**Artificial Intelligence Practical File**

**2021-2022**

**VI Semester**

**Name: Akanksha goel**

**Examination Roll No.: 19035570011**

**College Roll No.: 0726**

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

**Section.: A**

**13. 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.**

CODE:-

evenlength([]).

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

oddlength([\_]).

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

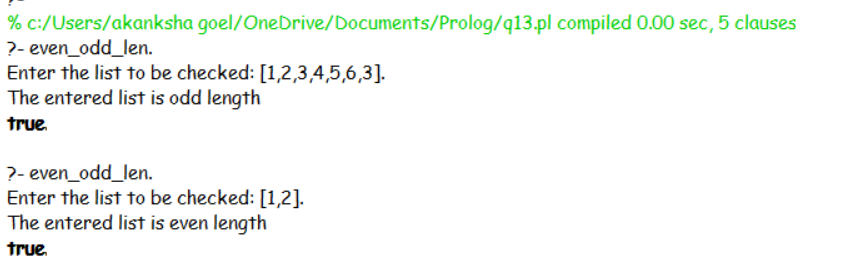
even\_odd\_len:-

write("Enter the list to be checked: "),read(L),

(evenlength(L)->write("The entered list is even length");

write("The entered list is odd length")),!.

OUTPUT



**14. 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 Nth element of L.**

CODE:-

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

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

nth\_element:-

write("Enter the list :: "),read(L),

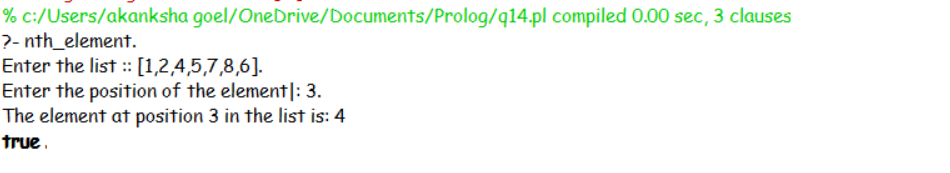
write("Enter the position of the element"),read(N),

nth\_element(N,L,X),

write("The element at position "),write(N),write(" in the list is: "),

write(X).

OUTPUT



**15. Write a Prolog program to implement maxlist(L, M) so that M is the**

**maximum number in the list.**

CODE:-

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

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

maxlist([],0):-!.

maxlist([M],M):-!.

maxlist([H|T],M):-maxlist(T,M1),max(H,M1,M),!.

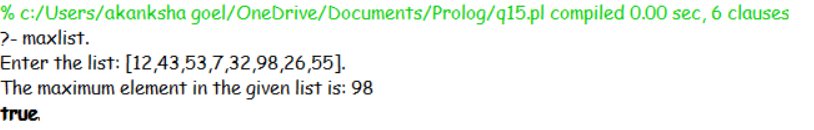
maxlist:-

write("Enter the list: "),read(L),

maxlist(L,M),

write("The maximum element in the given list is: "),write(M),!.

OUTPUT



**16. 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.**

CODE:-

insert\_nth(I,L,1,[I|L]).

insert\_nth(I,[H|L],N,[H|R]):- N1 is N-1,insert\_nth(I,L,N1,R).

insert\_nth:-

write("Enter the list: "),read(L),

write("Enter the position of the element to be inserted:

"),read(N),

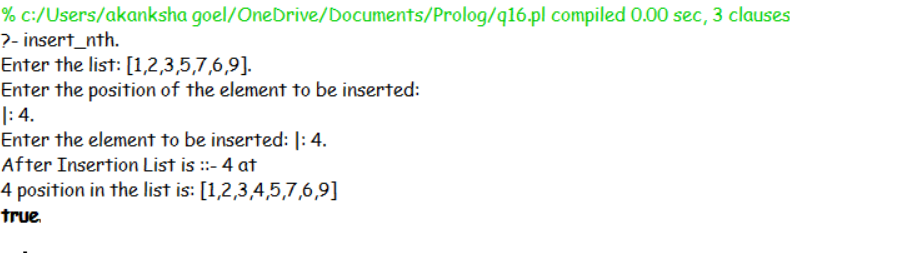
write("Enter the element to be inserted: "),read(I),

insert\_nth(I,L,N,R),

write("After Insertion List is ::- "),write(I),write(" at

"),write(N),write(" position in the list is: "),write(R),!.

OUTPUT



**17. 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.**

CODE:-

del\_nth([\_|L],1,L).

del\_nth([H|L],N,[H|R]):- N1 is N-1, del\_nth(L,N1,R).

del\_nth:-

write("Enter the list: "),read(L),

write("Enter the position of the element to be deleted: "),read(N),

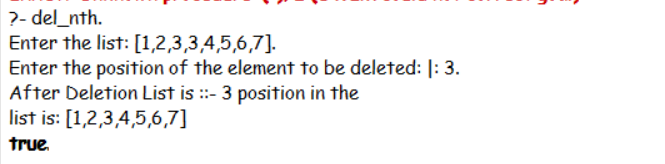
del\_nth(L,N,R),

write("After Deletion List is ::- "),write(N),write(" position in the

list is: "),

write(R),!.

OUTPUT



**18. 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.**

CODE:-

merge([],[],[]).

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]):- merge([H1|T1], T2, T3).

merge:-

write("Enter the first ordered list: "),read(L1),

write("Enter the second ordered list: "),read(L2),

merge(L1,L2,L3),

write("List After Merging is ::- "),write(L3),!.

OUTPUT

