**LAB MANUAL**

**SUB: ARTIFICIAL INTELLIGENCE**

**CODE: 3170716**

**PROLOG**

**Vision of the Institute**

To be an exemplary institute, transforming students into competent professionals with human values.

**Mission of the Institute**

* To provide a conducive academic environment for strengthening technical capabilities of the students.
* To strengthen linkage with industries, alumni and professional bodies.
* To organise various co-curricular and extra-curricular activities for overall development of the students.
* To practice good governance and conduct value- based activities for making students responsible citizens.

**Vision of the Department**

Transforming students into globally efficient professionals with moral values.

**Mission of the Department**

* To provide a strong foundation of computer engineering through effective teaching learning process.
* To enhance industry linkage & alumni network for better placement and real-world exposure.
* To provide various opportunities & platforms for all round development of students & encourage them for value-based practices.

**Program Educational Objectives (PEOs)**

Graduates will be able to

* Apply computer engineering theories, principles and skills to meet the challenges of the society.
* Communicate effectively, work collaboratively and manifest professionalism with ethics.
* Exhibit life-long learning attitude and adapt to rapid technological changes in industry.
* Advance their career in industry, pursue higher education or become an entrepreneur.

**Program Specific Outcomes (PSOs)**

Graduates will be able to

* **PSO1:** Apply fundamental knowledge of hardware and software aspects of computer systems.
* **PSO2:** Analyze, model and develop computer applications by adapting emerging technologies and standard practices of software project development to meet the requirements of industry and society.
* **PSO3:** Use different programming languages and open-source platforms.

Subject: Artificial Intelligence

Code: 3170716

After learning the course, the students will be able to:

|  |  |  |
| --- | --- | --- |
| **CO NO.** | **Course Outcomes** | **CL** |
| C70716.1 | Apply basic principles of AI in problem solving | Ap |
| C70716.2 | Evaluate knowledge representation issues using AI rules. | E |
| C70716.3 | Analyze working of reasoning in presence of incomplete and/or uncertain information. | A |
| C70716.4 | Demonstrate awareness of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models. | U |
| C70716.5 | Apply Prolog Programming language for Basic AI problems. | Ap |

# Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ex No.** | **EXPERIMENT** | **NO. OF LAB** | **Grades** | **Sign** |
| 1 | 1.What is prolog? | LAB 1 |  |  |
| 2.Difference between OOL and procedural Languages. |
| 3.Benefits of using prolog |
| 4.Feature of prolog |
| 5.Application of prolog. |
| 1 | Write a program for the following task, | LAB 2  &  LAB 3 |  |  |
| 2 | Write a program for the family tree. |  |  |
| 3 | Write a program to check whether given character is a character or digit. | LAB 4  &  LAB 5 |  |  |
| 4 | Write a program to generate a random number with respect to entered digit. |  |  |
| 5 | Write a program to implement Calculator recursively. |  |  |
| 6 | Write a program to implement login system. | LAB 6 |  |  |
| 7 | Write a program to implement login system recursively. |  |  |
| 8 | Write a program to implement login system using repeat predicate. |  |  |
| 9 | Write a program to display the element of given list. | LAB 7 |  |  |
| 10 | Write a program to check whether the given element is in the list or not. |  |  |
| 11 | Write a program to find the largest number from a given list. |  |  |
| 12 | Write a program to print last element of the list. | LAB 8 |  |  |
| 13 | Write a program to find length of list. |  |  |
| 14 | Write a program to generate sum of elements of given list. |  |  |
| 15 | Write a program to union of two lists. | LAB 9  &  LAB10 |  |  |
| 16 | Write a program to reverse a given list. |  |  |
| 17 | Write a program to convert integer list to equivalent symbol list. | LAB 11 &  LAB 12 |  |  |
| 18 | Write a program to create, read, write and delete a file |  |  |

**1. What is prolog?**

* Prolog stands for programming in logic. In the logic programming paradigm, prolog language is most widely available. Prolog is a declarative language, which means that a program consists of data based on the facts and rules (Logical relationship) rather than computing how to find a solution. A logical relationship describes the relationships which hold for the given application.
* To obtain the solution, the user asks a question rather than running a program. When a user asks a question, then to determine the answer, the run time system searches through the database of facts and rules.
* The first Prolog was 'Marseille Prolog', which is based on work by Colmerauer. The major example of fourth-generation programming language was prolog. It supports the declarative programming paradigm.
* In 1981, a Japanese computer Project of 5th generation was announced. After that, it was adopted Prolog as a development language. In this tutorial, the program was written in the 'Standard' Edinburgh Prolog. Prologs of PrologII family are the other kind of prologs which are descendants of Marseille Prolog.
* Prolog features are 'Logical variable', which means that they behave like uniform data structure, a backtracking strategy to search for proofs, a pattern-matching facility, mathematical variable, and input and out are interchangeable.
* To deduce the answer, there will be more than one way. In such case, the run time system will be asked to find another solution. To generate another solution, use the backtracking strategy. Prolog is a weakly typed language with static scope rules and dynamic type checking.
* Prolog is a declarative language that means we can specify what problem we want to solve rather than how to solve it.

1. **Difference between OOL and procedural Languages.**

|  |  |  |  |
| --- | --- | --- | --- |
| Procedural Oriented Programming | | Object Oriented Programming | |
| In procedural programming, program is divided into small parts called **functions**. | | In object oriented programming, program is divided into small parts called **objects**. | |
| Procedural programming follows **top down approach**. | Object oriented programming follows **bottom up approach**. |
| There is no access specifier in procedural programming. | Object oriented programming have access specifiers like private, public, protected etc. |
| Adding new data and function is not easy.  Procedural programming does not have | Adding new data and function is easy. |
| any proper way for hiding data so it is **less secure**. | Object oriented programming provides data hiding so it is **more secure**. |
| In procedural programming, overloading | Overloading is possible in object oriented |
| Procedural Oriented Programming | | Object Oriented Programming | |
| is not possible. | | programming. | |

In procedural programming, function is In object oriented programming, data is more important than data. more important than function.

Procedural programming is based Object oriented programming is based on **unreal world**. on **real world**.

Examples: C, FORTRAN, Pascal, Basic etc. Examples: C++, Java, Python, C# etc.

1. **Application of prolog.**

* Prolog is used in some areas like database, natural language processing, artificial intelligence, but it is pretty useless in some areas like a numerical algorithm or instance graphics.
* In artificial intelligence applications, prolog is used. The artificial intelligence applications can be automated reasoning systems, natural language interfaces, and expert systems. The expert system consists of an interface engine and a database of facts. The prolog's run time system provides the service of an interface engine.
* A basic logic programming environment has no literal values. An identifier with upper case letters and other identifiers denote variables. Identifiers that start with lower-case letters denote data values. The basic Prolog elements are typeless. The most implementations of prolog have been enhanced to include integer value, characters, and operations. The Mechanism of prolog describes the tuples and lists.
* Functional programming language and prolog have some similarities like Hugs. A logic program is used to consist of relation definition. A functional programming language is used to consist of a sequence of function definitions. Both the logical programming and functional programming rely heavily on recursive definitions.
* Specification Language
* Robot Planning
* Natural language understanding
* Machine Learning
* Problem Solving
* Intelligent Database retrieval
* Expert System
* Automated Reasoning

**4. Feature of prolog**

The main characteristics/notions of the Visual Prolog programming language are:

* based on logical programming with Horn clauses
* fully object oriented
* object predicate values (delegates)
* strongly typed
* algebraic data types
* pattern matching and unification
* controlled non-determinism
* fully integrated fact databases
* supports parametric polymorphism
* automatic memory management
* supports direct linkage with C/C++
* supports direct calling of Win32 API functions

**5. Benefits of using Prolog.**

* Easy to build database. Doesn’t need a lot of programming effort.
* Pattern matching is easy. Search is recursion based.
* It has built in list handling. Makes it easier to play with any algorithm involving lists.

**1. Write a program for the following task,**

1. **Students who are living in Rajkot.**
2. **Students whose age is greater than 15 and living in Rajkot.**
3. **Student who has more than 60%.**

**---------------------------------------------------------------------------------------------------------------**

domains

predicates

student(symbol,symbol,integer,integer) clauses student("Sita","Rajkot",20,70). student("Rita","Delhi",35,55). student("Nita","Bangalore",19,90). student("Mita","Rajkot",55,30). student("Hita","Mumbai",60,67).

goal

/\* student(Name,"Rajkot",Age,Percentage).\*/

/\* student(Name,"Rajkot",Age,Percentage), Age>50.\*/ /\* write("Students with percentage greater than 60").\*/ student(Name,City,Age,Percentage), Percentage > 60.

**---------------------------------------------------------------------------------------------------------------**

**2. Write a program for the family tree.**

## ---------------------------------------------------------------------------------------------------------------

/\*

|  |  |
| --- | --- |
|  | Rohan-Rina |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | | | | | |
|  | Nita-Numa Gita-Gautam Sonu-Sonia Monu-Mina |
| | | | |
|  | \_\_\_\_|\_\_\_\_ \_\_\_\_|\_\_\_\_\_\_ \_\_\_\_|\_\_\_\_\_ |
|  | | | | | | | |
| \*/ | Rosh Posh Sam Pam John Jenny |

domains

X,Y = String

predicates

male(String) female(String) parent(String,String) father(String,String) mother(String,String) brother(String,String) sister(String,String) grandfather(String,String) grandmother(String,String) uncle(String,String) uncle1(String,String) aunt(String,String)

aunt1(String,String)

clauses male("Rohan"). male("Sonu"). male("Monu"). male("Sam"). male("John"). male("Gautam"). male("Numa"). male("Rosh"). male("Posh").

female("Rina"). female("Nita"). female("Gita"). female("Pam"). female("Jenny"). female("Mina"). female("Sonia").

parent("Nita","Rohan").

parent("Gita","Rohan"). parent("Sonu","Rohan"). parent("Monu","Rohan"). parent("Nita","Rina"). parent("Gita","Rina"). parent("Sonu","Rina"). parent("Monu","Rina"). parent("Rosh","Numa"). parent("Posh","Numa"). parent("Rosh","Nita"). parent("Posh","Nita"). parent("Sam","Gautam"). parent("Pam","Gautam"). parent("Sam","Gita"). parent("Pam","Gita"). parent("","Sonu"). parent("","Sonia"). parent("John","Monu"). parent("Jenny","Monu"). parent("John","Mina"). parent("Jenny","Mina").

mother(X,Y):-

female(Y), parent(X,Y).

brother(X,Y):-

male(Y),

father(X,Z), father(Y,Z),X<>Y.

sister(X,Y):-

female(Y), father(X,Z), father(Y,Z),X<>Y.

grandfather(X,Y):- male(Y), parent(X,Z), parent(Z,Y).

grandmother(X,Y):- female(Y), parent(X,Z), parent(Z,Y).

uncle1(X,Y):- male(Y), parent(X,Z), brother(Z,Y).

uncle(X,Y):- uncle1(X,Y); aunt1(X,Z), mother(P,Z), father(P,Y).

aunt1(X,Y):-

female(Y), parent(X,Z), sister(Z,Y).

aunt(X,Y):- aunt1(X,Y);

uncle1(X,Z), parent(P,Z), mother(P,Y).

father(X,Y):-

|  |  |
| --- | --- |
|  | male(Y), |
|  | parent(X,Y). |
| goal |  |
|  | aunt1(X,Y). |

**Quiz:**

**Which is not a property of representation of knowledge?**

1. **Representational Verification**
2. **Representational Adequacy**
3. **Inferential Adequacy**
4. **Inferential Efficiency**

**ANSWER:**

**Turing developed a technique for determining whether a computer could or could not demonstrate the artificial Intelligence, Presently, this technique is called**

1. **Turing Test**
2. **Algorithm**
3. **Boolean Algebra**
4. **Logarithm ANSWER:**

**A Personal Consultant knowledge base contain information in the form of**

1. **parameters**
2. **contexts**
3. **production rules D. all of the mentioned**

**ANSWER:**

**Which approach to speech recognition avoids the problem caused by the variation in speech patterns among different speakers?**

1. **Continuous speech recognition**
2. **Isolated word recognition**
3. **Connected word recognition D. Speaker-dependent recognition**

**ANSWER:**

**Which of the following, is a component of an expert system?**

1. **inference engine**
2. **knowledge base**
3. **user interface D. all of the mentioned ANSWER:**

**3. Write a program to check whether given character is a character or digit.**

**---------------------------------------------------------------------------------------------------------------**

domains predicates

getinput()

check(integer)

clauses

getinput():-

readchar(X),

write("Input is ",X),

char\_int(X,Y), check(Y).

check(Y):-

Y>=97, Y<=122, write("\nIt is a lowercase character\n").

check(Y):-

Y>=65,

Y<=91, write("\nIt is a uppercase character\n").

check(Y):-

Y>=47,

Y<=56, write("\nIt is a digit\n").

goal

getinput(); write("\nIt is neither a digit nor a character\n").

**---------------------------------------------------------------------------------------------------------------**

**4. Write a program to generate a random number with respect to entered digit.**

## ---------------------------------------------------------------------------------------------------------------

domains predicates

getinput()

clauses

getinput():- readint(X),

write("Input is ",X),

X>0,

random(X,Y), write("\nRandom number is ",Y,"\n").

goal

getinput(); write("\nInput is invalid\n").

## ---------------------------------------------------------------------------------------------------------------

**5. Write a program to implement Calculator recursively.**

**---------------------------------------------------------------------------------------------------------------**

domains predicates

getinput() check(integer) calculator(integer,integer,integer)

clauses

getinput():-

write("\n1)ADDITION \n2)SUBTRACTION

\n3)MULTIPLICATIN \n4)DIVISION \n5)EXIT \n\nEnter your choice:-"),

readint(C),

check(C),

write("\nEnter first number: "), readint(X),

write("Enter second number: "),

readint(Y),

calculator(C,X,Y),

getinput(),

write("\n");

write("\nInput is invalid please try again\n"), getinput().

check(C):-

C=5,

exit(1);

C>5,

write("\nInput is invalid please try again\n"),

getinput();

C<5.

calculator(1,X,Y):-

Z=X+Y, write("\nAddition is ",Z,"\n").

calculator(2,X,Y):-

Z=X-Y, write("\nSubtraction is ",Z,"\n").

calculator(3,X,Y):-

Z=X\*Y, write("\nMultiplication is ",Z,"\n").

calculator(4,X,Y):-

Y<>0, Z=X/Y,

write("\nDivision is ",Z,"\n"); write("\nDivisin by zero is not allowed\n").

/\* calculator(C,X,Y):- write("\nInvalid choice\n").

\*/

goal getinput().

**Quiz:**

**Visual clues that are helpful in computer vision include**

1. **color and motion**
2. **depth and texture**
3. **height and weight**
4. **color and motion, depth and texture ANSWER:**

**General games involves**

1. **Single-agent**
2. **Multi-agent**
3. **Neither Single-agent nor Multi-agent**
4. **Only Single-agent and Multi-agent ANSWER:**

**Adversarial search problems uses**

1. **Competitive Environment**
2. **Cooperative Environment**
3. **Neither Competitive nor Cooperative Environment**
4. **Only Competitive and Cooperative Environment ANSWER:**

**Zero sum game has to be a game.**

1. **Single player**
2. **Two player**
3. **Multiplayer D. Three player ANSWER:**

**A game can be formally defined as a kind of search problem with the following components.**

1. **Initial State**
2. **Successor Function**
3. **Terminal Test D. All of the mentioned ANSWER:**

**Write a program to implement login system.**

## ---------------------------------------------------------------------------------------------------------------

domains

Name, Password = symbol predicates getinput(Name,Password). user(Name, Password). login.

clauses

getinput(Name,Password):- write(“Enter User Name : ”),nl, readln(Name),nl, write(“Enter Password”),nl, readln(Password),nl, user(Name,Password).

user(“abc”,”123”). user(“def”,”456”).

login:-

getinput(Name,Password),nl, write(“Login Successful.”).

login:- write(“Your are not authorized to login.”).

goal login.

**---------------------------------------------------------------------------------------------------------------**

**Write a program to implement login system recursively.**

## ---------------------------------------------------------------------------------------------------------------

domains

Name, Password = symbol

predicates getinput(Name,Password). user(Name, Password). login.

clauses

getinput(Name,Password):- write(“Enter User Name : ”),nl, readln(Name),nl, write(“Enter Password”),nl, readln(Password),nl, user(Name,Password).

user(“abc”,”123”). user(“def”,”456”).

login:-

getinput(Name,Password),nl, write(“Login Successful.”).

login:- write(“Your are not authorized to login.”). login.

goal login.

**---------------------------------------------------------------------------------------------------------------**

**Write a program to implement login system using repeat predicate.**

## ---------------------------------------------------------------------------------------------------------------

domains

Name, Password = symbol predicates getinput(Name,Password). user(Name, Password). login.

Repeat.

|  |  |
| --- | --- |
| clauses |  |
|  | repeat. |
|  | repeat:- |
|  | repeat. |
|  | getinput(Name,Password):- |
|  | write(“Enter User Name : ”),nl, |
|  | readln(Name),nl, |
|  | write(“Enter Password”),nl, |
|  | readln(Password),nl, |
|  | user(Name,Password). |
|  | user(“abc”,”123”). |
|  | user(“def”,”456”). |
|  | login:- |
|  | repeat, |
|  | getinput(\_,\_),nl, |
|  | write(“Login Successful.”). |
|  |  |
|  | login:- |
|  | repeat, |
| Again…”), | write(“Your are not authorized to login. Try |
|  | getinput(\_,\_),nl, |
|  | write(“Login Successful”). |

goal login.

## ---------------------------------------------------------------------------------------------------------------

**Quiz:**

**A computer vision technique that relies on image templates is**

1. **edge detection**
2. **binocular vision**

**8th IT- Al Question Bank**

1. **model-based vision**
2. **robot vision**

**ANSWER:**

**DARPA, the agency that has funded a great deal of American Artificial Intelligence research, is part of the Department of**

1. **Defense**
2. **Energy**
3. **Education**
4. **Justice**

**ANSWER:**

**Which of these schools was not among the early leaders in Artificial Intelligence research?**

1. **Dartmouth University**
2. **Harvard University**
3. **Massachusetts Institute of Technology**
4. **Stanford University ANSWER:**

**Who is the “father” of artificial intelligence?**

1. **Fisher Ada**
2. **John McCarthy**
3. **Allen Newell D. Alan Turning ANSWER:**

**A process that is repeated, evaluated, and refined is called**

1. **diagnostic**
2. **descriptive**
3. **interpretive**
4. **iterative ANSWER:**

**Write a program to display the element of given list.**

**---------------------------------------------------------------------------------------------------------------**

domains

list = symbol\*

predicates

go

disp(list)

clauses

go:-

X = [abc,asdf,qwer], disp(X).

disp([]).

disp([H|T]):-

write(H), nl, disp(T).

goal go.

**---------------------------------------------------------------------------------------------------------------**

**Write a program to check whether the given element is in the list or not.**

**---------------------------------------------------------------------------------------------------------------**

domains

list = integer\*

predicates

go

present(integer,list)

clauses

go:-

X = [1,2,3,0], write("Enter element: "),

readint(I),

write("Element = ",I),

nl, present(I,X).

present(I,[]):-

T=[],

write("Not present"), nl.

present(I,[H|T]):-

I=H,

write("Present"),

nl,

exit(0);

present(I,T).

goal go.

**---------------------------------------------------------------------------------------------------------------**

**Write a program to find the largest number from a given list.**

**---------------------------------------------------------------------------------------------------------------**

domains

list = integer\*

predicates

go

maximum(integer,list)

clauses

go:-

X = [1,7,3,6], maximum(-1,X).

maximum(Max,[]):-

T=[],

write("Maximum number : ", Max), nl.

maximum(Max,[H|T]):-

Max < H, maximum(H,T);

maximum(Max,T).

goal go.

**Quiz:**

**What is Artificial intelligence?**

1. **Putting your intelligence into Computer**
2. **Programming with your own intelligence**
3. **Making a Machine intelligent**
4. **Playing a Game ANSWER:**

**Strong Artificial Intelligence is**

**A. the embodiment of human intellectual capabilities within a computer B. a set of computer programs that produce output that would be considered**

**to reflect intelligence if it were generated by humans**

**C. the study of mental faculties through the use of mental models implemented on a computer D. all of the mentioned ANSWER:**

**In which of the following situations might a blind search be acceptable?**

1. **real-life situation**
2. **complex game**
3. **small search space D. all of the mentioned ANSWER:**

**Which search method takes less memory?**

1. **Depth-First Search**
2. **Breadth-First search**
3. **Optimal search**
4. **Linear Search ANSWER:**

**A heuristic is a way of trying**

1. **To discover something or an idea embedded in a program**
2. **To search and measure how far a node in a search tree seems to be from a goal**
3. **To compare two nodes in a search tree to see if one is better than the other is**
4. **All of the mentioned ANSWER:**

1. **Write a program to print last element of the list.**

**---------------------------------------------------------------------------------------------------------------**

domains

list = symbol\*

predicates

go

last(list)

clauses

go:-

X = [abc,asdf,qwer], last(X).

last([H|T]):-

T=[],

write("Last element is ",H), nl.

last([H|T]):- last(T).

goal go.

**---------------------------------------------------------------------------------------------------------------**

1. **Write a program to find length of list.**

**---------------------------------------------------------------------------------------------------------------**

domains

list = symbol\*

predicates

go

count(integer,list) clauses

go:-

X = [abc,asdf,qwer], count(0,X).

count(C,[]):-

T=[],

write("Total number of elements : ",C), nl.

count(C,[H|T]):-

B = C + 1, count(B,T).

goal go.

**Quiz:**

**The initial state and the legal moves for each side define the for the game.**

* 1. **Search Tree**
  2. **Game Tree**
  3. **State Space Search**
  4. **Forest**

**ANSWER:**

**General algorithm applied on game tree for making decision of win/lose is**

1. **DFS/BFS Search Algorithms**
2. **Heuristic Search Algorithms**
3. **Greedy Search Algorithms**
4. **MIN/MAX Algorithms ANSWER:**

**What is the complexity of minirnax algorithm?**

1. **Same as of DES**
2. **Space — bm and time — bm**
3. **Time — bm and space — bm**
4. **Same as BFS ANSWER:**

**Which is the most straightforward approach for planning algorithm?**

1. **Best-first search**
2. **State-space search**
3. **Depth-first search D. Hill-climbing search**

**ANSWER:**

**What are taken into account of state-space search?**

1. **Postconditions**
2. **Preconditions**
3. **Effects**
4. **Both Preconditions & Effects ANSWER:**

**14. Write a program to generate sum of elements of given list.**

**---------------------------------------------------------------------------------------------------------------**

domains

list = integer\*

predicates

go

count(integer,list) clauses

go:- X = [1,2,3], count(0,X).

count(Sum,[]):-

T=[],

write("Sum of elements : ",Sum), nl.

count(Sum1,[H|T]):- Sum = Sum1 + H, count(Sum,T).

goal go.

**---------------------------------------------------------------------------------------------------------------**

**15. Write a program to union of two lists.**

**---------------------------------------------------------------------------------------------------------------**

domains

list = symbol\*

predicates

go

union(list,list,list)

disp(list)

clauses

go:-

* 1. = [a1,b1,c1],
  2. = [d1,e1], Z = [], union(Z,X,Y).

union(Z,[],[]):- disp(Z).

union(Z,S,[H|T]):-

N = [H|Z], union(N,S,T).

union(Z,[H|T],S):-

M = [H|Z], union(M,T,S).

disp([]).

disp([H|T]):-

write(H), nl, disp(T).

goal go.

**Quiz:**

**How many ways are available to solve the state-space search?**

1. **1**
2. **2**
3. **3**
4. **4**

**ANSWER:**

**What is the other name for forward state-space search?**

1. **Progression planning**
2. **Regression planning**
3. **Test planning D. None of the mentioned ANSWER:**

**How many states are available in state-space search?**

1. **1**
2. **2**
3. **3**
4. **4**

**ANSWER:**

**What is the main advantage of backward state-space search?**

1. **Cost**
2. **Actions**
3. **Relevant actions D. All of the mentioned ANSWER:**

**What is the other name of the backward state-space search?**

1. **Regression planning**
2. **Progression planning**
3. **State planning**
4. **Test planning ANSWER:**

1. **Write a program to reverse a given list.**

**---------------------------------------------------------------------------------------------------------------**

domains

list = integer\*

predicates

go

rev(list,list)

disp(list)

clauses

go:-

X = [1,2,3], Y = [], rev(X,Y).

rev([],Y):- disp(Y).

rev([H|T],Y):-

M = [H|Y],

nl, rev(T,M).

disp([]).

disp([H1|T1]):-

write(H1), nl, disp(T1).

goal go.

**---------------------------------------------------------------------------------------------------------------**

1. **Write a program to convert integer list to equivalent symbol list.**

## ---------------------------------------------------------------------------------------------------------------

domains

ilist=integer\* slist=symbol\*

predicates

go

conv(ilist,slist) int\_sym(integer,symbol)

clauses int\_sym(0,zero). int\_sym(1,one). int\_sym(2,two). int\_sym(3,three). int\_sym(4,four). int\_sym(5,five). int\_sym(6,six). int\_sym(7,seven). int\_sym(8,eight). int\_sym(9,nine).

go:-

I=[1,2,3,4,5,6,7],

S=[], conv(I,S).

conv([],S):- write(S).

conv([H|T],S):- int\_sym(H,A), M=[A|S], conv(T,M).

goal go.

**Quiz:**

**What is meant by consistent in state-space search?**

1. **Change in the desired literals**
2. **Not any change in the literals**
3. **No change in goal state**
4. **None of the mentioned**

**ANSWER:**

**What will happen if a predecessor description is generated that is satisfied by the initial state of the planning problem?**

1. **Success**
2. **Error**
3. **Compilation**
4. **Termination**

**ANSWER: 0**

**Which approach is to pretend that a pure divide and conquer algorithm will work?**

1. **Goal independence**
2. **Subgoal independence**
3. **Both Goal & Subgoal independence**
4. **None of the mentioned ANSWER:**

**Which search is equal to minimax search but eliminates the branches that can’t influence the final decision?**

1. **Depth-first search**
2. **Breadth-first search**
3. **Alpha-beta pruning D. None of the mentioned ANSWER:**

**Which values are independant in minimax search algorithm?**

1. **Pruned leaves x and y**
2. **Every states are dependant**
3. **Root is independant D. None of the mentioned ANSWER:**

**18. Write a program to create, read, write and delete a file.**

## ---------------------------------------------------------------------------------------------------------------

domains

file=textFile

predicates go

operation(integer) readfile writefile

clauses

go:-

disk("C:\\"), write("1.create"),nl, write("2.read"),nl, write("3.write"),nl, write("4.delete"),nl, write("5.exit"),nl, write("enter choice:"), readint(C),

operation(C), go.

operation(1):-

write("enter Filename to create:"), readln(Name), openwrite(textFile,Name), writedevice(textFile),nl, closefile(textFile), writedevice(screen), write("File Sucessfully created.").

operation(2):-

write("enter Filename to read:"), readln(Name), openread(textFile,Name), readfile, closefile(textFile).

operation(3):-

write("enter Filename to Write:"), readln(Name), openwrite(textFile,Name), writefile,

writedevice(screen), write("file successfully written.").

operation(4):-

write("Enter filename to delete:"), readln(Name), deletefile(Name),

write("file successfully deleted"),nl.

operation(\_):- write("invalid choice."),nl.

readfile:-

readdevice(textFile),

readln(L),

writedevice(screen), readdevice(keyboard), write(L),nl, eof(textFile); readfile.

writefile:- writedevice(screen), write("enter file content:"), readln(N),

writedevice(textFile), write(N),nl, writedevice(screen), write("do you want to enter more?[y/n]"), readchar(C),

C='y', writefile; C='n', closefile(textFile).

goal go.

**Quiz:**

**To which depth does the alpha-beta pruning can be applied?**

1. **10 states**
2. **8 States**
3. **6 States**
4. **Any depth ANSWER:**

**Which search is similar to minimax search?**

**A. Hill-climbing search B. Depth-first search**

**C. Breadth-first search D. All of the mentioned ANSWER:**

**Which value is assigned to alpha and beta in the alpha-beta pruning?**

1. **Alpha = max**
2. **Beta = min**
3. **Beta = max**
4. **Both Alpha = max & Beta = min ANSWER:**

**Where does the values of alpha-beta search get updated?**

1. **Along the path of search**
2. **Initial state itself**
3. **At the end D. None of the mentioned ANSWER:**

**How the effectiveness of the alpha-beta pruning gets increased?**

1. **Depends on the nodes**
2. **Depends on the order in which they are executed**
3. **All of the mentioned D. None of the mentioned ANSWER:**