValue = Value1       % PRUNE
   ;
max_value(Moves, Position, Alpha1, Beta,
              BestMove1, Value1, BestMove, BestValue)
   ).

 ---------- MIN PLAYER ----------

min_value([], _, _, _, BestMove, BestValue, BestMove, BestValue).

min_value([Move|Moves], Position, Alpha, Beta, _, Value,
BestMove, BestValue) :-

make_move(Position, Move, NewPos),
alphabeta(NewPos, max, Alpha, Beta, _, NewValue),

( NewValue< Value ->
      Beta1 is min(Beta, NewValue),
      BestMove1 = Move,
      Value1 = NewValue
   ;
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
    Beta1 = Beta,
      BestMove1 = BestMove,
      Value1 = Value
   ),
( Alpha>= Beta1 ->
BestMove = BestMove1,
BestValue = Value1        % PRUNE
   ;
min_value(Moves, Position, Alpha, Beta1,
              BestMove1, Value1, BestMove, BestValue)
   ).
```

4. Simple Example Game (for Testing)

Toy Game Tree

```
terminal(c).
terminal(d).
terminal(e).
terminal(f).
utility(c, 3).
utility(d, 5).
utility(e, 2).
utility(f, 9).
moves(a, max, [b1, b2]).
moves(b, min, [c, d]).
moves(b2, min, [e, f]).
make_move(a, b1, b).
make_move(a, b2, b2).
make_move(b, c, c).
make_move(b, d, d).
make_move(b2, e, e).
make_move(b2, f, f)
```

5. Running It in SWI-Prolog

Query

```
?- alphabeta(a, max, Move, Value).
```

Output

```
Move = b2,
Value = 9.
```

Alpha-Beta **pruned unnecessary branches**

Same result as Minimax, but faster

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENT NO–13

OBJECTIVE: Build a simple chatbot using rules or pre-trained logic (can use regex or basic intent matching).

1. Chatbot Features

- Greets the user
- Responds to basic intents:
  - greetings
  - asking name
  - asking help
  - asking time
  - saying goodbye
- Default fallback response

2. SWI-Prolog Chatbot Code

Save this as **chatbot.pl**

Simple Rule-Based Chatbot

Entry point

```
start :-
write('Chatbot: Hello! Type something and press enter.'), nl,
write('Chatbot: Type "bye" to exit.'), nl,
   chat.

% Main chat loop
chat :-
write('You: '),
read_line_to_string(user_input, Input),
respond(Input, Response),
write('Chatbot: '),
write(Response), nl,
( Input = "bye" ->true ;chat ).
```

Rules / Intent Matching

```
respond(Input, "Hello! How can I help you?") :-
member(Input, ["hi", "hello", "hey"]).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
respond (Input, "I am a simple Prolog chatbot.") :-
member (Input, ["who are you", "what are you"]).

respond(Input, "I can answer simple questions. Try saying hello!") :-
member(Input, ["help", "what can you do"]).
respond(Input, Time Response) :-
Input = "time",
get_time(T),
format_time(string(Time Response),
            "The current time is %H:%M:%S.", T).
respond ("bye", "Goodbye! Have a nice day!").
% Fallback rule
respond (_, "Sorry, I did not understand that.").
```

3. How to Run (SWI-Prolog)

```
?- [chatbot].
?- start.
```

4. Sample Output

```
Chatbot: Hello! Type something and press enter.
Chatbot: Type "bye" to exit.
You: hi
Chatbot: Hello! How can I help you?

You: who are you
Chatbot: I am a simple Prolog chatbot.

You: time
Chatbot: The current time is 14:32:08.

You: bye
Chatbot: Goodbye! Have a nice day!
```

5. Optional Enhancements

If you want, I can help you add:

- Regex-based matching
- Tokenization (split_string/4)
- Knowledge base (facts + questions)
- Menu-driven chatbot
- NLP-style intent matching

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENT NO–14

OBJECTIVE: Design a Constraint Satisfaction Problem solver, e.g., Sudoku, or Map Coloring.

Problem: Map Coloring (Australia)
Goal: Color each region so that no adjacent regions have the same color.
Regions:
- WA, NT, SA, Q, NSW, V, T

Colors:
- red, green, blue

Constraints:
- Adjacent regions must have different colors.

SWI-Prolog Program (Map Coloring CSP)

```
:-use_module(library(clpfd)).

Map coloring solver
map_coloring(Solution) :-

Variables: one per region
   Solution = [WA, NT, SA, Q, NSW, V, T],

    Domain: 1=red, 2=green, 3=blue
   Solution ins 1..3,

Adjacency constraints
WA  #\= NT,
WA  #\= SA,

NT  #\= SA,
NT  #\= Q,

SA  #\= Q,
SA  #\= NSW,
SA  #\= V,

   Q   #\= NSW,

   NSW #\= V,
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

Tasmania has no neighbors (no constraint)

Labeling (search for solution)

```
labeling([], Solution).
```

Convert numeric solution to color names

```
color(1, red).
color(2, green).
color(3, blue).
print_solution([WA, NT, SA, Q, NSW, V, T]) :-
color(WA, WA_Color),
color(NT, NT_Color),
color(SA, SA_Color),
color(Q, Q_Color),
color(NSW, NSW_Color),
color(V, V_Color),
color(T, T_Color),
format('WA  = ~w~n', [WA_Color]),
format('NT  = ~w~n', [NT_Color]),
format('SA  = ~w~n', [SA_Color]),
format('Q   = ~w~n', [Q_Color]),
format('NSW = ~w~n', [NSW_Color]),
format('V   = ~w~n', [V_Color]),
format('T   = ~w~n', [T_Color]).
```

How to Run in SWI-Prolog

1. Save as map_coloring.pl
2. Open SWI-Prolog
3. Load the file:

```
?- [map_coloring].
```

4. Run the solver:

```
?-map_coloring(S), print_solution(S).
```

Sample Output

```
WA  = red
NT  = green
SA  = blue
Q   = red
NSW = green
V   = red
T   = red
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

CSP Components Explained

CSP Element	In Program
Variables	WA, NT, SA, Q, NSW, V, T
Domains	ins 1..3
Constraints	#\= (not equal)
Search	labeling/2

EXPERIMENT NO–15

OBJECTIVE: Perform simple Bayesian reasoning for a probability-based decision problem (e.g., medical diagnosis).

Problem (Medical Diagnosis)

- Disease: **Flu**
- Test: **Flu Test**

Given:

- Prior probability of flu:

$$P(Flu) = 0.05$$

- Test sensitivity (true positive rate):

$$P(Positive|Flu) = 0.90$$

- Test specificity (true negative rate):

$$P(Negative|NoFlu) = 0.95$$

$$\Rightarrow P(Positive|NoFlu) = 0.05$$

Goal:

- Compute **posterior probability**

$$P(Flu|Positive)$$

Bayesian Formula

$$P(Flu|Positive) = \frac{P(Positive|Flu)\,P(Flu)}{P(Positive|Flu)\,P(Flu) + P(Positive|NoFlu)\,P(NoFlu)}$$ Type equation here.

SWI-Prolog Program

% Prior probabilities

```
prior(flu, 0.05).
prior(no_flu, 0.95).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
% Test likelihoods
likelihood(positive, flu, 0.90).
likelihood(positive, no_flu, 0.05).

% Bayes theorem computation
posterior(Disease, Probability) :-
prior(Disease, Prior),
likelihood(positive, Disease, Likelihood),
   % Total probability of positive test
total_positive(Total),
   Probability is (Likelihood * Prior) / Total.

% Total probability of a positive test
total_positive(Total) :-
prior(flu, PFlu),
likelihood(positive, flu, LFlu),
prior(no_flu, PNoFlu),
likelihood(positive, no_flu, LNoFlu),
   Total is (LFlu * PFlu) + (LNoFlu * PNoFlu).
```

Running the Program (SWI-Prolog)

Query

```
?- posterior(flu, P).
```

Output

P = 0.4864864864864865.

Interpretation

- Even with a **positive test**, the probability of actually having the flu is ~**48.6%**
- This shows how **false positives** and **low prior probability** affect diagnosis
- A classic example of why Bayesian reasoning is critical in medicine

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENT NO–16

OBJECTIVE: Build a Sentiment Analysis Classifier using Machine Learning (Naïve Bayes/Logistic Regression)

Problem Overview

- Classify text as **positive** or **negative**
- Use **Machine Learning (Naïve Bayes)**
- Implemented fully in **SWI-Prolog**
- Includes **training data, classifier, and output**

Approach (Naïve Bayes)

1. Training data labelled as positive or negative
2. Tokenize sentence into words
3. Compute word frequencies per class
4. Calculate probabilities
5. Predict sentiment based on highest probability

SWI-Prolog Program: Sentiment Analysis (Naïve Bayes)

```
/* ====================================
   SENTIMENT ANALYSIS - NAIVE BAYES
   SWI-Prolog
   ==================================== */
:- dynamic word_count/3.
:- dynamic class_count/2.
:- dynamic total_words/2.
% ----------------------------
% Training Data
% ----------------------------
train(positive, "i love this movie").
train(positive, "this film is great").
train(positive, "amazing experience").
train(positive, "i enjoyed the movie").

train(negative, "i hate this movie").
train(negative, "this film is terrible").
train(negative, "worst experience").
train(negative, "i disliked the movie").
% ----------------------------
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
% Tokenizer
% ----------------------------
tokenize(Text, Tokens) :-
split_string(Text, " ", "", Words),
maplist(string_lower, Words, Tokens).

% ----------------------------
% Training Phase
% ----------------------------
train_model :-

retractall(word_count(_,_,_)),
retractall(class_count(_,_)),
retractall(total_words(_,_)),
forall(train(Class, Text),
       (
tokenize(Text, Tokens),
update_class(Class),
update_words(Class, Tokens)
       )
   ).
update_class(Class) :-
( class_count(Class, N)
     -> N1 is N + 1,
retract(class_count(Class, N)),
assert(class_count(Class, N1))
;  assert(class_count(Class, 1))
   ).
update_words(_, []).
update_words(Class, [Word|Rest]) :-
( word_count(Class, Word, N)
     -> N1 is N + 1,
retract(word_count(Class, Word, N)),
assert(word_count(Class, Word, N1))
;  assert(word_count(Class, Word, 1))
   ),
( total_words(Class, TW)
     -> TW1 is TW + 1,
retract(total_words(Class, TW)),
assert(total_words(Class, TW1))
;  assert(total_words(Class, 1))
   ),
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
update_words(Class, Rest).

% ------------------------------
% Prediction
% ------------------------------
predict(Text, Sentiment) :-
tokenize(Text, Tokens),
score_class(positive, Tokens, PosScore),
score_class(negative, Tokens, NegScore),
( PosScore> NegScore
    -> Sentiment = positive
;  Sentiment = negative
  ).
score_class(Class, Tokens, Score) :-
class_count(Class, ClassDocs),
findall(N, class_count(_, N), AllDocs),
sum_list(AllDocs, TotalDocs),
Prior is log(ClassDocs / TotalDocs),
total_words(Class, TW),
score_words(Class, Tokens, TW, WordScore),
  Score is Prior + WordScore.
score_words(_, [], _, 0).
score_words(Class, [Word|Rest], TW, Score) :-
( word_count(Class, Word, WC)
    -> Prob is (WC + 1) / (TW + 1000)
;  Prob is 1 / (TW + 1000)
  ),
LogProb is log(Prob),
score_words(Class, Rest, TW, RestScore),
  Score is LogProb + RestScore.
% ------------------------------
% Run Example
% ------------------------------
run :-
train_model,
write("Enter a sentence: "), nl,
read_line_to_string(user_input, Text),
predict(Text, Sentiment),
nl,
write("Predicted Sentiment: "),
write(Sentiment), nl.
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

How to Run in SWI-Prolog

?- run.

Sample Output

Input:

i love this film

Output:

Predicted Sentiment: positive

Input:

this movie is terrible

Output:

Predicted Sentiment: negative

Key Concepts Used

- Machine Learning: **Naïve Bayes**
- Natural Language Processing
- Tokenization
- Probabilistic classification
- Dynamic predicates in Prolog

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

EXPERIMENT NO–17

OBJECTIVE: Implement Genetic Algorithm (GA) to solve an optimization problem (eg; TSP or String Matching).

SWI-Prolog implementation of a Genetic Algorithm (GA) for a String matching optimization problem.

Goal: Evolve a population of strings until it matches a target string
Fitness: Number of matching characters in correct positions
Operators: Selection, Crossover, Mutation
Output: Shows generation-by-generation improvement

Problem Definition
Target string: "hello"
Each chromosome is a list of characters of the same length.

GA in SWI-Prolog (String Matching)
Code: ga_string_match.pl

```
:-use_module(library(random)).

% ----------------------------
% PARAMETERS
% ----------------------------
population_size(10).
mutation_rate(0.1).
max_generations(100).
target("hello").

% ----------------------------
% ENTRY POINT
% ----------------------------
run :-
target(Target),
string_chars(Target, TargetChars),
length(TargetChars, Len),
population_size(PopSize),
init_population(PopSize, Len, Population),
evolve(0, Population, TargetChars).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
% ----------------------------
% INITIAL POPULATION
% ----------------------------
init_population(0, _, []).
init_population(N, Len, [Chrom|Rest]) :-
   N > 0,
random_string(Len, Chrom),
   N1 is N - 1,
init_population(N1, Len, Rest).

random_string(0, []).
random_string(N, [C|Rest]) :-
   N > 0,
random_between(97, 122, Code), % a-z
char_code(C, Code),
   N1 is N - 1,
random_string(N1, Rest).
% ----------------------------
% FITNESS FUNCTION
% ----------------------------
fitness([], [], 0).
fitness([H|T1], [H|T2], F) :-
fitness(T1, T2, F1),
   F is F1 + 1.
fitness([H1|T1], [H2|T2], F) :-
   H1 \= H2,
fitness(T1, T2, F).

% ----------------------------
% EVOLUTION LOOP
% ----------------------------
evolve(Gen, Population, Target) :-
evaluate(Population, Target, Scored),
sort(2, @>=, Scored, Sorted),
   Sorted = [(Best, BestFit)|_],

format("Generation ~w: Best = ~w Fitness = ~w~n",
        [Gen, Best, BestFit]),
length(Target, Len),
( BestFit=:= Len ->
format(" Solution found: ~w~n", [Best])
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
max_generations(MaxGen),
        Gen <MaxGen,
next_generation(Sorted, NewPopulation),
        Gen1 is Gen + 1,
evolve(Gen1, NewPopulation, Target)
    ).
% ----------------------------
% EVALUATION
% ----------------------------
evaluate([], _, []).
evaluate([Chrom|Rest], Target, [(Chrom, Fit)|Scored]) :-
fitness(Chrom, Target, Fit),
evaluate(Rest, Target, Scored).

% ----------------------------
% SELECTION + CROSSOVER
% ----------------------------

next_generation(Scored, NewPopulation) :-
findall(Child,
        (member((Parent1,_), Scored),
         member((Parent2,_), Scored),
crossover(Parent1, Parent2, Child),
mutate(Child)),
        Children),
population_size(PopSize),
length(NewPopulation, PopSize),
append(Children, _, NewPopulation).

crossover(P1, P2, Child) :-
length(P1, Len),
random_between(1, Len, Point),
length(Front, Point),
append(Front, Back1, P1),
append(_, Back2, P2),
append(Front, Back2, Child).
```

Vision of the Institute	Mission of the Institute
"To become a leading institute of providing professionally competent and socially responsive technocrats with high moral values."	⇨ To create an eco-system for the dissemination of technical knowledge, to achieve academic excellence. ⇨ To develop technocrats with creative skills and leadership qualities, to solve local and global challenges. ⇨ To impart human values and ethics in students, to make them socially and Eco-friendly responsible.

```
% ----------------------------
% MUTATION
% ----------------------------
mutate(Chrom) :-
mutation_rate(Rate),
maybe_mutate(Chrom, Rate).

maybe_mutate([], _).
maybe_mutate([H|T], Rate) :-
random_float(R),
( R< Rate ->
random_between(97, 122, Code),
char_code(NewH, Code),
      H = NewH
   ;
      true
   ),
maybe_mutate(T, Rate).
```

How to Run (SWI-Prolog)

```
?- [ga_string_match].
?- run.
```

Sample Output

```
Generation 0: Best = [h,e,q,l,a] Fitness = 3
Generation 1: Best = [h,e,l,l,a] Fitness = 4
Generation 2: Best = [h,e,l,l,o] Fitness = 5
Solution found: [h,e,l,l,o]
```

Key GA Concepts Demonstrated

GA Component	Implemented
Population	✓
Fitness Function	✓
Selection	✓
Crossover	✓
Mutation	✓
Termination	✓