

**Course : B.Sc. (h) Computer Science**

**Year : III**

**Semester : VI**

**Name : Aditya Kumar College Rollno : CSC/20/10 University Rollno : 20059570008**

**Artificial Intelligence Practical File Submitted to :- Ms. Neha Kumari**

Table of Content

[1. Write a Prolog Program to calculate the sum of two numbers. 4](#_Toc133024228)

[1.1. Code 4](#_Toc133024229)

[1.2. Output 4](#_Toc133024230)

[2. Write a Prolog program to implement max(X, Y, M) so that M is the maximum of two numbers X and Y 5](#_Toc133024231)

[2.1. Code 5](#_Toc133024232)

[2.2. Output 5](#_Toc133024233)

[3. Write a Prolog Program to implement factorial (N, F) where F represents the factorial of a number N. 6](#_Toc133024234)

[3.1. Code 6](#_Toc133024235)

[3.2. Output 6](#_Toc133024236)

[4. Write a Prolog Program to implement generate\_fib(N,T) where T represents the Nth term of the Fibonacci Series. 7](#_Toc133024237)

[4.1. Code 7](#_Toc133024238)

[4.2. Output 7](#_Toc133024239)

[5. Write a Prolog program to implement GCD of two numbers. 8](#_Toc133024240)

[5.1. Code 8](#_Toc133024241)

[5.2. Output 8](#_Toc133024242)

[6. Write a Prolog Program to implement power (Num,Pow, ) : where Num is raised to the power Pow to get. 9](#_Toc133024243)

[6.1. Code 9](#_Toc133024244)

[6.2. Output 9](#_Toc133024245)

[7. Write a Prolog Program implement multi (N1, N2, R) : where N1 and N2 denotes the numbers to be multiplied and R represents the result. 10](#_Toc133024246)

[7.1. Code 10](#_Toc133024247)

[7.2. Output](#_Toc133024248) 10

[8. Write a Prolog Program to implement memb(X, L): to check whether X is a member of L or not. 11](#_Toc133024249)

[8.1. Code 11](#_Toc133024250)

[8.2. Output 11](#_Toc133024251)

[9. Write a Prolog Program implement conc (L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3. 12](#_Toc133024252)

[9.1. Code 12](#_Toc133024253)

[9.2. Output 12](#_Toc133024254)

[10. Write a Prolog Program to implement reverse (L, R) where List L is original and List R is reversed list. 13](#_Toc133024255)

[10.1. Code 13](#_Toc133024256)

[10.2. Output 13](#_Toc133024257)

[11. Write a Prolog Program to implement palindrome (L) which checks whether a list L is a palindrome or not. 14](#_Toc133024258)

[11.1. Code 14](#_Toc133024259)

[11.2. Output 14](#_Toc133024260)

[12. Write a Prolog Program implement sumlist(L, S) so that S is the sum of a given list L. 15](#_Toc133024261)

[12.1. Code 15](#_Toc133024262)

[12.2. Output 15](#_Toc133024263)

[13. Write a Prolog Program 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. 16](#_Toc133024264)

[13.1. Code 16](#_Toc133024265)

[13.2. Output 16](#_Toc133024266)

[14. Write a Prolog Program implement nth\_element (N, L, X) where N is the desired position, L is a list and X represents the Nth element of L. 17](#_Toc133024267)

[14.1. Code 17](#_Toc133024268)

[14.2. Output 17](#_Toc133024269)

[15. Write a Prolog Program to implement maxlist(L, M) so that M is the maximum number in the list. 18](#_Toc133024270)

[15.1. Code 18](#_Toc133024271)

[15.2. Output 18](#_Toc133024272)

[16. Write a Prolog Program implement insert\_nth (I, N, L, R) that inserts an item I into Nth position of list L to generate a list R. 19](#_Toc133024273)

[16.1. Code 19](#_Toc133024274)

[16.2. Output 20](#_Toc133024275)

[17. Write a Prolog Program implement delete\_nth (N, L, R) that removes the element on Nth position from a list L to generate a list R. 21](#_Toc133024276)

[17.1. Code 21](#_Toc133024277)

[17.2. Output 21](#_Toc133024278)

[18. Write a Prolog Program implement merge (L1, L2, L3) where L1 is first ordered list and L2 is second ordered list and L3 represents the merged list. 22](#_Toc133024279)

[18.1. Code 22](#_Toc133024280)

[18.2. Output](#_Toc133024281) 22

# Write a Prolog Program to calculate the sum of two numbers.

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## sum(A,B):- Z is A+B,write(Z).

## sum:-

## write('Enter first number :- '),nl,read(A),

## write('Enter second number :- '),nl,read(B),

## write('Sum of '),write(A+B),write(' is '),sum(A,B).

## Output

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

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## maxmax(A,B):- A>=B,write(A).

## max(A,B):- B>A,write(B).

## max:-

## write('Enter 1st number '),nl,read(A),

## write('Enter 2nd number '),nl,read(B),

## write('Maximum value is '),max(A,B).

## Output

# Write a Prolog Program to implement factorial (N, F) where F represents the factorial of a number N.

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## show(1,Z):- write('Factorial of the number is :- '),write(Z),nl.

## show(X,Z):-

## Temp is Z\*X,

## Y is X-1,

## show(Y,Temp).

## fact :- write('Enter a Number whose Factorial is to be Calculated :- '),read(A),

## show(A,1),nl.

## Output

# Write a Prolog Program to implement generate\_fib(N,T) where T represents the Nth term of the Fibonacci Series.

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## print\_fibo(\_A,\_B,\_C,L):- L=:=1,write('0.').

## print\_fibo(\_A,\_B,\_C,L):- L=:=0,write(' - '),nl.

## print\_fibo(A,\_B,C,\_L):- C=:=1,write(A),write('.'),nl.

## print\_fibo(A,B,C,L):- write(A),write(','),

## E is A+B,

## D is B,

## F is C-1,print\_fibo(D,E,F,L).

## 

## fibonacci:-

## write('Enter number of Elements you want from Fibonacci Series :- '),read(A),nl,

## write('Fibonacci Series :- '),print\_fibo(0,1,A,A),nl.

## Output

# Write a Prolog program to implement GCD of two numbers.

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## gcd(0,\_X):-write("GCD is 0").

## gcd(\_X,0):-write("GCD is 0").

## gcd(1,\_X):-write("GCD is 1").

## gcd(\_X,1):-write("GCD is 1").

## gcd(A,B):-startnow(A,B,2).

## confirm(0,0,\_A,\_B,C):-write("GCD is :- "),write(C).

## confirm(\_X,\_Y,A,B,C):- C<A,S is C+1,startnow(A,B,S).

## startnow(A,B,C):-X is A mod C,Y is B mod C,confirm(X,Y,A,B,C).

## gcd:-

## write("Enter 1st number(Smaller) :- "),read(A),nl,

## write("Enter 2nd number(Bigger) :- "),read(B),nl,

## write("GCD is :- "),gcd(A,B),nl.

## Output

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

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## calc\_power(A,B):- Z is A^B,write(Z),nl.

## base:-

## write('Enter Base Value :- '),read(Num),nl,

## write('Enter Power Value :- '),read(Pow),nl,

## write('Resultant Value :- '),calc\_power(Num,Pow),nl.

.

## Output

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

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## mul(A,B,\_Z):- Temp is A\*B,write(Temp),nl.

## mul:-

## write('Enter 1st number :- '),read(A),nl,

## write('Enter 2nd number :- '),read(B),nl,

## write('Mulitplication Result :- '),mul(A,B,1).

## Output

## 

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

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## list([1,2,3,4,5,6,7,8,9,10]).

## check(X):-list(L),member(X,L).

## findinlist:-

## write("Enter Number to check in Given List (1,2,3,4,5,6,7,8,9,10) :- "),read(A),check(A),nl.

## Output

## 

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

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

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

## conc([H|T], L2, [H|L3]):- conc(T, L2, L3).

## concat :-

## write('Enter first list: '),

## read(List1),

## write('Enter second list: '),

## read(List2),

## conc(List1, List2, Result),

## write('Concatenated list: '),

## write(Result).

## Output

## 

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

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## reverse([], []).

## reverse([H|T], R) :- reverse(T, RevT), append(RevT, [H], R).

## reverse\_list :-

## write('Enter a list: '),

## read(List),

## reverse(List, RevList),

## write('Original List: '), write(List), nl,

## write('Reversed List: '), write(RevList), nl.

## Output

## 

# Write a Prolog Program to implement palindrome (L) which checks whether a list L is a palindrome or not.

## Code

% CSC/20/10 Aditya Kumar University\_Roll\_No:- 20059570008

check([H|T],[X|Y]):-H=:=X,check(T,Y).

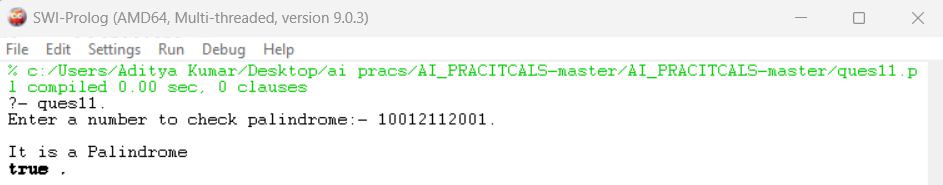
check([\_H|\_T],[\_X|\_Y]):-write("Not a Plaindrome").

check([],[]):-write("Palindrome").

palindrome([H|T]):- reverse([H|T],[X|Y]),check([H|T],[X|Y]).

ques11:- write("Enter a List :- "),read(L),nl,palindrome(L).

## Output



# Write a Prolog Program implement sumlist(L, S) so that S is the sum of a given list L.

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## sumlist(List) :- sumlist(List, 0).

## sumlist([H|T], S) :-

## C is H + S,

## sumlist(T, C).

## sumlist([], S) :-

## write("Sum of List is: "),

## write(S),

## nl.

## main :-

## write("Enter a list of numbers (separated by commas): "),

## read(List),

## sumlist(List).

## Output

## 

# Write a Prolog Program 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.

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## len([], 0).

## len([\_H | T], A) :- len(T, A1), A is A1 + 1.

## evenlength :-

## write("Enter list elements: "), read(L),

## len(L, Len), S is Len mod 2, checkeven(S).

## checkeven(X) :- X =:= 0, write("List is of Even Length :- True").

## checkeven(\_) :- write("List is of Even Length :- False").

## oddlength :-

## write("Enter list elements: "), read(L),

## len(L, Len), S is Len mod 2, checkodd(S).

## checkodd(X) :- X =:= 1, write("List is of Odd Length :- True").

## checkodd(\_) :- write("List is of Odd Length :- False").

## Output

# Write a Prolog Program implement nth\_element (N, L, X) where N is the desired position, L is a list and X represents the Nth element of L.

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## nth\_element(1, [H|\_], H).

## nth\_element(N, [\_|T], X) :- N > 1, N1 is N-1, nth\_element(N1, T, X).

## main :- write('Enter a list: '),

## read(List),

## write('Enter the position: '),

## read(Position),

## nth\_element(Position, List, X),

## write('Element at position '), write(Position), write(' is '), write(X).

## ?- main.

## Output

## 

# Write a Prolog Program to implement maxlist(L, M) so that M is the maximum number in the list.

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## max([H|T],M):- H>M,max(T,H).

## max([H|T],M):- H=<M,max(T,M).

## max([],M):- M>=0,write("Greatest number is :- "),write(M),nl.

## maxList([]):-write("List can't be Empty").

## maxList([H|T]):- M is H,max(T,M),nl.

## get\_list(List) :-

## write('Enter a list of numbers: '),

## read(List).

## :- initialization(main).

## main :-

## get\_list(List),

## maxList(List).

## Output

## 

# Write a Prolog Program implement insert\_nth (I, N, L, R) that inserts an item I into Nth position of list L to generate a list R.

## Code

## % CSC/20/10 Aditya Kumar University\_roll\_no :- 20059570008

## 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).

## take\_input(List) :-

## write('Enter a list: '),

## read(List).

## take\_input\_element(E) :-

## write('Enter the element to insert: '),

## read(E).

## take\_input\_position(P) :-

## write('Enter the position to insert: '),

## read(P).

## insert :-

## take\_input(List),

## take\_input\_element(E),

## take\_input\_position(P),

## insert\_nth(E, P, List, Result),

## write('Original list: '),

## write(List),

## nl,

## write('Element inserted list: '),

## write(Result).

## Output

## 

# Write a Prolog Program implement delete\_nth (N, L, R) that removes the element on Nth position from a list L to generate a list R.

## Code

% CSC/20/10 Aditya Kumar Univ\_Roll\_No:20059570008

% showing list

show([\_H|T]):- nl,write("List is :- "),write(T).

% calculating length of list

len([],0).

len([\_H|T],A):-len(T,A1),A is A1+1.

removelast([H|T]):- reverse([H|T],[\_H1|T1]),reverse(T1,[\_X|Y]),write("List is :- "),write(Y).

delete\_nth(\_N,[]):-write(" Nothing to Delete : List is Empty (UnderFlow).").

delete\_nth(N,[H|T]):- len([H|T],L),delete\_nth(N,[H|T],1,[\_],L).

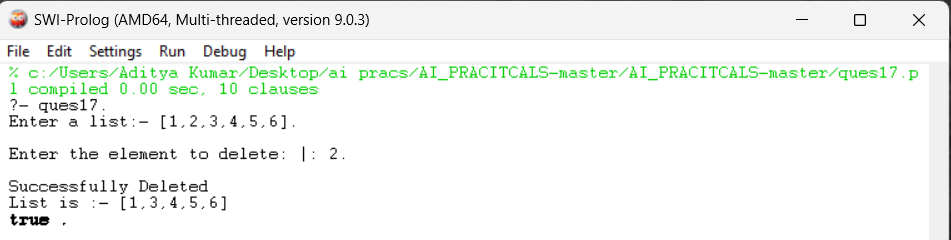
delete\_nth(N,[\_H|T],C,[H1|T1],\_L):- C=:=N,append([H1|T1],T,V),write("Successfully Deleted"),show(V).

delete\_nth(N,[H|T],C,[H1|T1],L):- C<N,S is C+1,append([H1|T1],[H],V),delete\_nth(N,T,S,V,L).

delete\_nth(N,[H|T],\_C,[\_H1|\_T1],L):- N=:=L,removelast([H|T]).

ques17:- write("Enter Index to be Deleted :- "),read(P),nl,write("Enter List :- "),read(L),nl,delete\_nth(P,L).

## Output



# Write a Prolog Program implement merge (L1, L2, L3) where L1 is first ordered list and L2 is second ordered list and L3 represents the merged list.

## Code

% CSC/20/10 Aditya Kumar Univ\_Roll\_No:- 20059570008

show([\_H|T]):- write("List is :- "),write(T),nl.

merge([H|T],[X|Y],[H1|T1]):-H>X,append([H1|T1],[X],R),merge([H|T],Y,R).

merge([H|T],[X|Y],[H1|T1]):-X>H,append([H1|T1],[H],R),merge(T,[X|Y],R).

merge([H|T],[X|Y],[]):-H>X,R is [X],merge([H|T],Y,R).

merge([H|T],[X|Y],[]):-X>H,R is [H],merge(T,[X|Y],R).

merge([],[X|Y],[H1|T1]):- append([H1|T1],[X|Y],R),show(R).

merge([X|Y],[],[H1|T1]):- append([H1|T1],[X|Y],R),show(R).

ques18:- write("Enter List 1 :- "),read(L1),nl,write("Enter List 2 :- "),read(L2),nl,merge(L1,L2,Temp),nl.

## Output

## 