



Atma Ram Sanatan Dharma College
University of Delhi



Artificial Intelligence
Practical File
Paper Code:- (BHCS13)

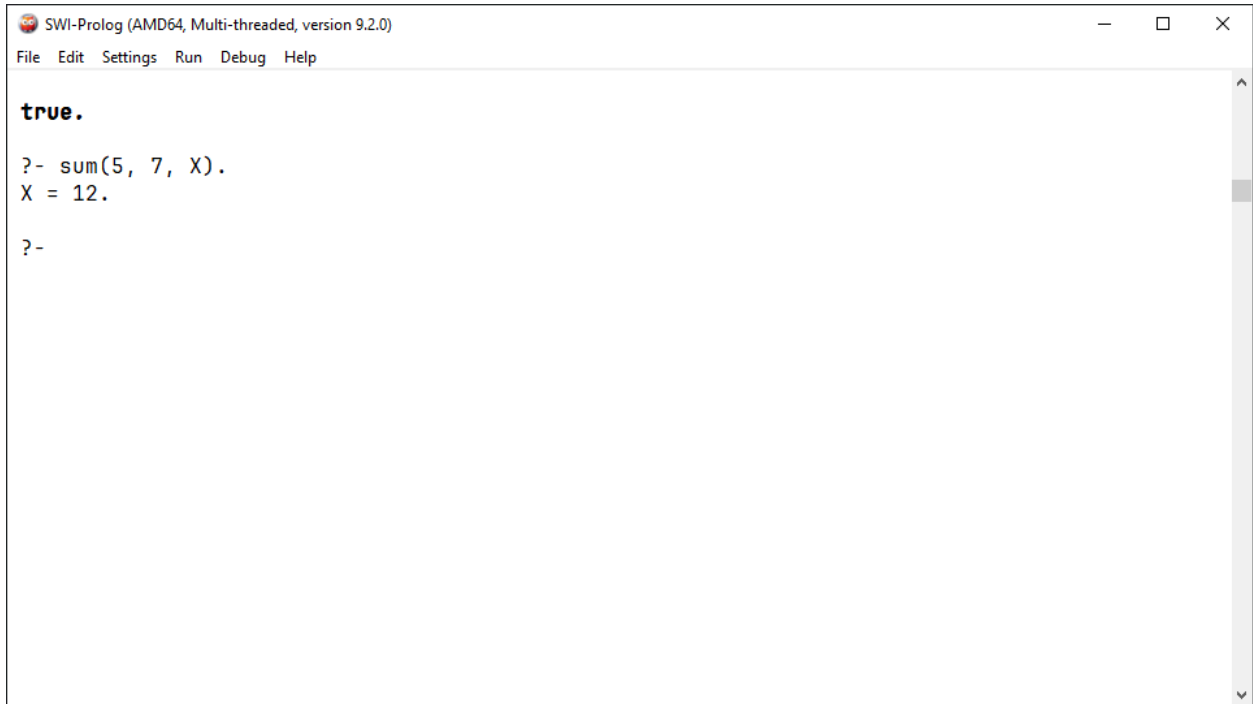
Submitted By :
Chirag Wadhwa
College Roll No: 21/18080
B.Sc (Hons) Computer Science

Submitted To :
Dr. Parul Jain
Department of Computer Science

INDEX

- Q1. Write a prolog program to calculate the sum of two numbers.
- Q2. Write a Prolog program to implement `max(X, Y, M)` so that M is the maximum of two numbers X and Y.
- Q3. Write a program in PROLOG to implement factorial (N, F) where F represents the factorial of a number N.
- Q4. Write a program in PROLOG to implement `generate_fib(N,T)` where T represents the Nth term of the fibonacci series.
- Q5. Write a Prolog program to implement GCD of two numbers.
- Q6. Write a Prolog program to implement `power (Num,Pow, Ans)` : where Num is raised to the power Pow to get Ans.
- Q7. Prolog program to implement `multi (N1, N2, R)` : where N1 and N2 denotes the numbers to be multiplied and R represents the result.
- Q8. Write a prolog program to implement `memb(X,L)`: to check whether X is a member of L or not.
- Q9. Write a prolog program to implement `conc(L1,L2,L3)` where L2 is the list to be appended with L1 to get the resulting list in L3.
- Q10. Write a prolog program to implement `reverse(L,R)` where List L is original and list R is reversed list.
- Q11. Write a programming prolog to implement `palindrome(L)` which checks whether a list L is palindrome or not.
- Q12. Write a prolog program to implement `sumlist(L,S)` so that S is the sum of a given list L.
- 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.
- 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 Nth element of L.
- Q15. Write a Prolog program to implement `maxlist(L, M)` so that M is the maximum number in the list.
- Q16. Write a prolog program to implement `insert_nth (I, N, L, R)` that inserts an item I into the Nth position of list L to generate a list R.
- 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.
- 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.

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

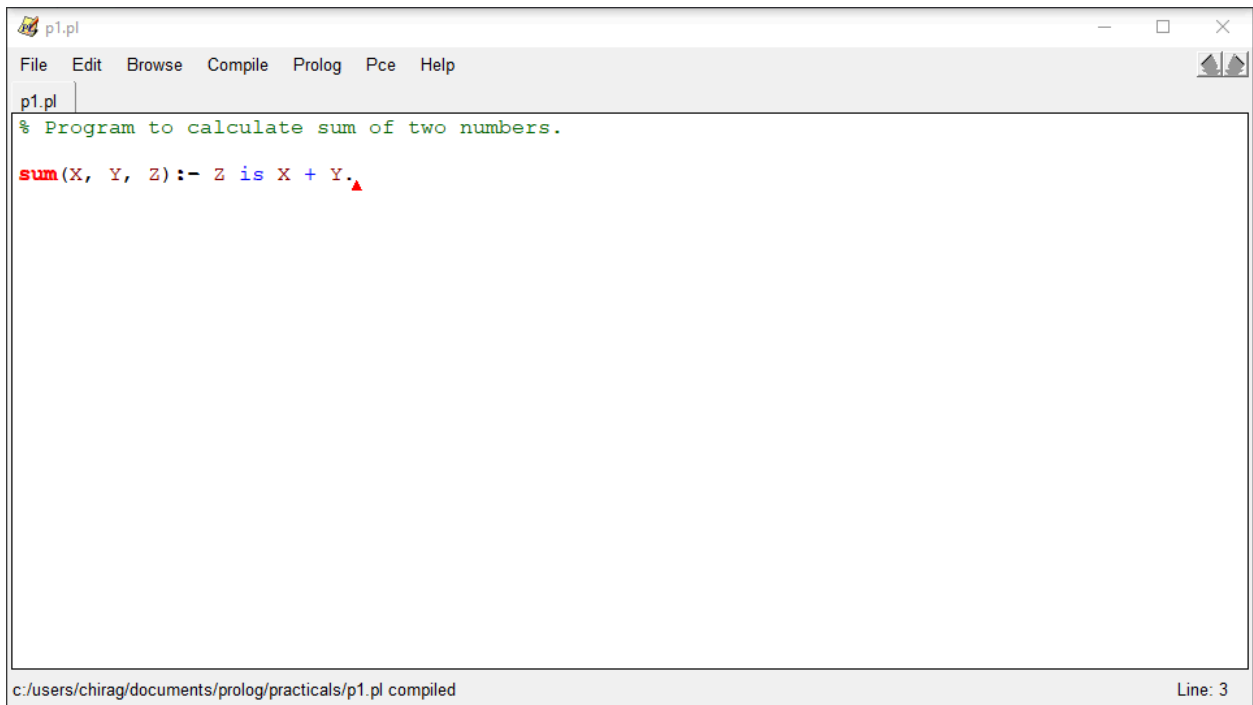


```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.0)
File Edit Settings Run Debug Help

true.

?- sum(5, 7, X).
X = 12.

?-
```

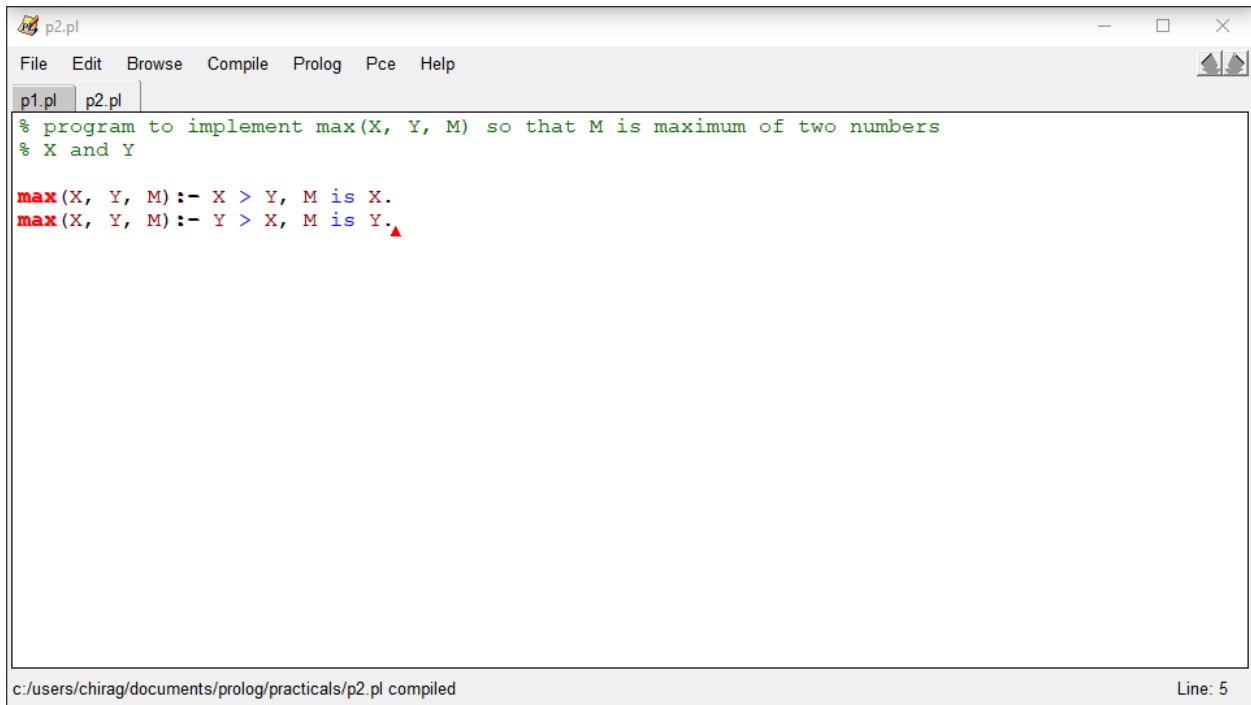


```
p1.pl
File Edit Browse Compile Prolog Pce Help

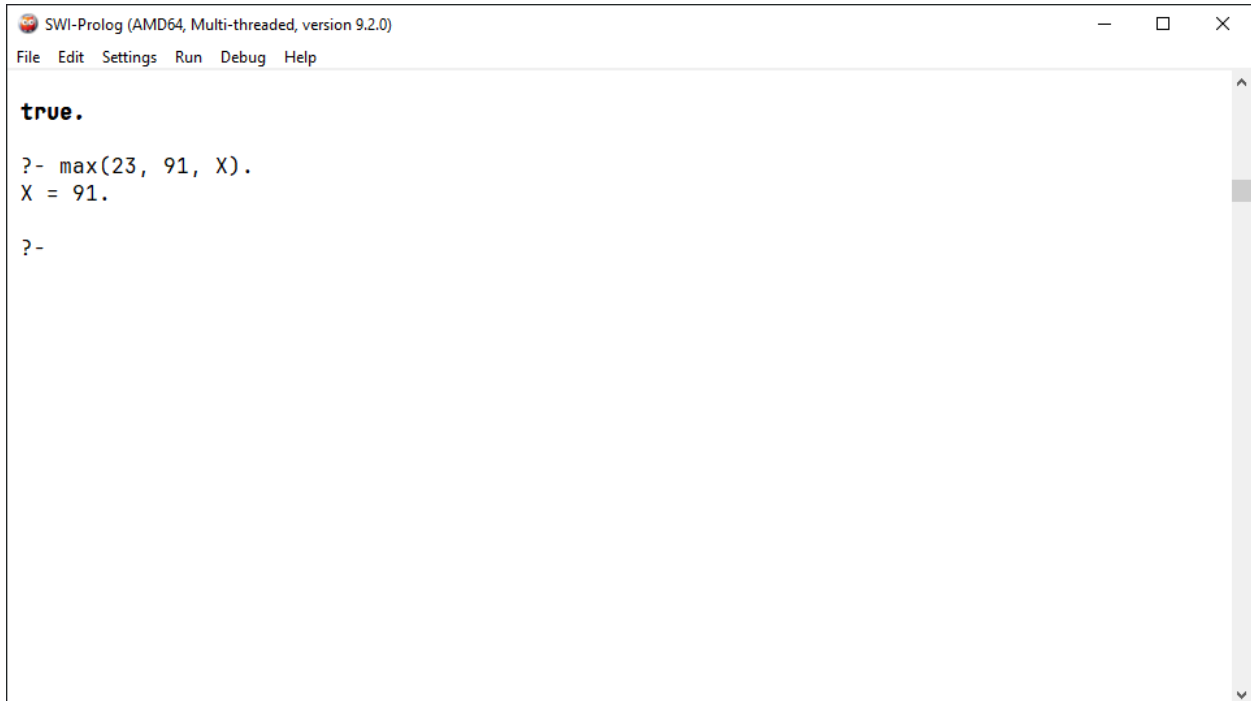
p1.pl
% Program to calculate sum of two numbers.
sum(X, Y, Z) :- Z is X + Y.

c:/users/chirag/documents/prolog/practicals/p1.pl compiled
Line: 3
```

Q2. Write a Prolog program to implement $\text{max}(X, Y, M)$ so that M is the maximum of two numbers X and Y .

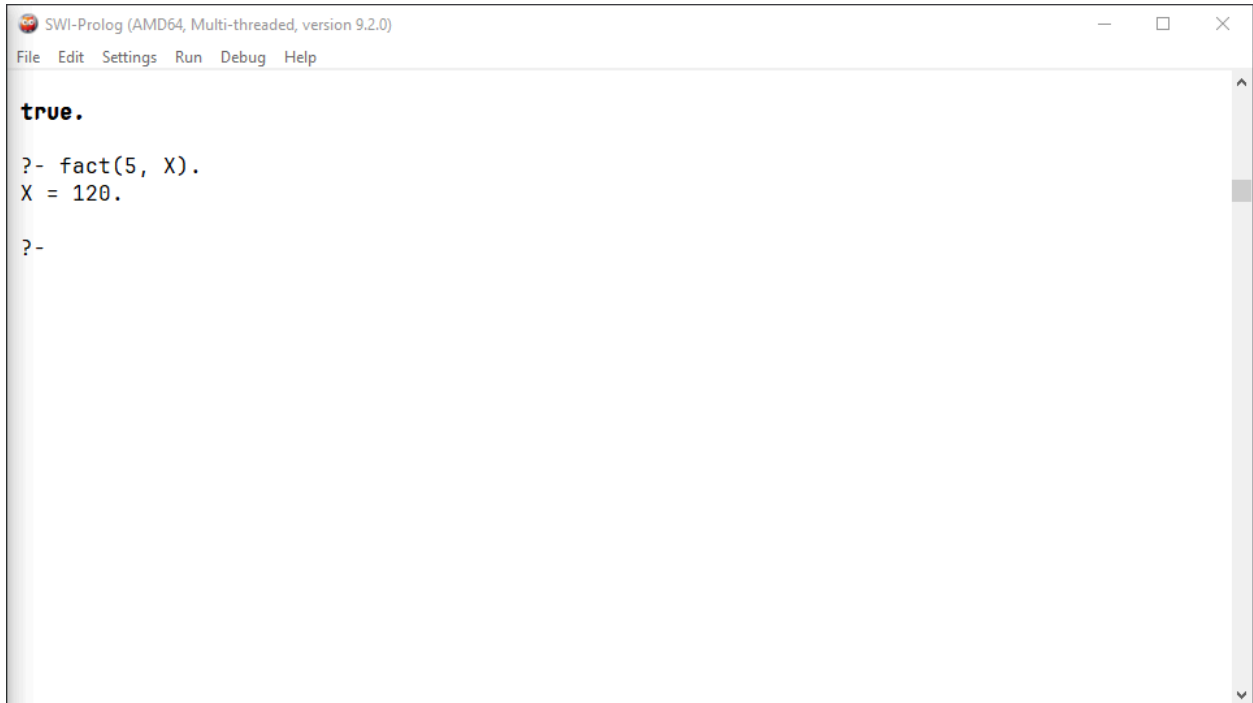


```
p2.pl
File Edit Browse Compile Prolog Pce Help
p1.pl p2.pl
% program to implement max(X, Y, M) so that M is maximum of two numbers
% X and Y
max(X, Y, M) :- X > Y, M is X.
max(X, Y, M) :- Y > X, M is Y.
c:/users/chirag/documents/prolog/practicals/p2.pl compiled
Line: 5
```



```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.0)
File Edit Settings Run Debug Help
true.
?- max(23, 91, X).
X = 91.
?-
```

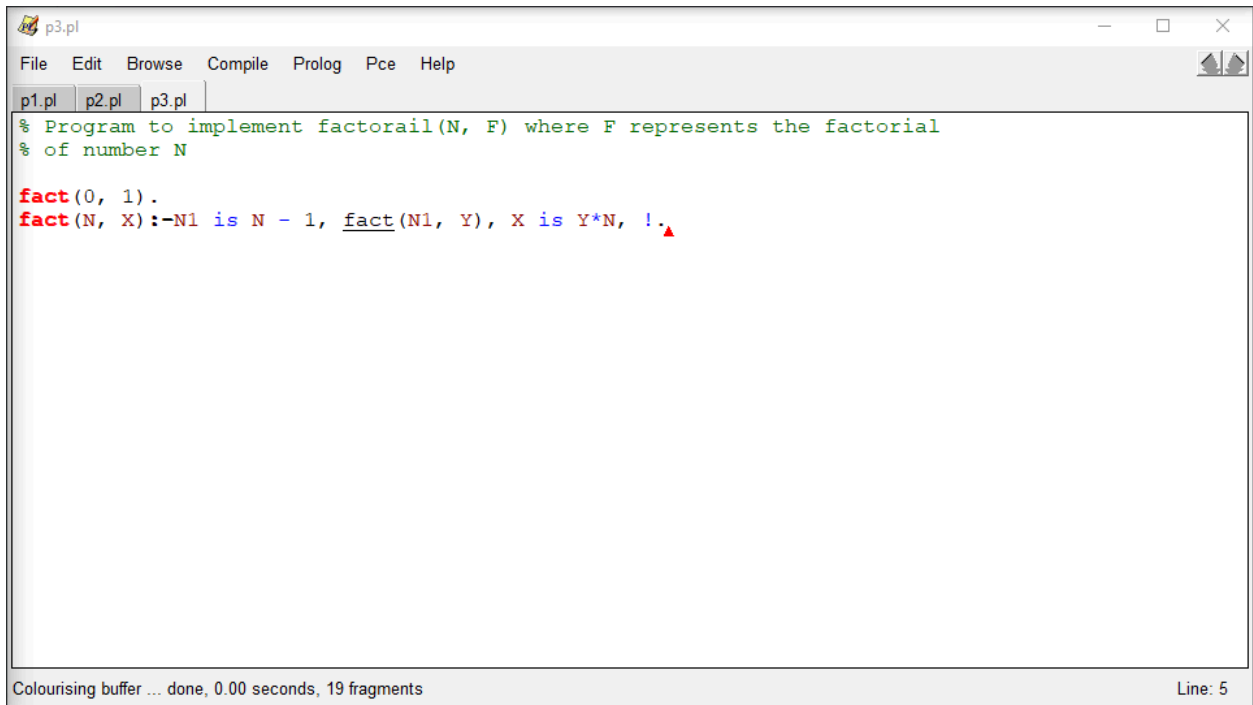
Q3. Write a program in PROLOG to implement factorial (N, F) where F represents the factorial of a number N.



SWI-Prolog (AMD64, Multi-threaded, version 9.2.0)

File Edit Settings Run Debug Help

```
true.  
  
?- fact(5, X).  
X = 120.  
  
?-
```



p3.pl

File Edit Browse Compile Prolog Pce Help

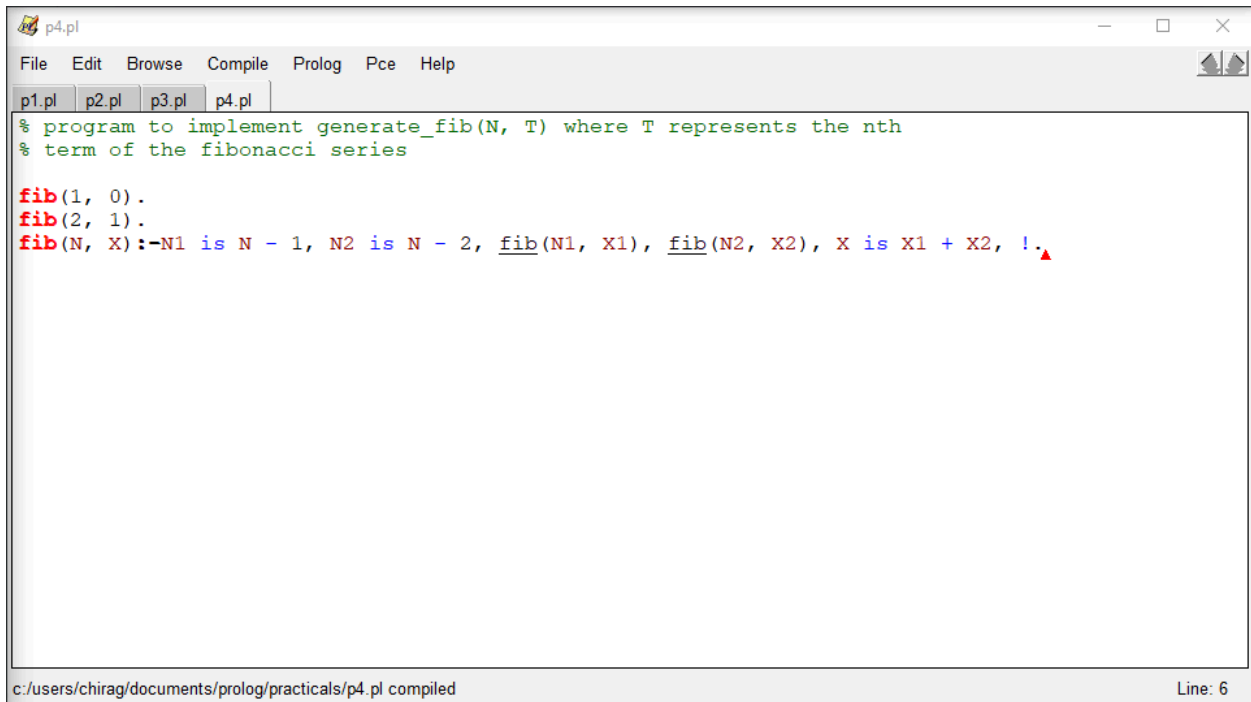
p1.pl p2.pl p3.pl

```
% Program to implement factorial(N, F) where F represents the factorial  
% of number N  
  
fact(0, 1).  
fact(N, X):-N1 is N - 1, fact(N1, Y), X is Y*N, !.
```

Colourising buffer ... done, 0.00 seconds, 19 fragments

Line: 5

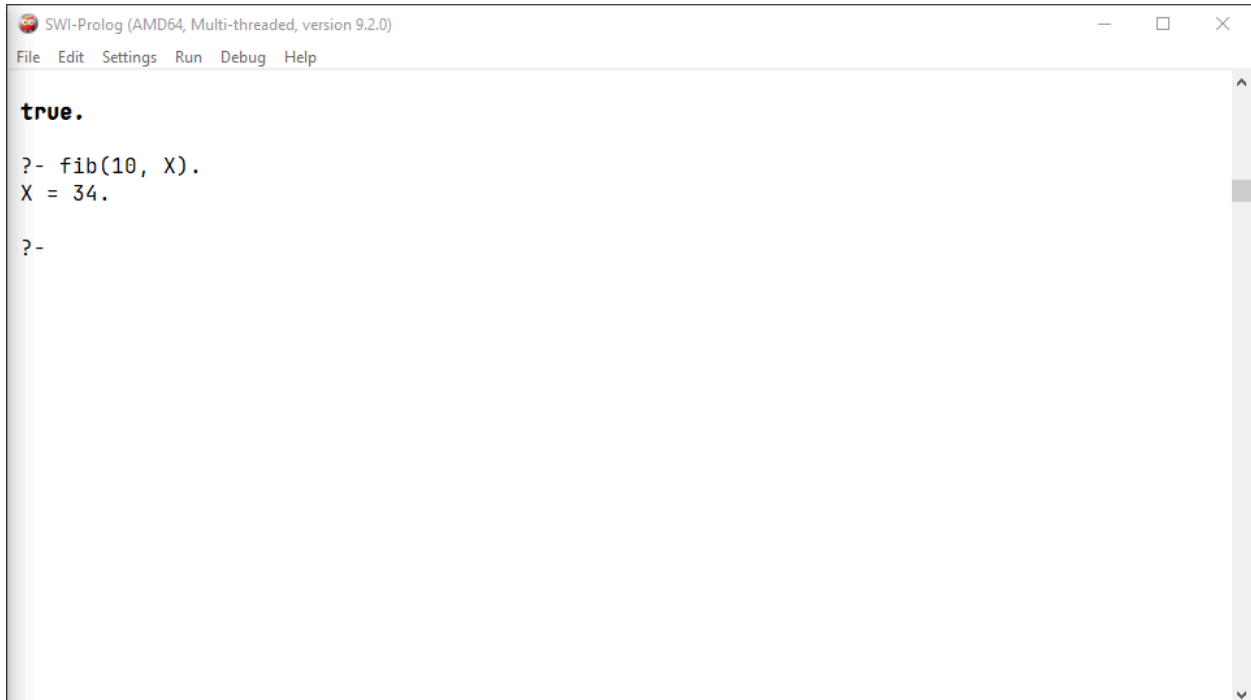
Q4. Write a program in PROLOG to implement generate_fib(N,T) where T represents the Nth term of the fibonacci series.



```
p4.pl
File Edit Browse Compile Prolog Pce Help
p1.pl p2.pl p3.pl p4.pl
% program 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, !.
```

c:/users/chirag/documents/prolog/practicals/p4.pl compiled Line: 6



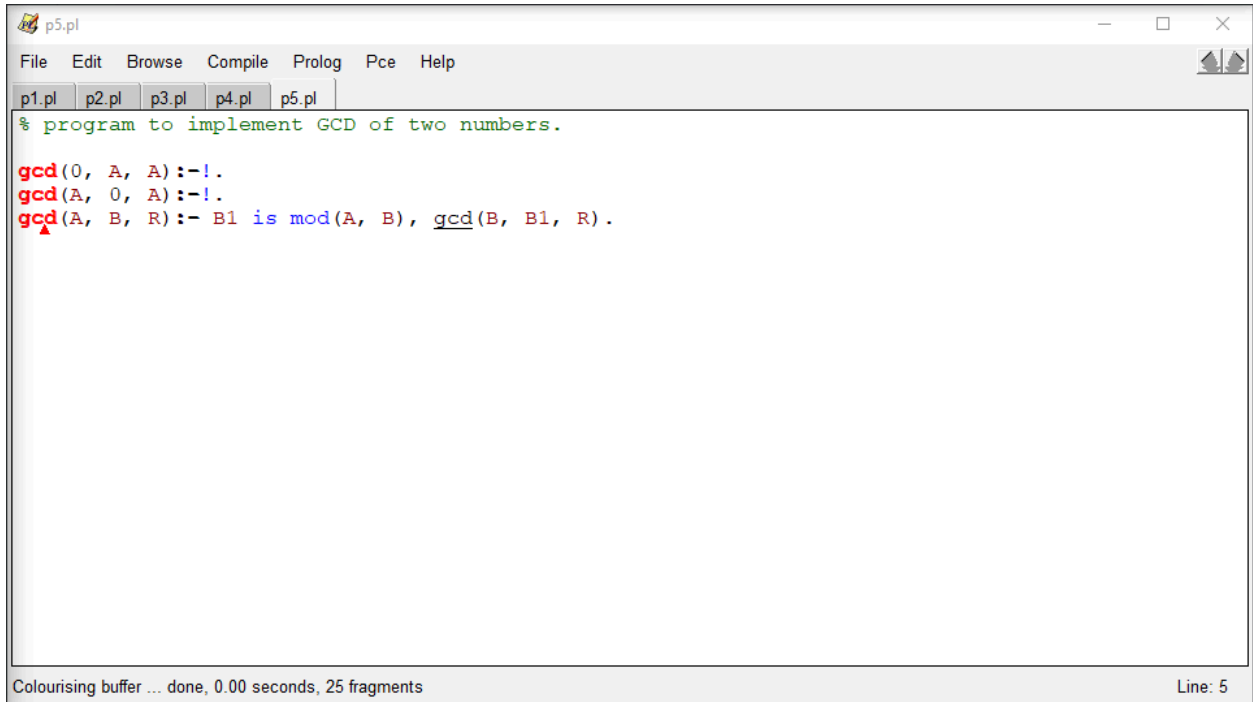
```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.0)
File Edit Settings Run Debug Help

true.

?- fib(10, X).
X = 34.

?-
```

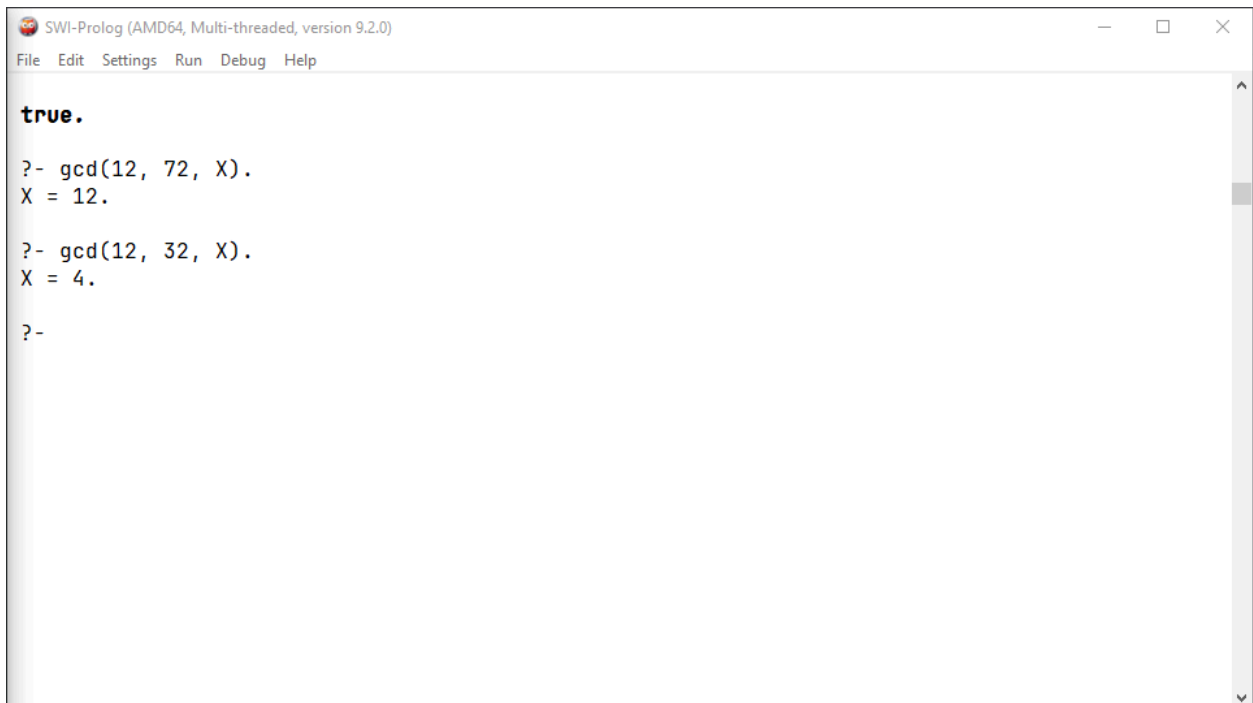
Q5. Write a Prolog program to implement GCD of two numbers.



The screenshot shows a Prolog editor window titled 'p5.pl'. The menu bar includes 'File', 'Edit', 'Browse', 'Compile', 'Prolog', 'Pce', and 'Help'. Below the menu bar, there are tabs for 'p1.pl', 'p2.pl', 'p3.pl', 'p4.pl', and 'p5.pl'. The main text area contains the following Prolog code:

```
% 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).
```

At the bottom of the window, a status bar indicates 'Colourising buffer ... done, 0.00 seconds, 25 fragments' on the left and 'Line: 5' on the right.

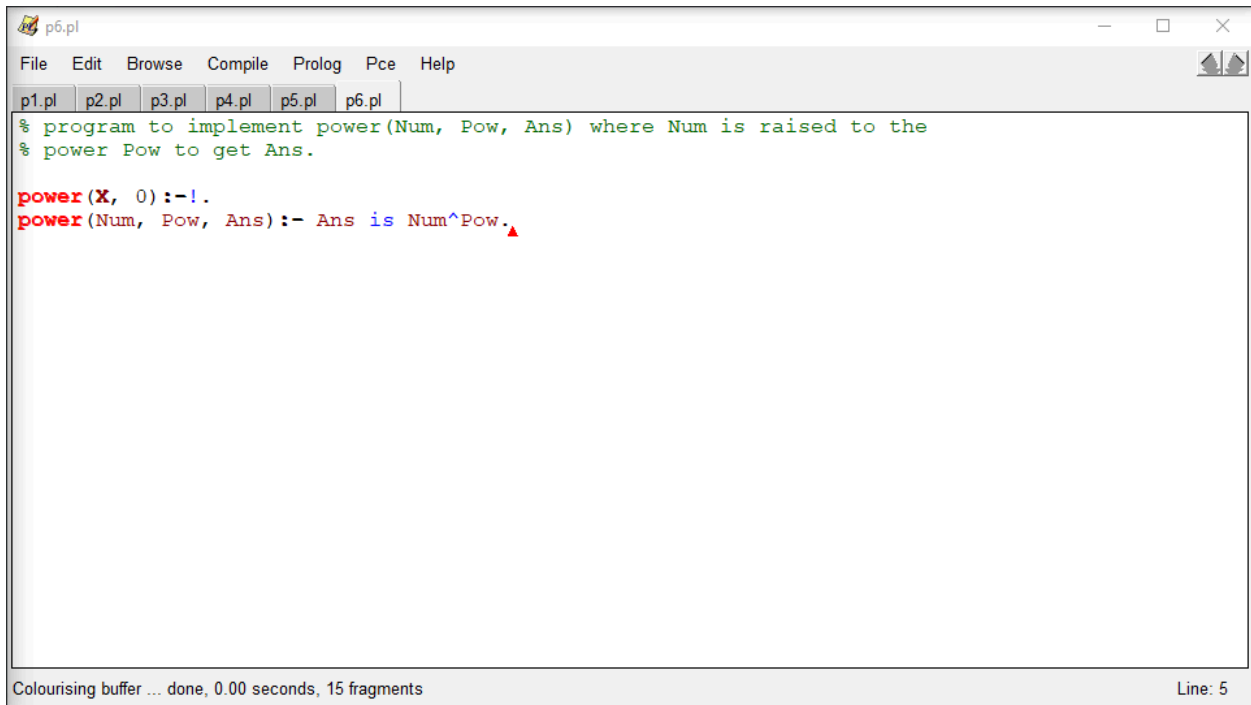


The screenshot shows a SWI-Prolog interpreter window titled 'SWI-Prolog (AMD64, Multi-threaded, version 9.2.0)'. The menu bar includes 'File', 'Edit', 'Settings', 'Run', 'Debug', and 'Help'. The main text area shows the following interaction:

```
true.  
?- gcd(12, 72, X).  
X = 12.  
  
?- gcd(12, 32, X).  
X = 4.  
  
?-
```

The window has a vertical scrollbar on the right side.

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

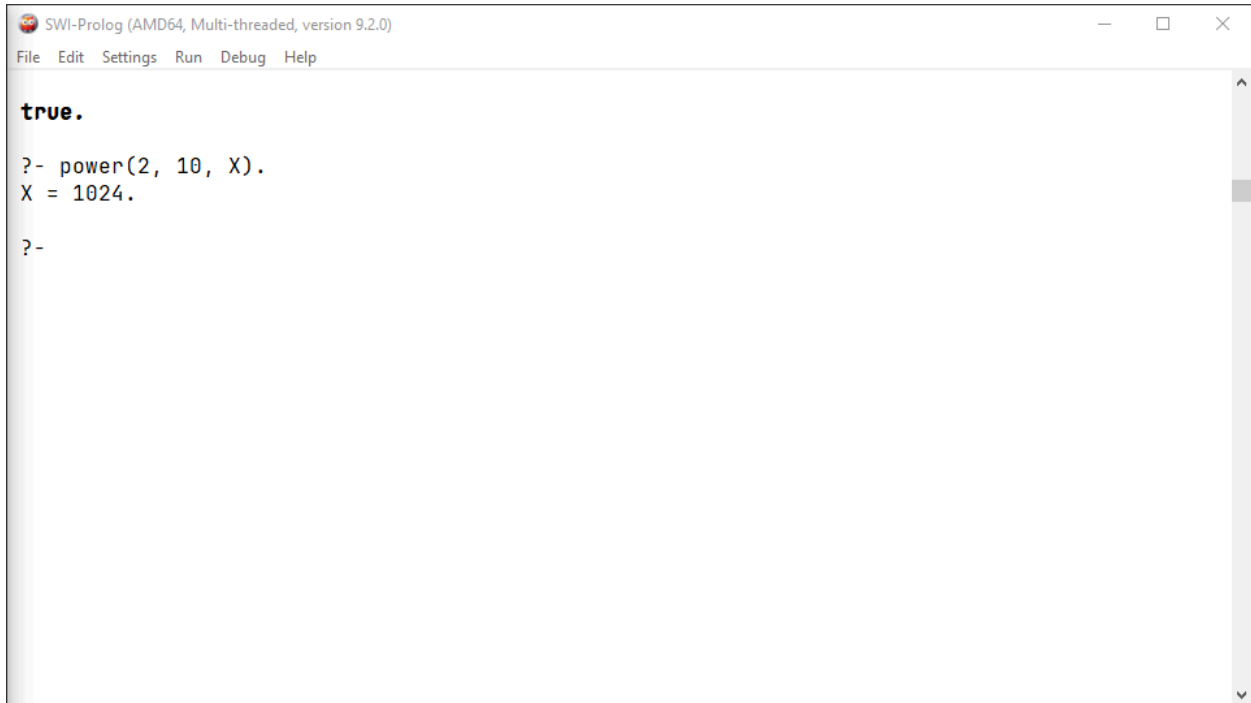


The screenshot shows a Prolog editor window titled 'p6.pl'. The menu bar includes File, Edit, Browse, Compile, Prolog, Pce, and Help. Below the menu bar are tabs for p1.pl, p2.pl, p3.pl, p4.pl, p5.pl, and p6.pl. The main text area contains the following Prolog code:

```
% 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.▲
```

The status bar at the bottom indicates 'Colourising buffer ... done, 0.00 seconds, 15 fragments' and 'Line: 5'.



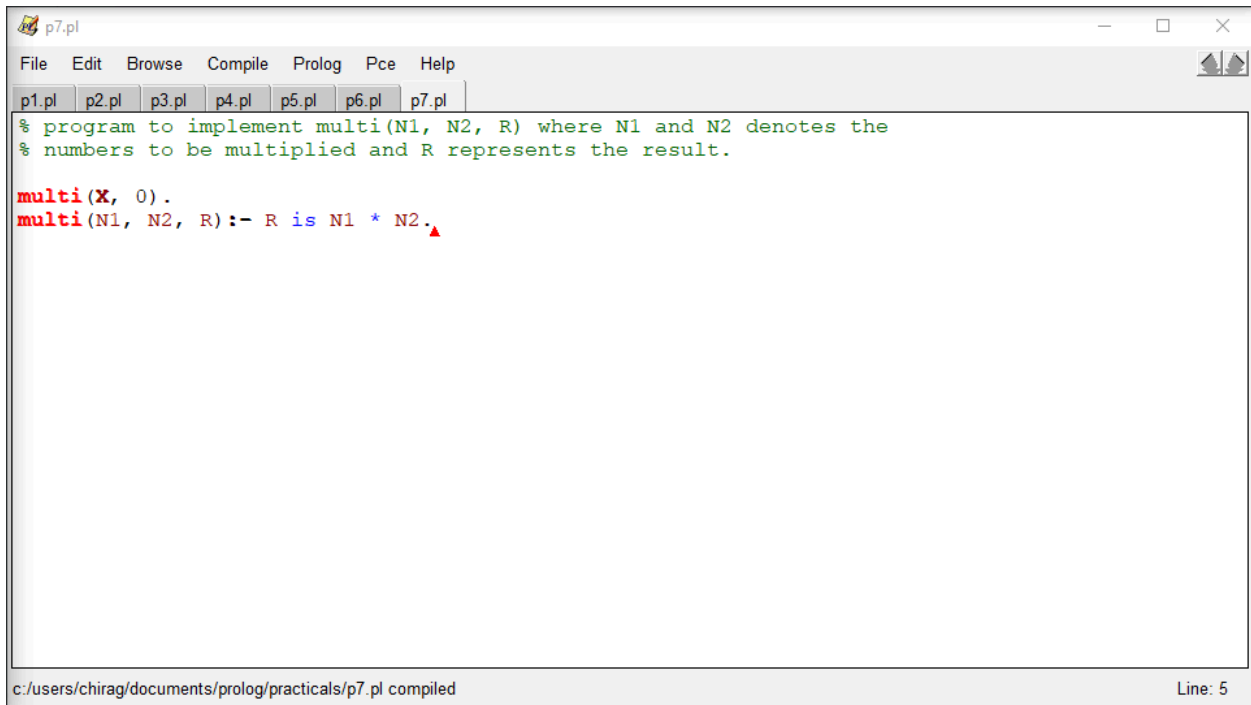
The screenshot shows the SWI-Prolog console window titled 'SWI-Prolog (AMD64, Multi-threaded, version 9.2.0)'. The menu bar includes File, Edit, Settings, Run, Debug, and Help. The console output is as follows:

```
true.

?- power(2, 10, X).
X = 1024.

?-
```


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

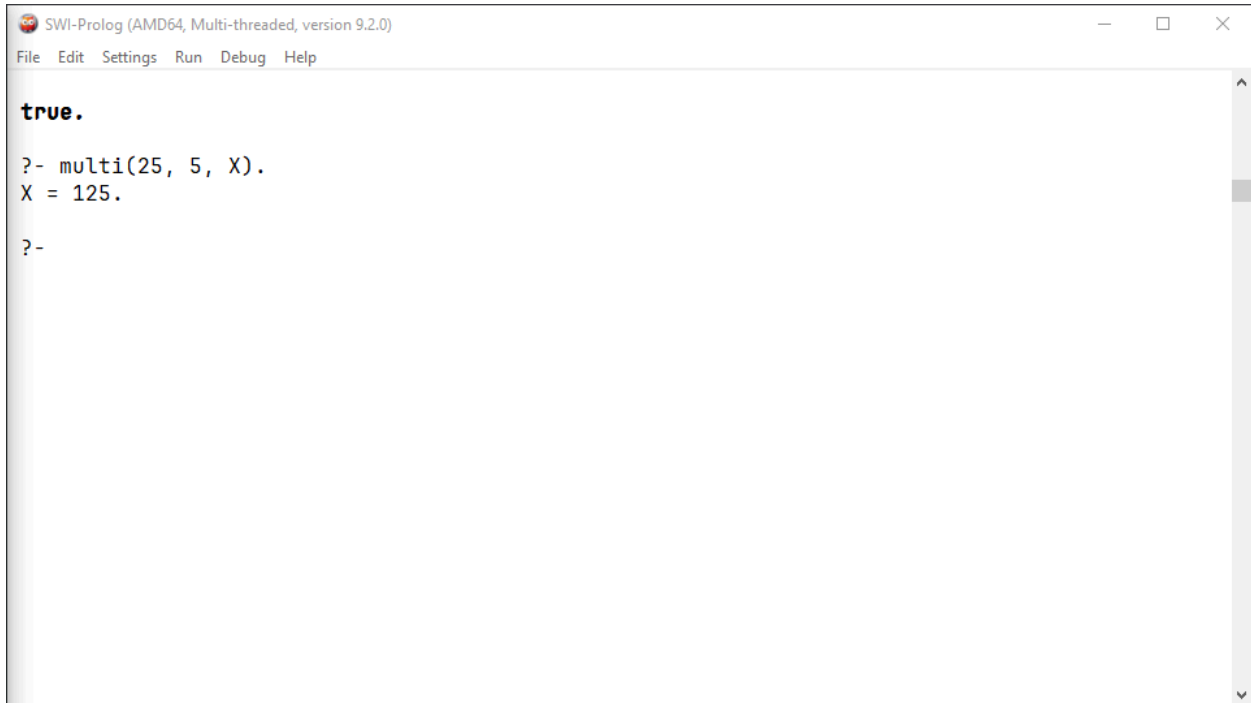


The screenshot shows a Prolog editor window titled "p7.pl". The menu bar includes File, Edit, Browse, Compile, Prolog, Pce, and Help. Below the menu bar is a tab bar with tabs for p1.pl, p2.pl, p3.pl, p4.pl, p5.pl, p6.pl, and p7.pl. The main text area contains the following Prolog code:

```
% 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.
```

The status bar at the bottom indicates "c:/users/chirag/documents/prolog/practicals/p7.pl compiled" and "Line: 5".



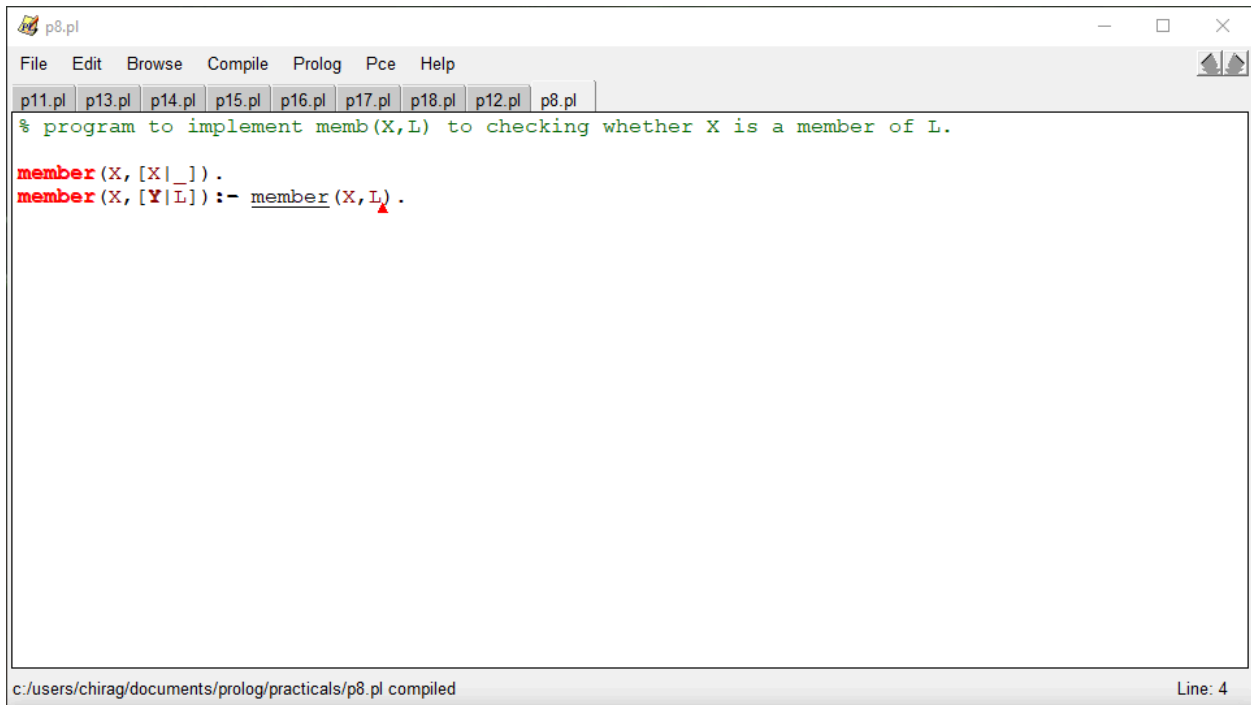
The screenshot shows the SWI-Prolog interpreter window titled "SWI-Prolog (AMD64, Multi-threaded, version 9.2.0)". The menu bar includes File, Edit, Settings, Run, Debug, and Help. The main text area shows the following interaction:

```
true.

?- multi(25, 5, X).
X = 125.

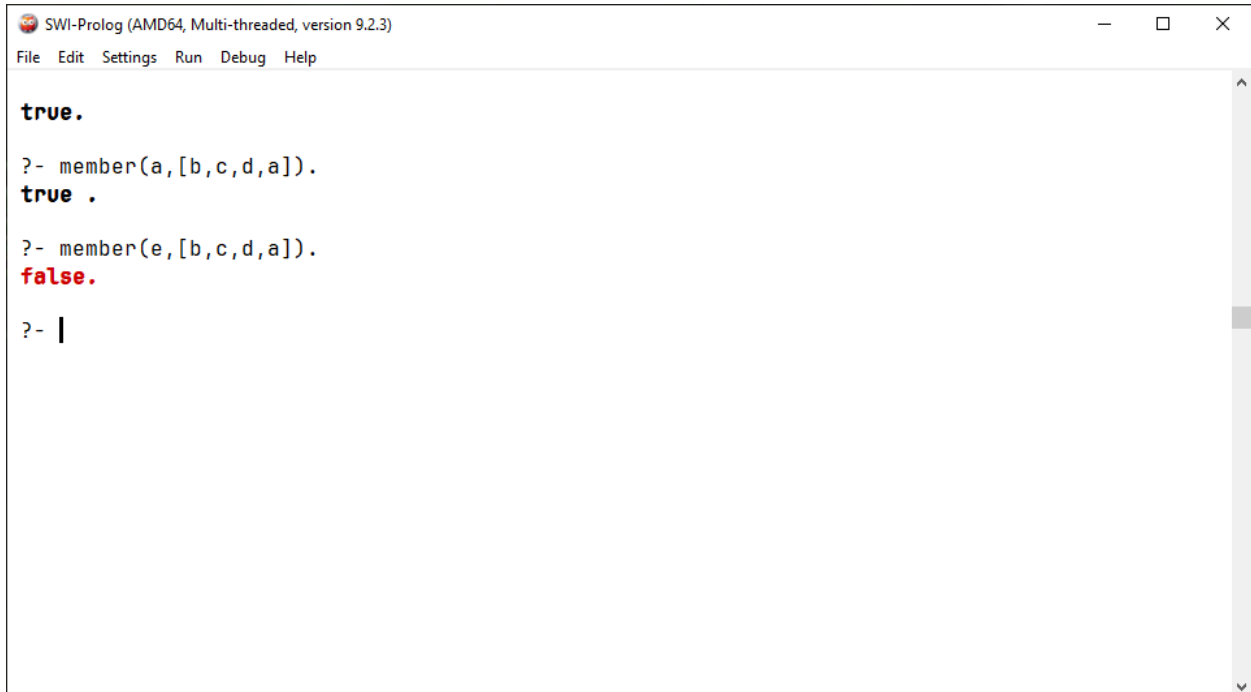
?-
```

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



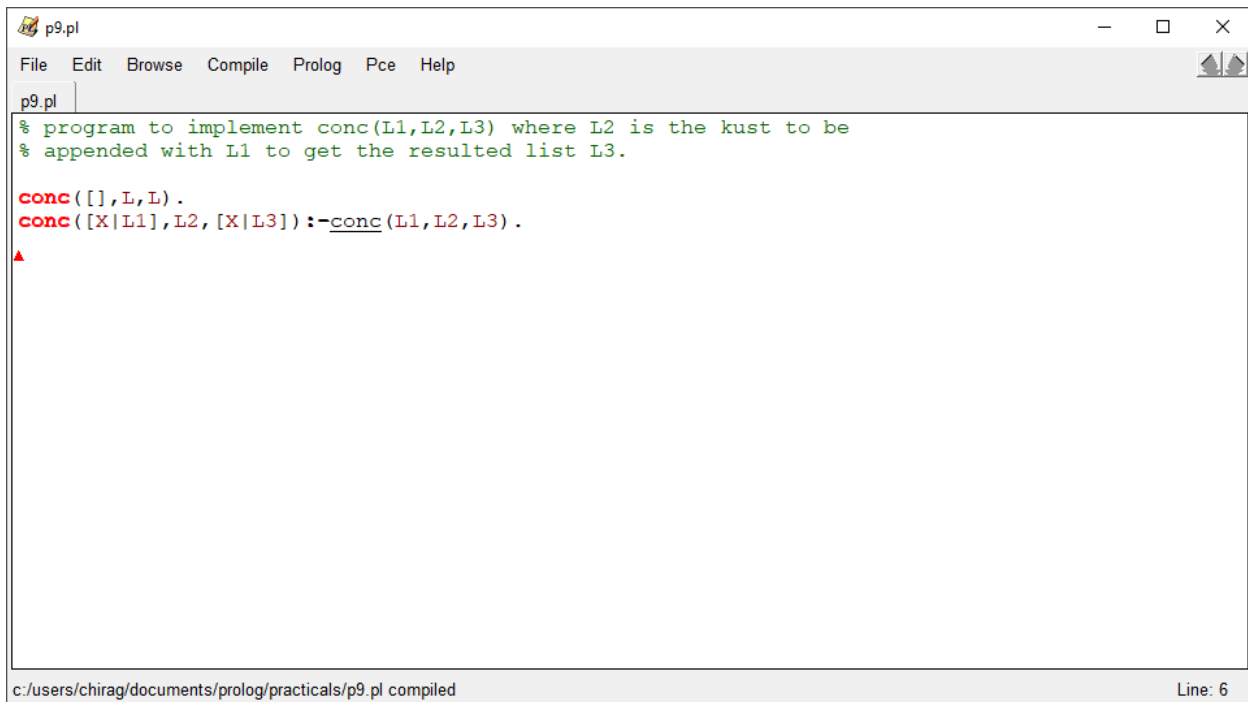
```
p8.pl
File Edit Browse Compile Prolog Pce Help
p11.pl p13.pl p14.pl p15.pl p16.pl p17.pl p18.pl p12.pl p8.pl
% program to implement memb(X,L) to checking whether X is a member of L.
member(X, [X|_]) .
member(X, [_|L]) :- member(X, L).
```

c:/users/chirag/documents/prolog/practicals/p8.pl compiled Line: 4



```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.3)
File Edit Settings Run Debug Help
true.
?- member(a,[b,c,d,a]).
true.
?- member(e,[b,c,d,a]).
false.
?- |
```

Q9. Write a prolog program to implement `conc(L1,L2,L3)` where L2 is the list to be appended with L1 to get the resulting list in L3.



```
p9.pl
File Edit Browse Compile Prolog Pce Help
p9.pl
% 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).
```

c:/users/chirag/documents/prolog/practicals/p9.pl compiled Line: 6



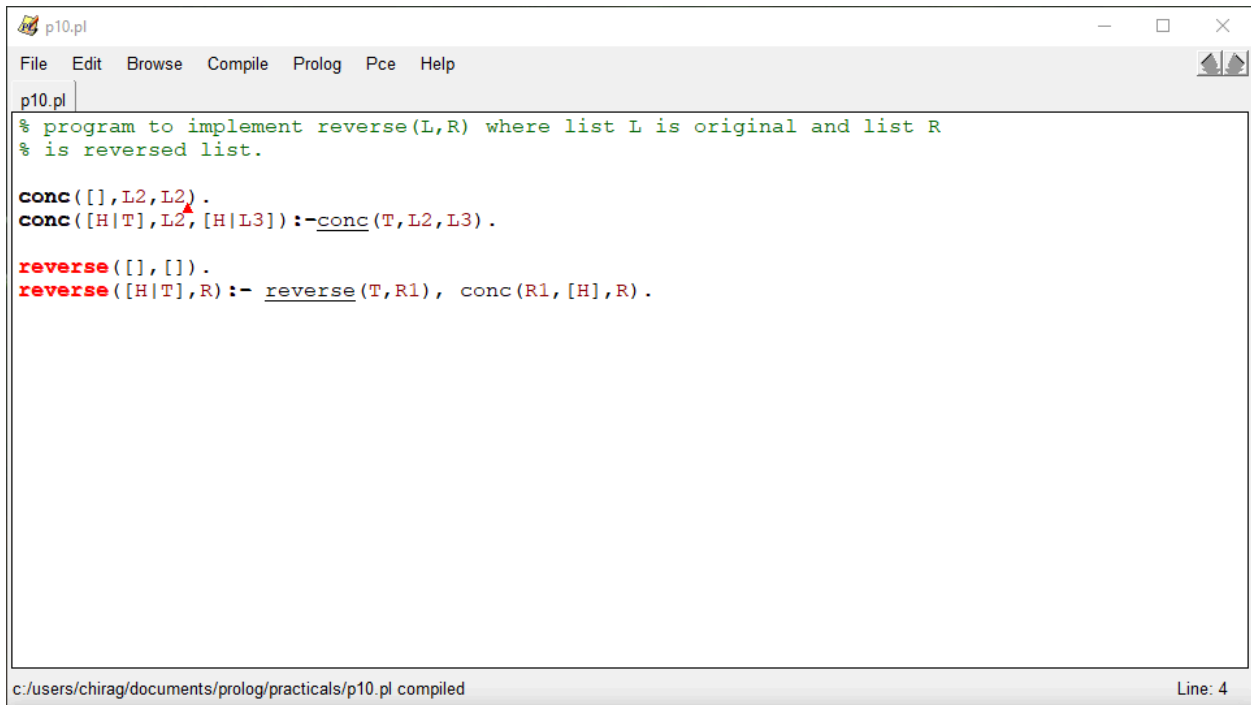
```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.3)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- conc([1,2,3],[cat,banana,rice],R).
R = [1, 2, 3, cat, banana, rice].

?-
```

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

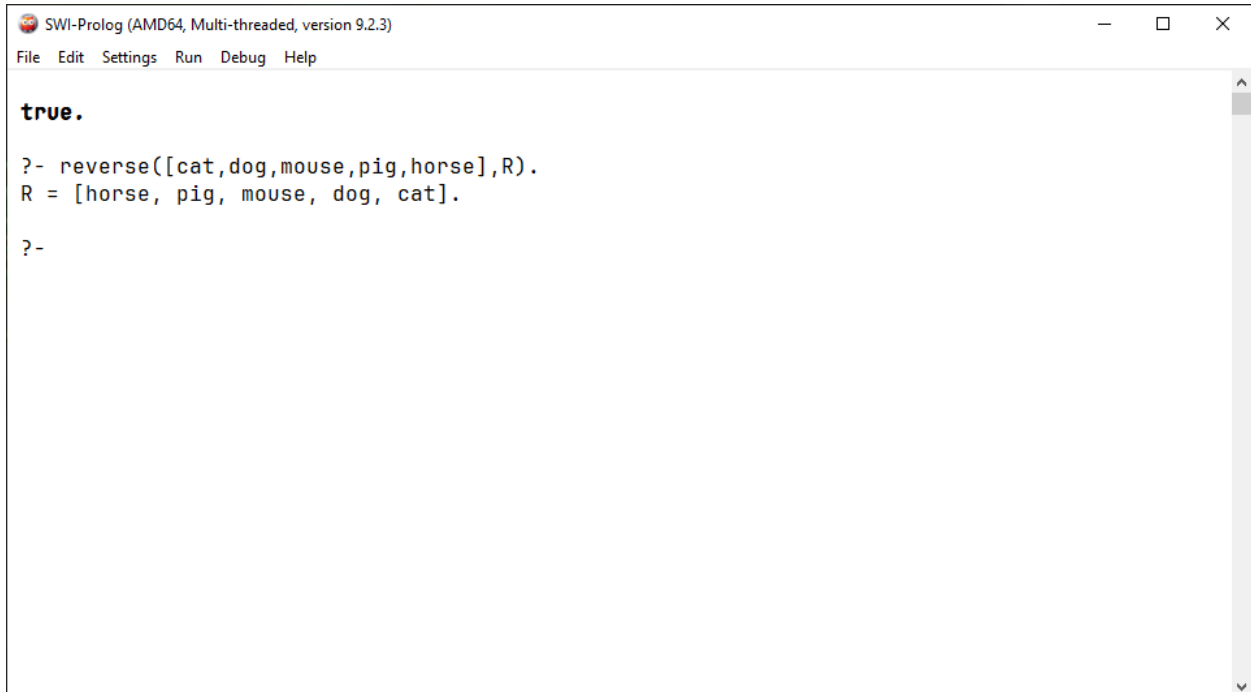


```
p10.pl
File Edit Browse Compile Prolog Pce Help
p10.pl
% program to implement reverse(L,R) where list L is original and list R
% is reversed list.

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

reverse([],[]).
reverse([H|T],R) :- reverse(T,R1), conc(R1,[H],R).
```

c:/users/chirag/documents/prolog/practicals/p10.pl compiled Line: 4



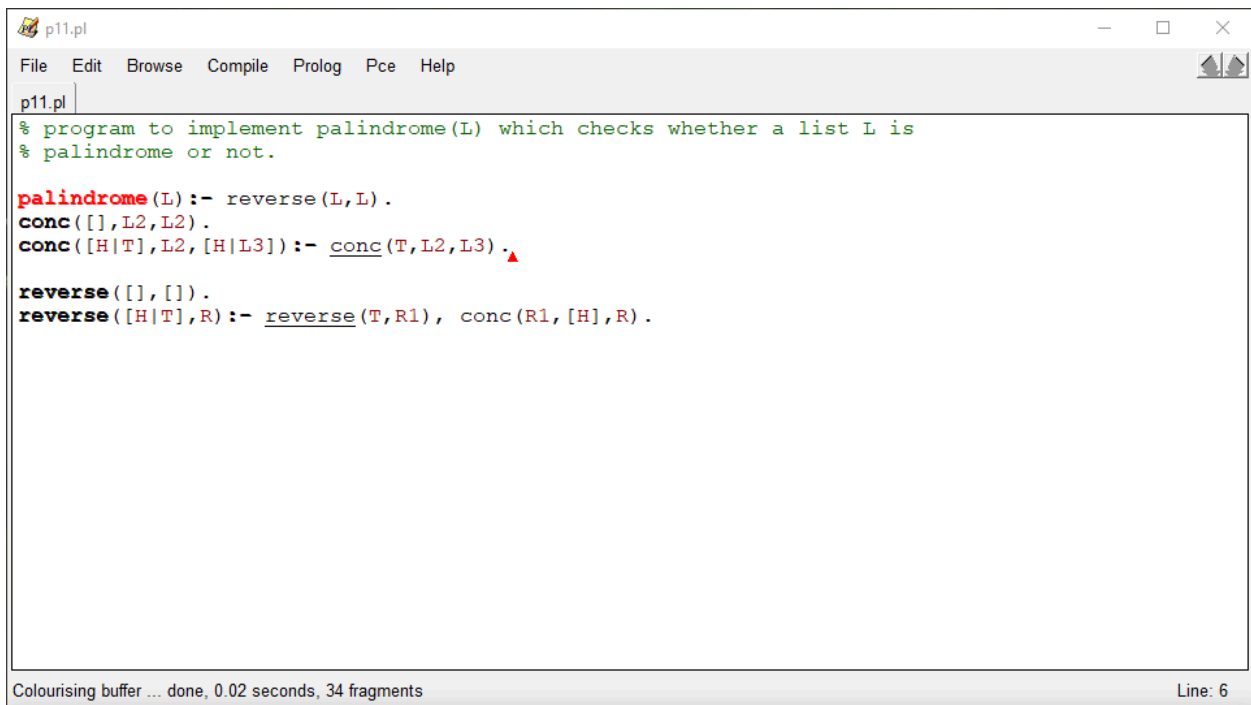
```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.3)
File Edit Settings Run Debug Help

true.

?- reverse([cat,dog,mouse,pig,horse],R).
R = [horse, pig, mouse, dog, cat].

?-
```

Q11. Write a programming prolog to implement `palindrome(L)` which checks whether a list `L` is palindrome or not.



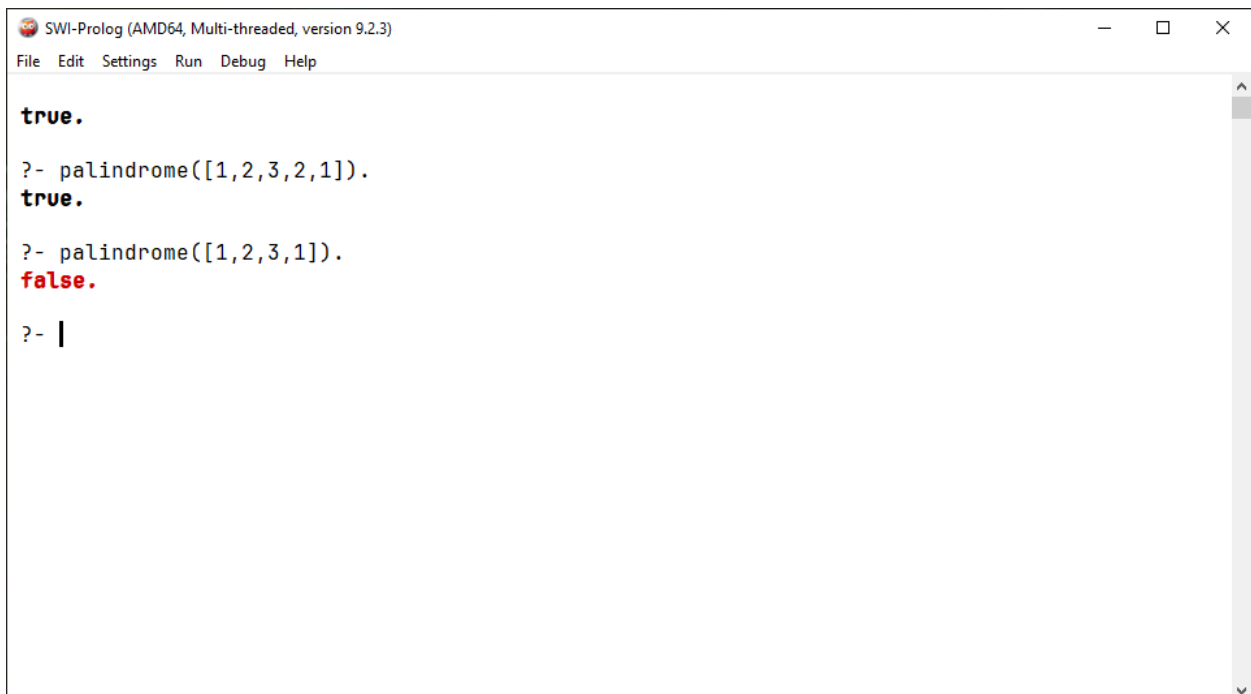
```
p11.pl
File Edit Browse Compile Prolog Pce Help

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

palindrome(L) :- reverse(L,L) .
conc([],L2,L2) .
conc([H|T],L2,[H|L3]) :- conc(T,L2,L3) .

reverse([],[]) .
reverse([H|T],R) :- reverse(T,R1), conc(R1,[H],R) .
```

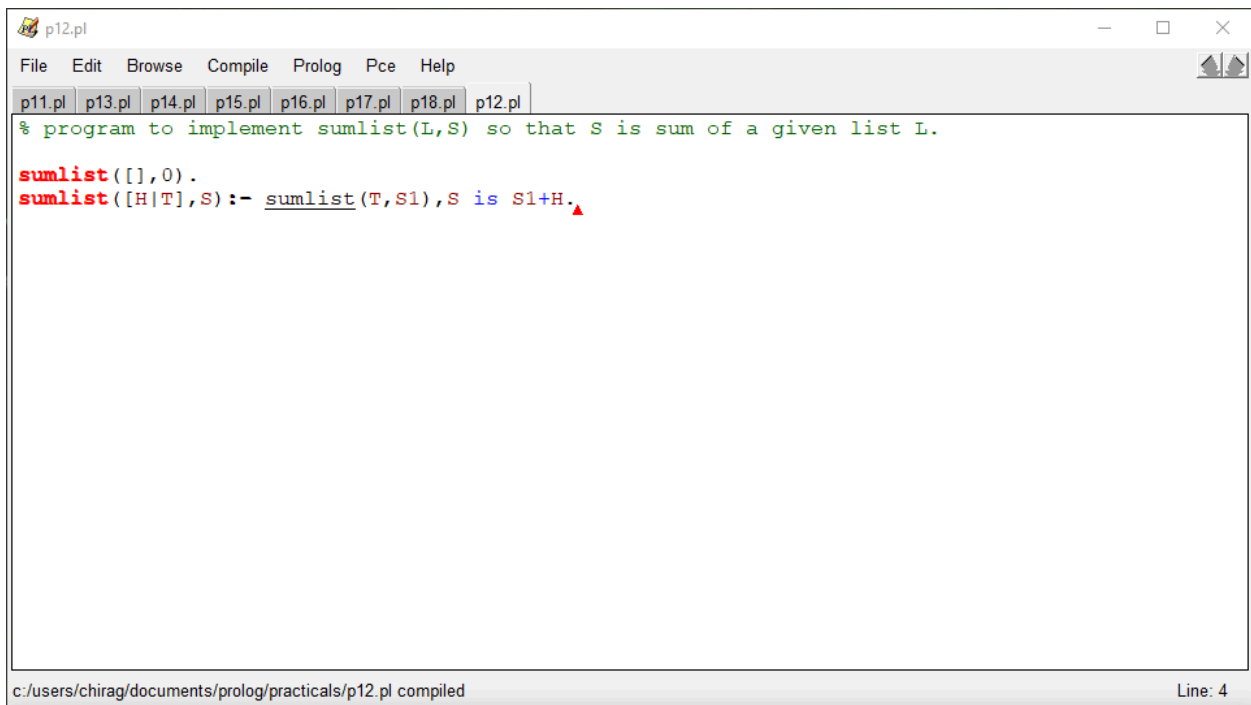
Colourising buffer ... done, 0.02 seconds, 34 fragments Line: 6



```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.3)
File Edit Settings Run Debug Help

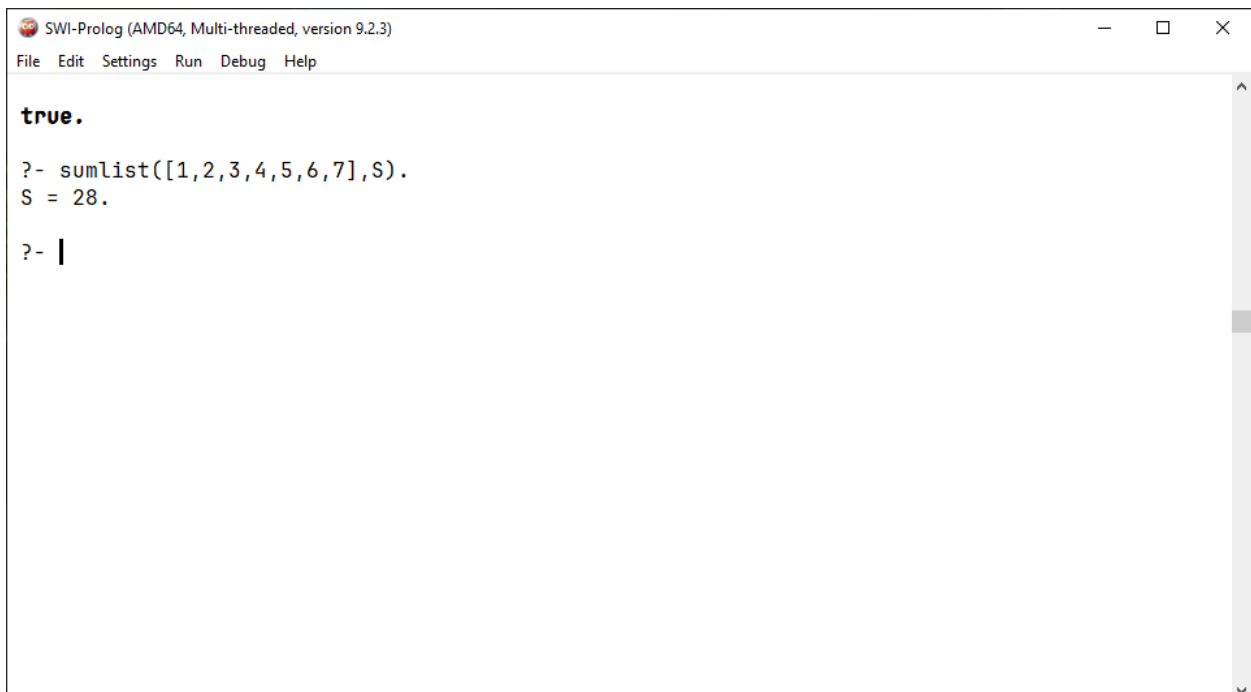
true.
?- palindrome([1,2,3,2,1]).
true.
?- palindrome([1,2,3,1]).
false.
?- |
```

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



```
p12.pl
File Edit Browse Compile Prolog Pce Help
p11.pl p13.pl p14.pl p15.pl p16.pl p17.pl p18.pl p12.pl
% program to implement sumlist(L,S) so that S is sum of a given list L.
sumlist([],0).
sumlist([H|T],S):- sumlist(T,S1),S is S1+H.▲
```

c:/users/chirag/documents/prolog/practicals/p12.pl compiled Line: 4



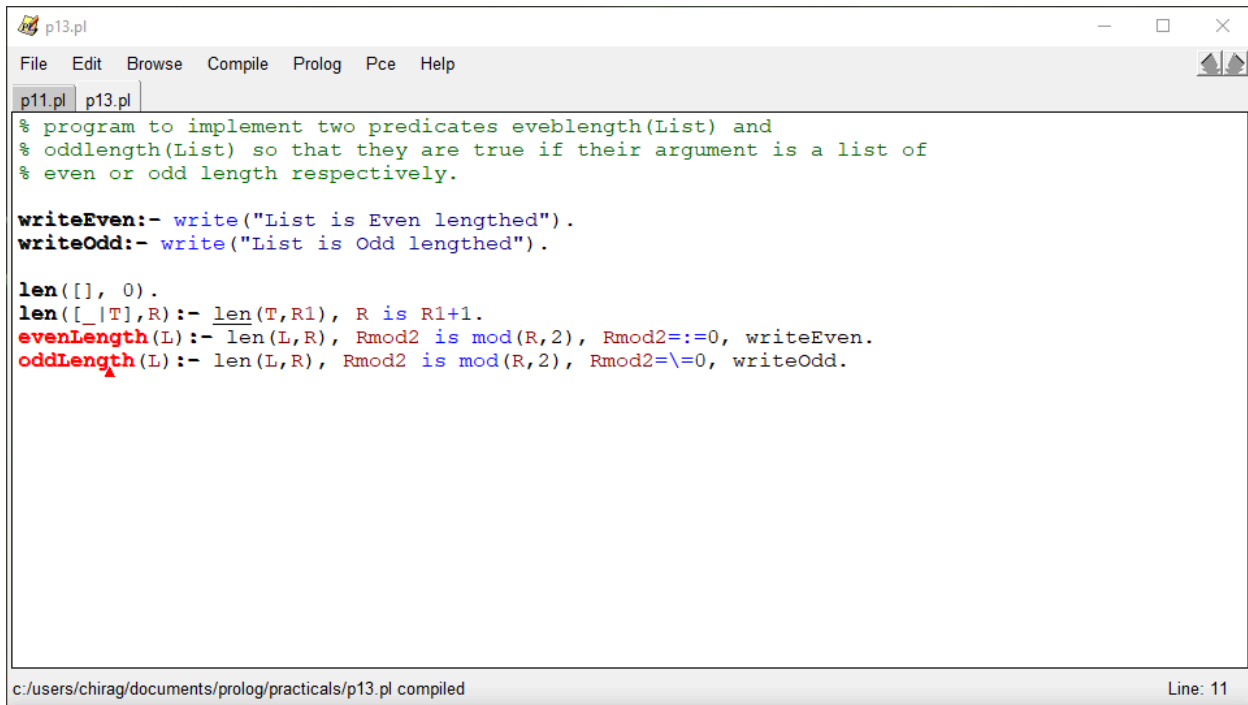
```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.3)
File Edit Settings Run Debug Help

true.

?- sumlist([1,2,3,4,5,6,7],S).
S = 28.

?- |
```

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.

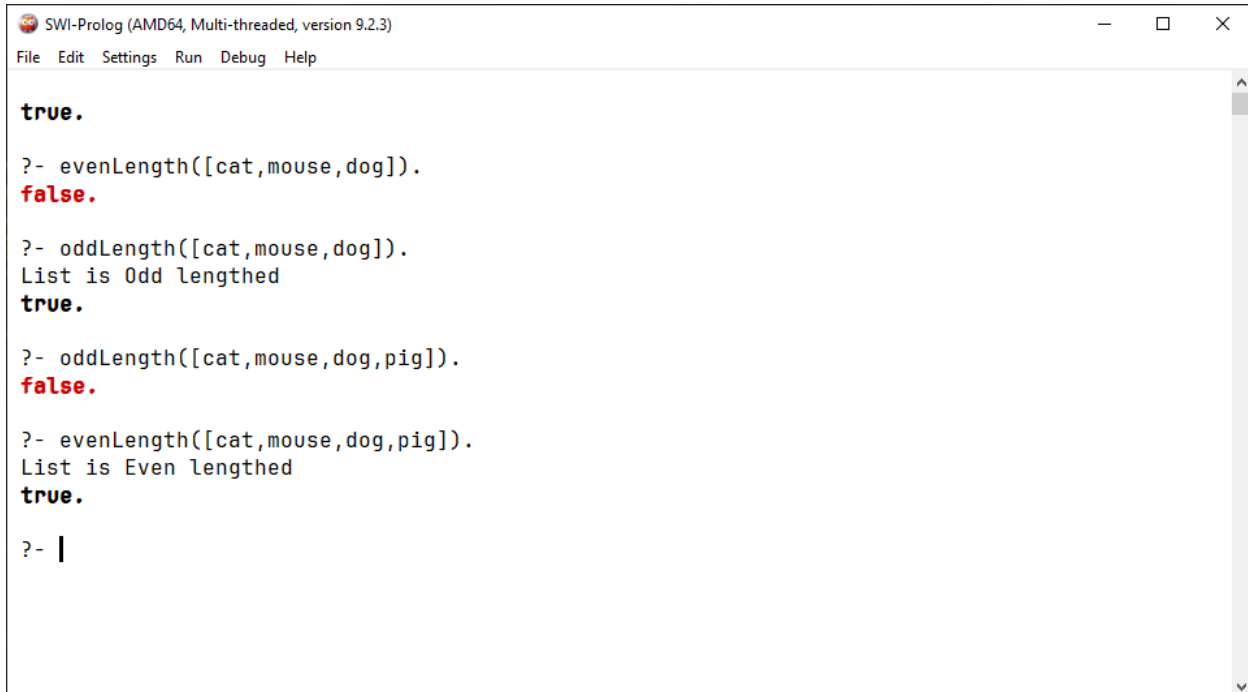


```
p13.pl
File Edit Browse Compile Prolog Pce Help
p11.pl p13.pl
% program to implement two predicates evenlength(List) and
% oddlength(List) so that they are true if their argument is a list of
% even or odd length respectively.

writeEven:- write("List is Even lengthed").
writeOdd:- write("List is Odd lengthed").

len([], 0).
len([_|T],R):- len(T,R1), R is R1+1.
evenLength(L):- len(L,R), Rmod2 is mod(R,2), Rmod2=:=0, writeEven.
oddLength(L):- len(L,R), Rmod2 is mod(R,2), Rmod2=\=0, writeOdd.

c:/users/chirag/documents/prolog/practicals/p13.pl compiled
Line: 11
```



```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.3)
File Edit Settings Run Debug Help

true.

?- evenLength([cat,mouse,dog]).
false.

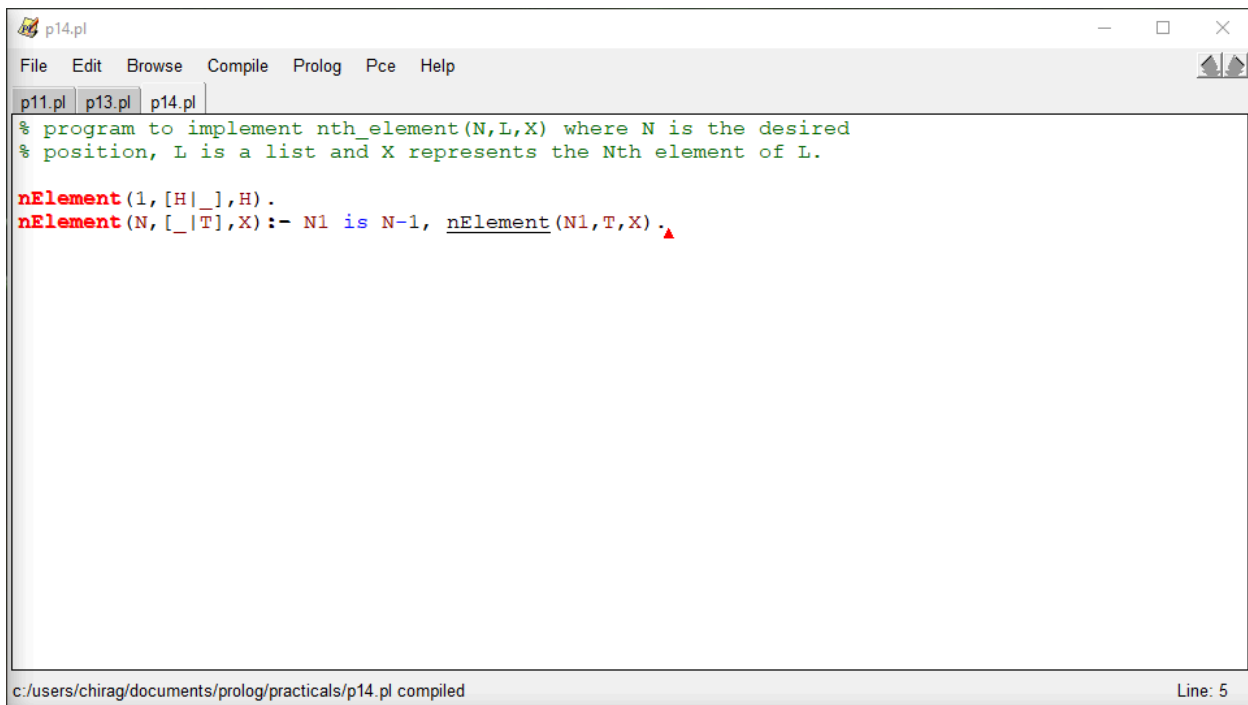
?- oddLength([cat,mouse,dog]).
List is Odd lengthed
true.

?- oddLength([cat,mouse,dog,pig]).
false.

?- evenLength([cat,mouse,dog,pig]).
List is Even lengthed
true.

?- |
```


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 Nth element of L.



```
p14.pl
File Edit Browse Compile Prolog Pce Help
p11.pl p13.pl p14.pl
% 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.

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

c:/users/chirag/documents/prolog/practicals/p14.pl compiled Line: 5



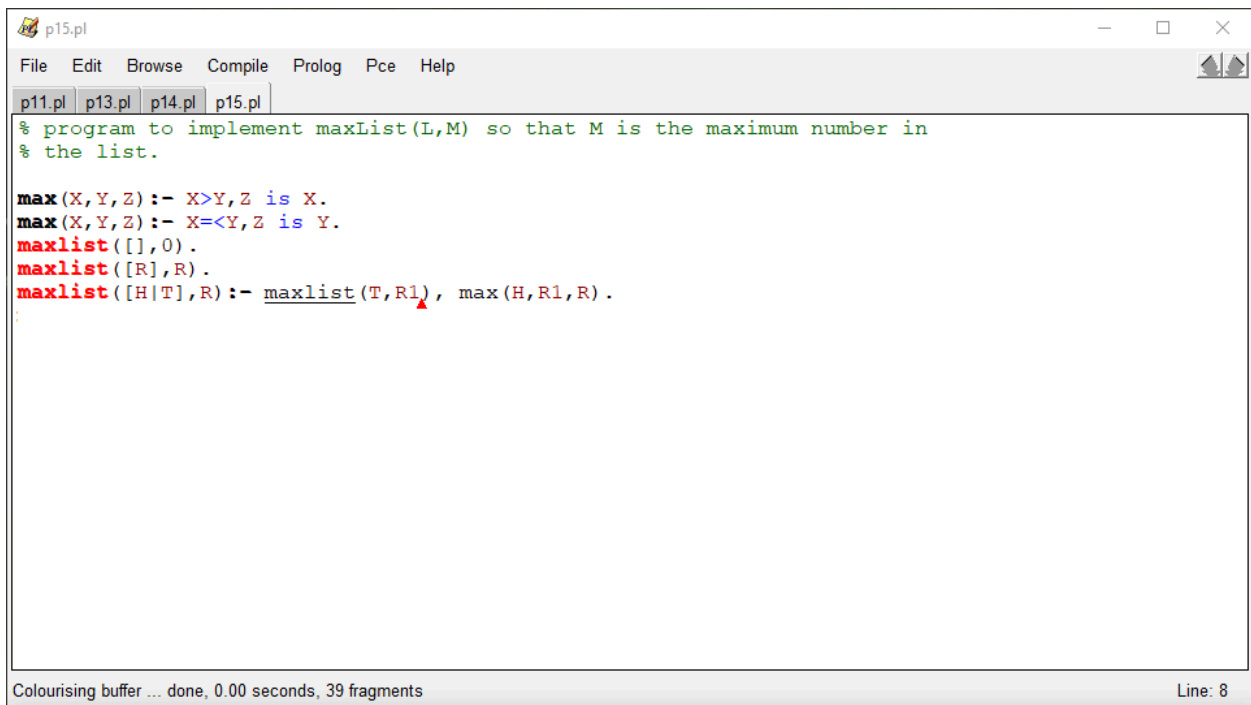
```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.3)
File Edit Settings Run Debug Help

true.

?- nElement(2,[cat,mouse,dog,pig], X).
X = mouse .

?-
```

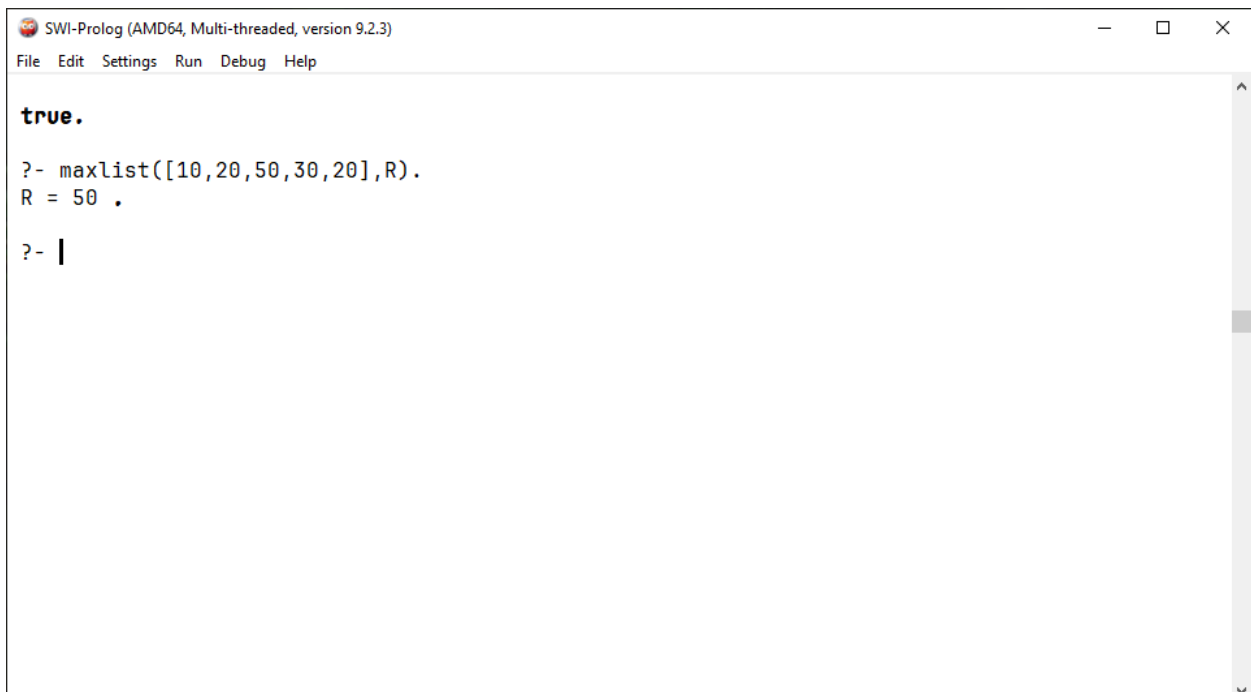

Q15. Write a Prolog program to implement `maxlist(L, M)` so that `M` is the maximum number in the list.



```
p15.pl
File Edit Browse Compile Prolog Pce Help
p11.pl p13.pl p14.pl p15.pl
% program to implement maxList(L,M) so that M is the maximum number in
% the list.

max(X,Y,Z) :- X>Y,Z is X.
max(X,Y,Z) :- X<Y,Z is Y.
maxlist([],0).
maxlist([R],R).
maxlist([H|T],R) :- maxlist(T,R1), max(H,R1,R).
```

Colourising buffer ... done, 0.00 seconds, 39 fragments Line: 8



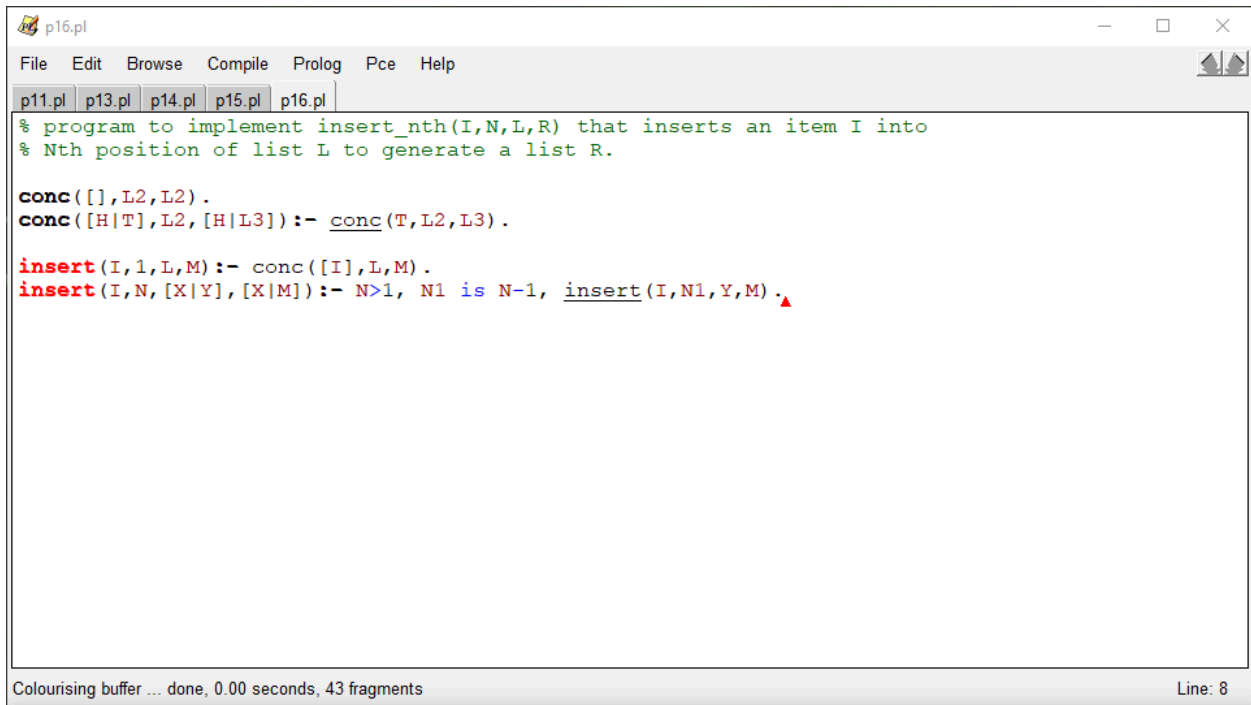
```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.3)
File Edit Settings Run Debug Help

true.

?- maxlist([10,20,50,30,20],R).
R = 50 .

?- |
```

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

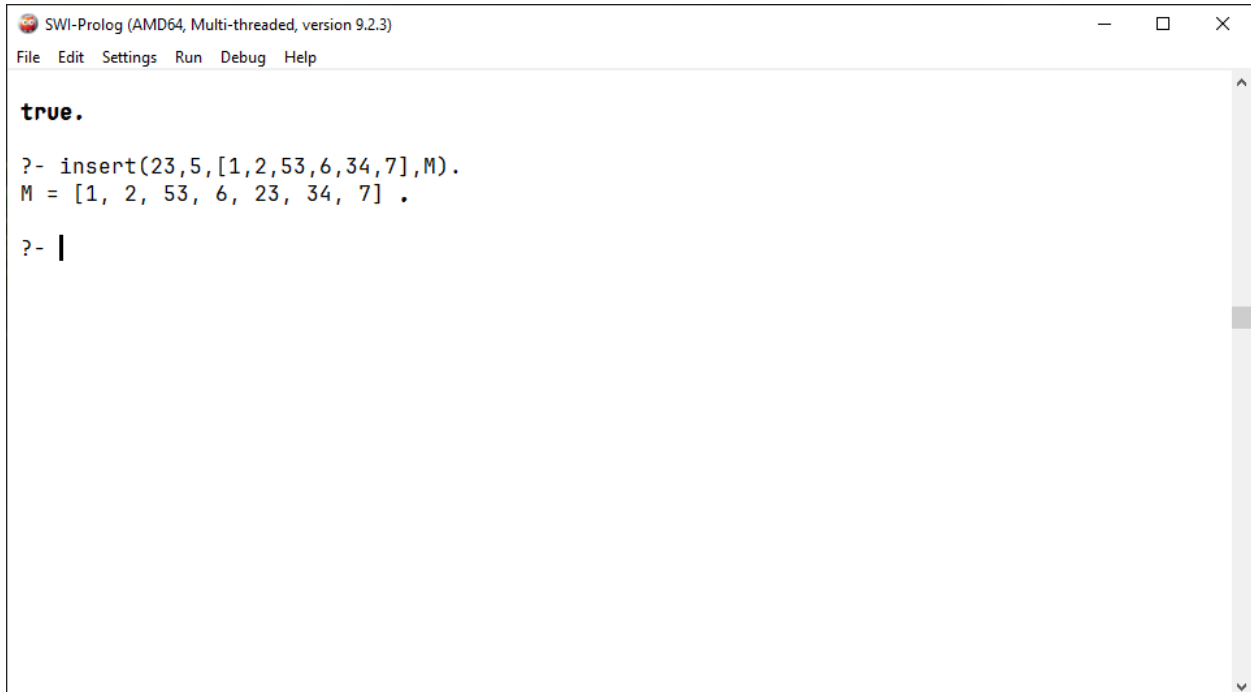


```
p16.pl
File Edit Browse Compile Prolog Pce Help
p11.pl p13.pl p14.pl p15.pl p16.pl
% program to implement insert_nth(I,N,L,R) that inserts an item I into
% Nth position of list L to generate a list R.

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

insert(I,1,L,M) :- conc([I],L,M).
insert(I,N,[X|Y],[X|M]) :- N>1, N1 is N-1, insert(I,N1,Y,M).▲
```

Colourising buffer ... done, 0.00 seconds, 43 fragments Line: 8



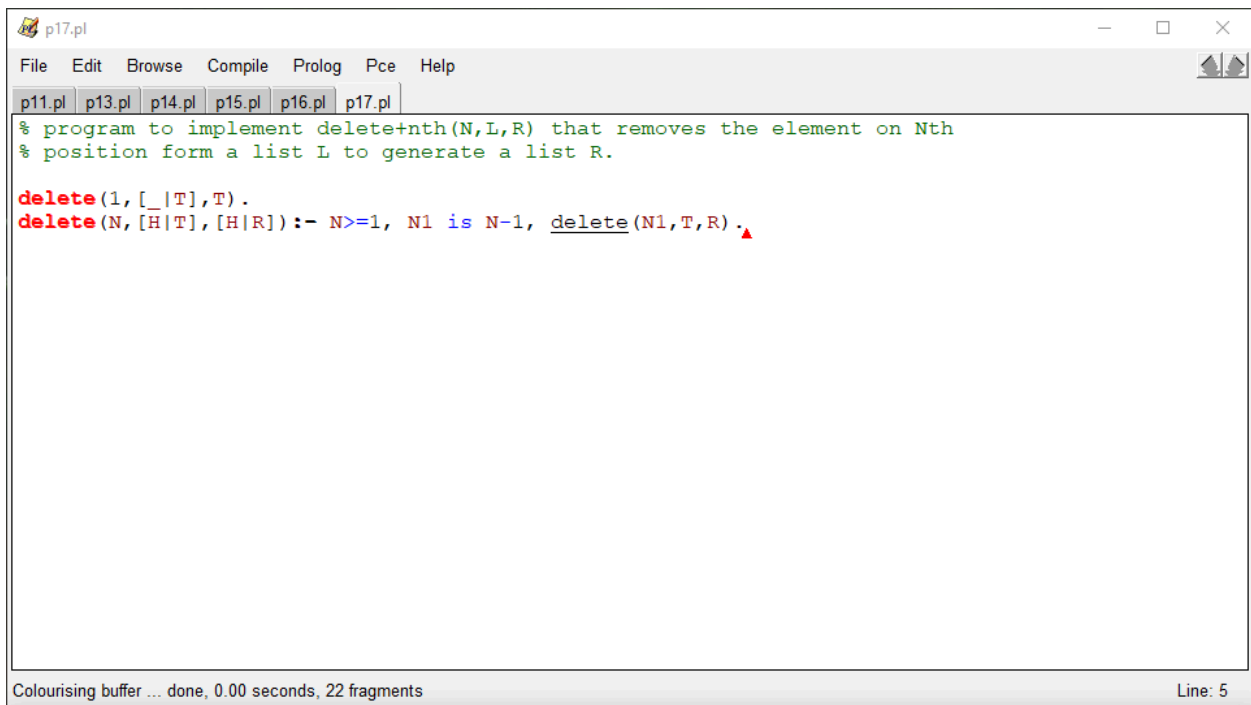
```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.3)
File Edit Settings Run Debug Help

true.

?- insert(23,5,[1,2,53,6,34,7],M).
M = [1, 2, 53, 6, 23, 34, 7] .

?- |
```

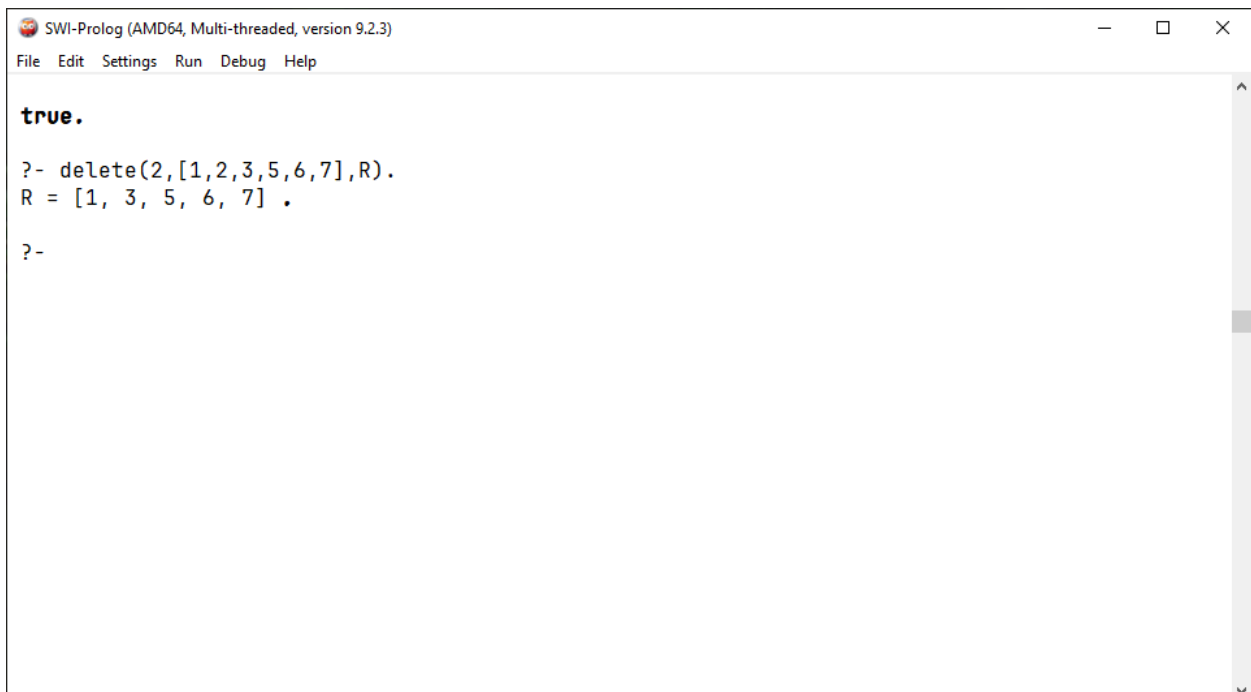
Q17. Write a Prolog program to implement `delete_nth(N, L, R)` that removes the element on Nth position from a list L to generate a list R.



```
p17.pl
File Edit Browse Compile Prolog Pce Help
p11.pl p13.pl p14.pl p15.pl p16.pl p17.pl
% 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(1, [_|T], T).
delete(N, [H|T], [H|R]) :- N >= 1, N1 is N-1, delete(N1, T, R).
```

Colourising buffer ... done, 0.00 seconds, 22 fragments Line: 5



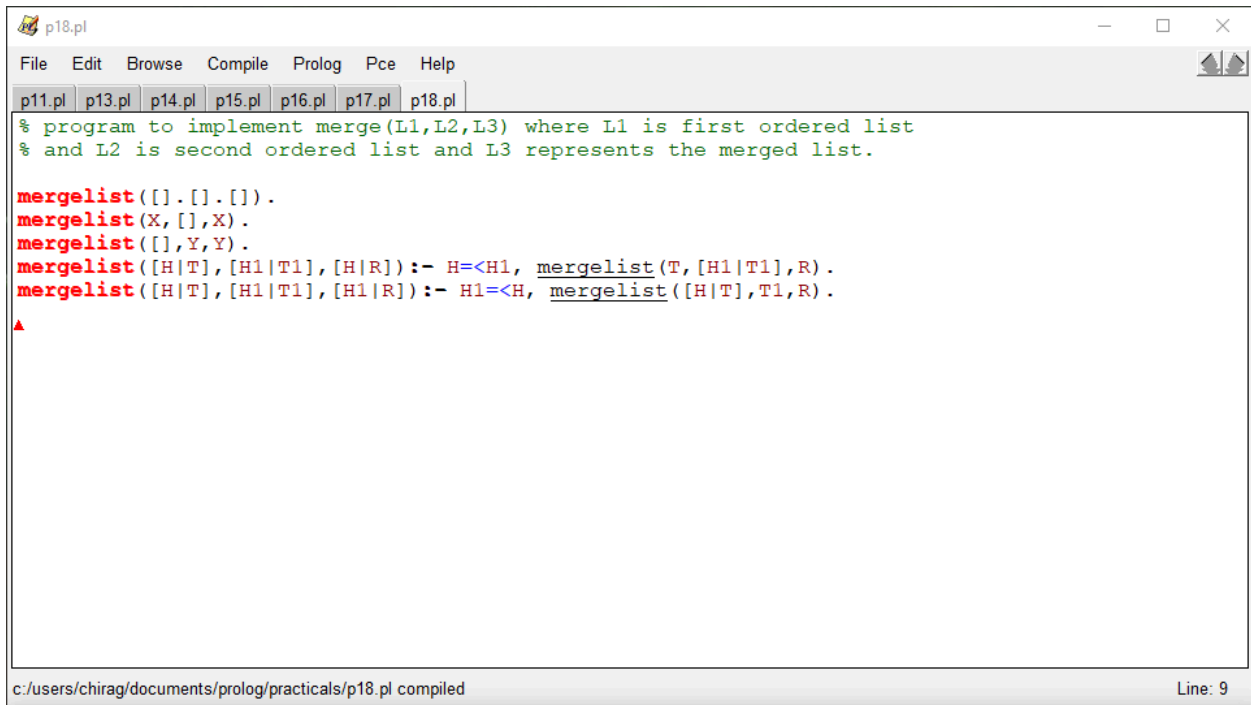
```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.3)
File Edit Settings Run Debug Help

true.

?- delete(2, [1,2,3,5,6,7], R).
R = [1, 3, 5, 6, 7] .

?-
```

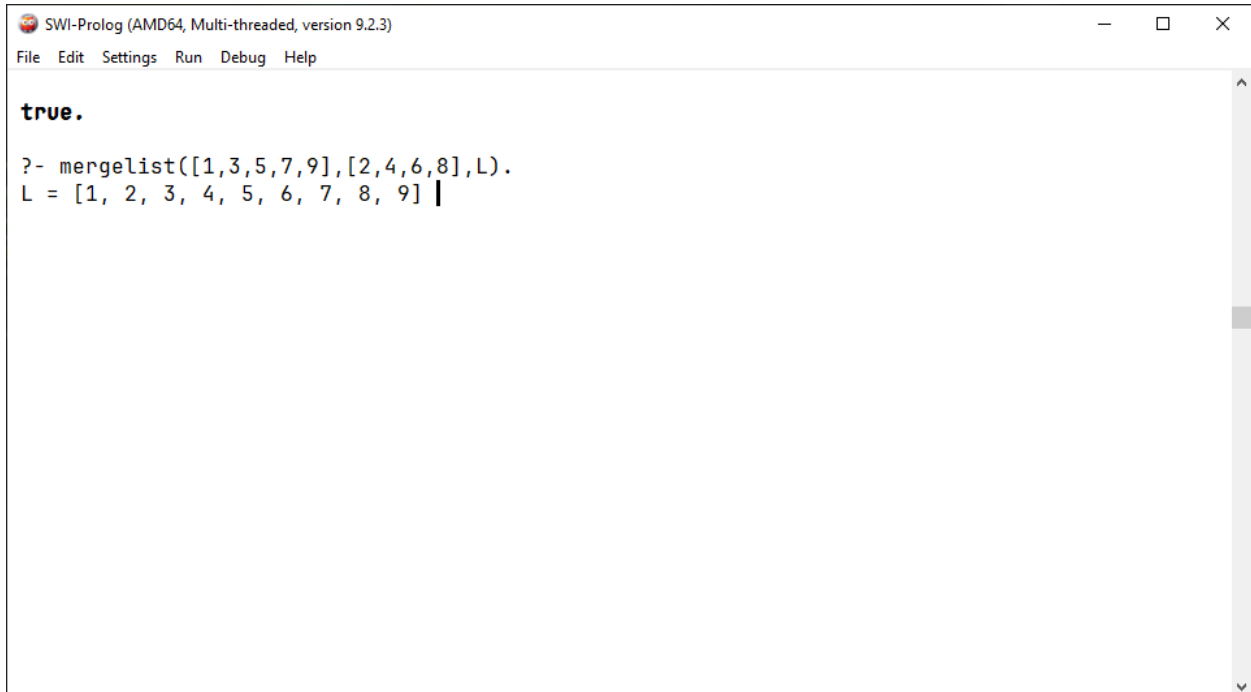
Q18. Write a program in PROLOG to implement merge (L1, L2, L3) where L1 is first ordered list and L2 is second ordered list and L3 represents the merged list.



```
p18.pl
File Edit Browse Compile Prolog Pce Help
p11.pl p13.pl p14.pl p15.pl p16.pl p17.pl p18.pl
% program to implement merge(L1,L2,L3) where L1 is first ordered list
% and L2 is second ordered list and L3 represents the merged list.

mergelist([],[],[]).
mergelist(X,[],X).
mergelist([],Y,Y).
mergelist([H|T],[H1|T1],[H|R]):- H<H1, mergelist(T,[H1|T1],R).
mergelist([H|T],[H1|T1],[H1|R]):- H1<H, mergelist([H|T],T1,R).
^
```

c:/users/chirag/documents/prolog/practicals/p18.pl compiled Line: 9



```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.3)
File Edit Settings Run Debug Help

true.

?- mergelist([1,3,5,7,9],[2,4,6,8],L).
L = [1, 2, 3, 4, 5, 6, 7, 8, 9] |
```