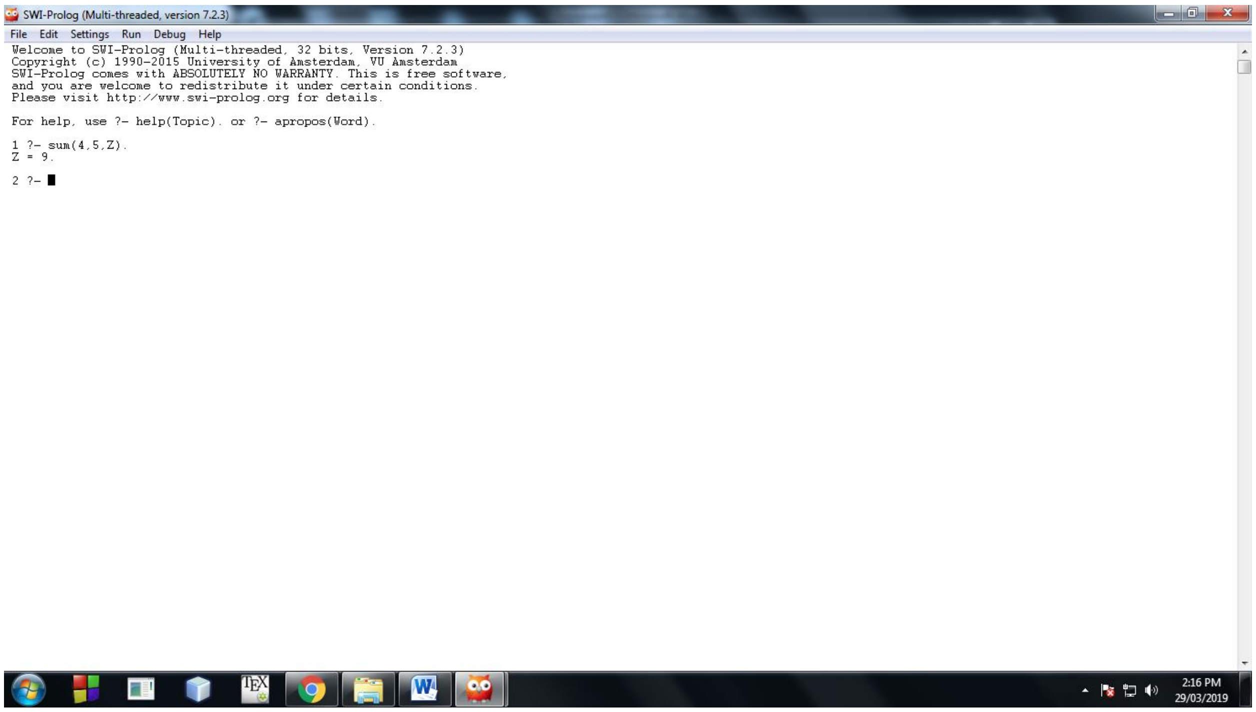
TOPIC : PROLOG PROGRAMS

Write a prolog program to calculate the sum of two numbers.

sum(X,Y,Z):- Z is X+Y.


SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help
2:16 PM
29/03/2019

```
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 7.2.3)
Copyright (c) 1990-2015 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- sum(4,5,Z).
Z = 9.

2 ?-
```

Write a Prolog program to implement max(X, Y, M) so that M is the maximum of two numbers X and Y

max(X,Y,M):-X>Y,M is X.

max(X,Y,M):-Y>=X,M is Y.

Write a program in PROLOG to implement factorial (N, F) where F represents the factorial of a number N.

fact(0,1).

```
fact(N,X):-N1 is N-1,fact(N1,Y),X is Y*N,!.
```

Write a program in PROLOG to implement generate_fib(N,T) where T represents the Nth term of the fibonacci series.

```
fib(1,0).

fib(2,1).

fib(N,X):- N1 is N-1,N2 is N-2,fib(N1,X1),fib(N2,X2),X is X1+X2,!.
```

Write a Prolog program to implement GCD of two numbers.

```
gcd(0,A,A):-!.

gcd(A,0,A):-!.

gcd(A,B,R):-B1 is mod(A,B),gcd(B,B1,R).
```

Write a Prolog program to implement power (Num,Pow, Ans) : where Num is raised to the power Pow to get Ans.

```
power(X,0):- !.

power(Num,Pow,Ans):-Ans is Num^Pow.
```

Prolog program to implement multi (N1, N2, R) : where N1 and N2 denotes the numbers to be multiplied and R represents the result.

```
multi(X,0).

multi(N1,N2,R):- R is N1*N2.
```

Write a program in PROLOG to implement towerofhanoi (N) where N represents the number of discs.

```
move(1,X,Y,_):-      write('Move disk from '),

                             write(X),

                             write(' to '),

                             write(Y),nl.

move(N,X,Y,Z):- N>1,M is N-1,

                     move(M,X,Z,Y),

                             move(1,X,Y,_),

                             move(M,Z,Y,X).


tower_of_hanoi(N) :- move(N,left,right,center).
```

Consider a cyclic directed graph [edge (p, q), edge (q, r), edge (q, r), edge (q, s), edge (s,t)] where edge (A,B) is a predicate indicating directed edge in a graph from a node A to a node B. Write a program to check whether there is a route from one node to another node.

```
edge(p,q).

edge(q,r).

edge(r,q).

edge(q,s).

edge(s,t).
```

```
route(X,Y):- edge(X,Y).

route(X,Y):- edge(X,Z),

          route(Z,Y).
```

Write a Prolog program to implement memb(X, L): to check whether X is a member of L or not.

```
memb(X,[X|Tail]).

memb(X,[Head|Tail]):-memb(X,Tail).
```

Write a Prolog program to implement conc (L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3.

```
conc([],L,L).

conc([X|L1],L2,[X|L3]):-conc(L1,L2,L3).
```

Write a Prolog program to implement reverse (L, R) where List L is original and List R is reversed list.

```
append([],L,L).

append([X|L1],L2,[X|L3]):- append(L1,L2,L3).

reverse([],[]).

reverse([H|T],R):- reverse(T,L1),append(L1,[H],R).
```

Write a program in PROLOG to implement palindrome (L) which checks whether a list L is a palindrome or not.

```
app([],L,L).

app([X|L1],L2,[X|L3]):- app(L1,L2,L3).

pal([]).

pal([_]).

pal(Plist):-app([H|T],[H],Plist),pal(T).
```

References/Resources

- Dan. W. Patterson, Artificial Intelligence and Expert Systems, Prentice Hall, 2004
- Elaine Rich, Kevin Knight, & Shivashankar B Nair, Artificial Intelligence, McGraw Hill, 3rd ed.,2009

NOTE: Please go through the above programs carefully and practice them (on machine if possible).