



Practical File

Name: Debarpann Chatterjee

Roll no: 20078570020

Semester: VI

Subject: Artificial Intelligence

Course: B.Sc Hons Computer Science

Q1 - Write a prolog program to calculate the sum of two numbers.

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

```
?- sum(5,4,X)
|
X = 9.
```

Q2 - 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.
```

```
?- max(5,8,X).
X = 8.
```

Q3 - Write a program in PROLOG to implement factorial (N, F) where F represents the factorial of a number N.

```
fac(0,1).
fac(N,F) :- N>0, N1 is N-1, fac(N1,F1), F is F1*N.
```

```
?- fac(5,F).
F = 120 .
```

Q4 - Write a program in PROLOG to implement generate_fib(N,T) where T represents the Nth term of the fibonacci series.

```
fib(0,0).  
fib(1,1).  
fib(N,F) :- N>0,N1 is N-1, fib(N1,F1), N2 is N-2, fib(N2,F2), F is F1+F2.
```

```
?- fib(5,X).  
X = 5 .
```

Q5 - Write a Prolog program to implement GCD of two numbers.

```
gcd(X,Y,Z):- X=Y, Z is X.  
gcd(X,Y,Z):- X>Y, X1 is X-Y, gcd(X1,Y,Z).  
gcd(X,Y,Z):- Y>X, Y1 is Y-X, gcd(X,Y1,Z).
```

```
?- gcd(90,80,X).  
X = 10 .
```

Q6 - Write a Prolog program to implement power (Num, Pow, Ans) : where Num is raised to the power Pow to get Ans.

```
pow(_,0,1).  
pow(X,Y,Z) :- X>0, Y1 is Y-1, pow(X,Y1,Z1), Z is Z1*X.
```

```
?- pow(5,5,X).  
X = 3125 .
```

Q7 - Prolog program to implement multi (N1, N2, R) : where N1 and N2 denotes the numbers to be multiplied and R represents the result.

```
mul(X,Y,Z):- Z is X*Y.
```

```
?- mul(4,5,X).  
X = 20.
```

Q8 - Write a Prolog program to implement memb(X, L): to check whether X is a member of L or not.

```
mem(H,[H|_]).  
mem(X, [_|T]):- mem(X,T).
```

```
?- mem(3, [4,3,5,6]).  
true .
```

Q9 - 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([],L2,L2).  
conc([H1|L1],L2,[H1|L3]):- conc(L1,L2,L3).
```

```
?- conc([1,2],[6,7],X).  
X = [1, 2, 6, 7].
```

Q10 - Write a Prolog program to implement reverse (L, R) where List L is original and List R is reversed list.

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

```
?- reverse([1,2,3],X).  
X = [3, 2, 1].
```

Q11 - Write a program in PROLOG to implement palindrome (L) which checks whether a list L is a palindrome or not.

```
reverse([], []).  
reverse([H|T], R) :- reverse(T, RevT), conc(RevT, [H], R).  
palindrome(L) :- reverse(L, L).
```

```
?- palindrome([2,3,2]).  
true.
```

Q12 - Write a Prolog program to implement sumlist(L, S) so that S is the sum of a given list L.

```
sumlist([], 0).  
sumlist([H|T], S) :- sumlist(T, ST), S is H + ST.
```

```
?- sumlist([1,2,3],X).  
X = 6.
```

Q13 - Write a Prolog program to implement two predicates `evenlength(List)` and `oddlength(List)` so that they are true if their argument is a list of even or odd length respectively.

```
evenlength([]).  
evenlength(_|T) :- oddlength(T).  
  
oddlength([]).  
oddlength(_|T) :- evenlength(T).
```

```
?- oddlength([3,4,5]).  
true .  
  
?- oddlength([3,4,5,7]).  
false.
```

Q14 - Write a Prolog program to implement `nth_element(N, L, X)` where `N` is the desired position, `L` is a list and `X` represents the `N`th element of `L`.

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

```
?- nth_element(3, [1,2,4,5,6,7], X).  
X = 4 .
```

Q15 - Write a Prolog program to implement `maxlist(L, M)` so that `M` is the maximum number in the list.

```
maxlist([X], X).  
maxlist([H|T], M) :- maxlist(T, M1), (H > M1 -> M is H ; M is M1).
```

```
?- maxlist([1,4,3,2,6,9],X).  
X = 9 .
```

Q16 - Write a prolog program to implement `insert_nth(I, N, L, R)` that inserts an item `I` into `Nth` position of list `L` to generate a list `R`.

```
insert_nth(I, 1, L, [I|L]).  
insert_nth(I, N, [H|T], [H|R]) :- N > 1, N1 is N-1, insert_nth(I, N1, T, R).
```

```
?- insert_nth(2,1,[0,4,5,6], L).  
L = [2, 0, 4, 5, 6] .
```

Q17 - Write a Prolog program to implement `delete_nth(N, L, R)` that removes the element on `Nth` position from a list `L` to generate a list `R`.

```
delete_nth(I, [_|T], T).  
delete_nth(N, [H|T], [H|R]) :- N > 1, N1 is N-1, delete_nth(N1, T, R).
```

```
?- delete_nth(1, [1,2,3],X).  
X = [2, 3] .
```

Q18 - Write a program in PROLOG to implement merge (L1, L2, L3) where L1 is first ordered list and L2 is second ordered list and L3 represents the merged list.

```
merge([], L2, L2).  
merge(L1, [], L1).  
merge([H1|T1], [H2|T2], [H1|T3]) :- H1 <= H2, merge(T1, [H2|T2], T3).  
merge([H1|T1], [H2|T2], [H2|T3]) :- H2 < H1, merge([H1|T1], T2, T3).
```

```
?- merge([1,2,3,4],[2,3,4,5],X).  
X = [1, 2, 2, 3, 3, 4, 4, 5]
```