Q.1. Answer:

```
% Define a predicate for addition
add(X, Y, Sum) :-
    Sum is X + Y.
```

Q.2. Answer:

```
% Define a predicate to find the maximum of two numbers
maximum(X, Y, Max) :-
    (X >= Y -> Max = X ; Max = Y).
```

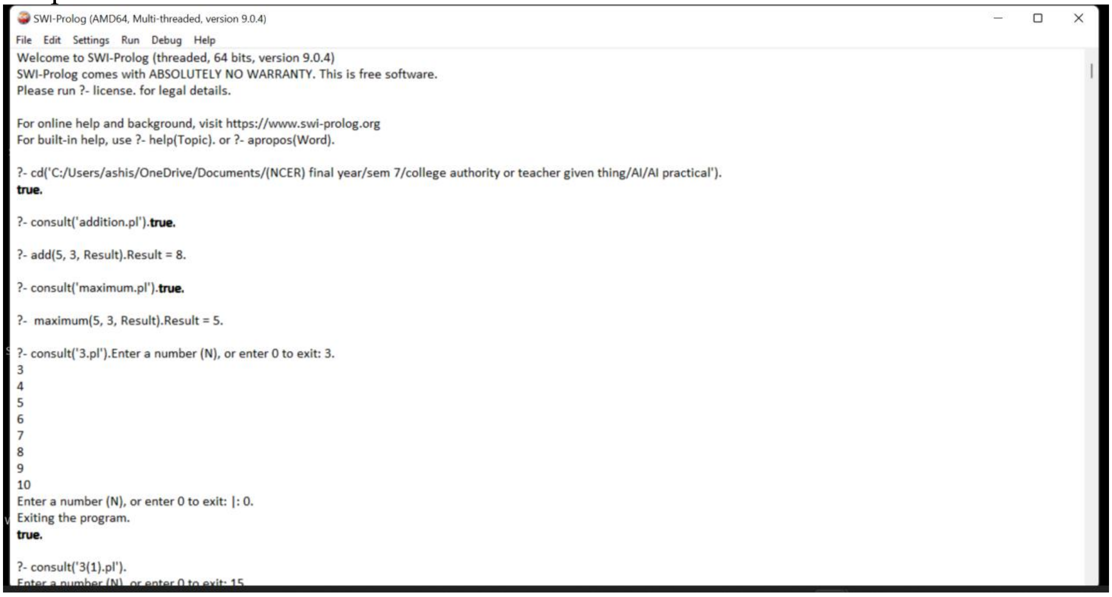
Q.3. Answer:

```
% Define a predicate to count from N to 10
count_from_n_to_10(N) :-
    between(N, 10, X),
    writeln(X),
    X = 10, !.  % Stop when X reaches 10

% Main predicate to read N and start counting
main :-
    write('Enter a number (N), or enter 0 to exit: '),
    read(N), % Read a number from the user

    % Check if the user wants to exit
    (N =:= 0 ->
        writeln('Exiting the program.')
    ;
        % Start counting from N to 10
        count_from_n_to_10(N),
        % Continue the loop
        main
    ).

% Run the program by calling the main predicate
:- initialization(main).
```

Another program.

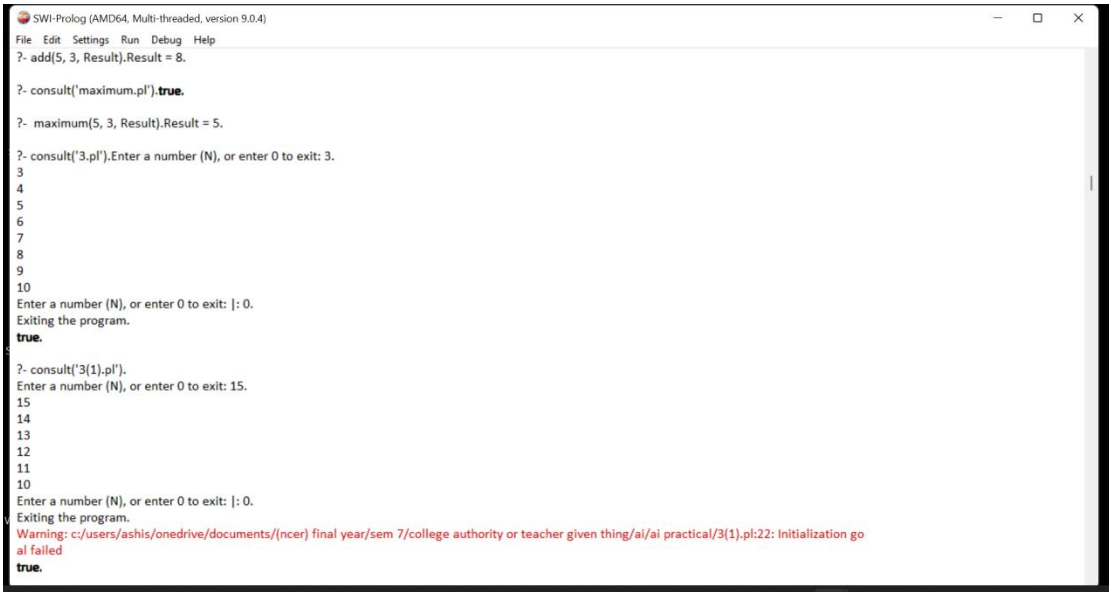
```
% Define a predicate to count from N to 10
count_down(N) :-
    N >= 10,        % Stop the loop when N reaches 10 or exceeds it
    writeln(N),     % Print the current value of N
    Next is N - 1,  % Decrement N by 1
    count_down(Next). % Recursively call count_down with the next value of N

% Main predicate to repeatedly take N and start the counting loop
main_loop :-
    repeat,         % Repeat the loop indefinitely
    write('Enter a number (N), or enter 0 to exit: '),
    read(N),        % Read a number from the user
    (N =:= 0 ->        % Check if N is 0 using =:= for integer comparison
        writeln('Exiting the program.'),
        !
    ;
        count_down(N) % Start counting from N to 10
    ),
    fail.            % Fail to trigger backtracking and repeat the loop

% Run the program by calling the main_loop predicate
:- initialization(main_loop).
```

Q.1,2,3.
Output:



SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
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).

?- cd('C:/Users/ashis/OneDrive/Documents/(NCER) final year/sem 7/college authority or teacher given thing/AI/AI practical').
true.

?- consult('addition.pl').true.

?- add(5, 3, Result).Result = 8.

?- consult('maximum.pl').true.

?-  maximum(5, 3, Result).Result = 5.

?- consult('3.pl').Enter a number (N), or enter 0 to exit: 3.
3
4
5
6
7
8
9
10
Enter a number (N), or enter 0 to exit: |: 0.
Exiting the program.
true.

?- consult('3(1).pl').
Enter a number (N), or enter 0 to exit: 15
```



SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help

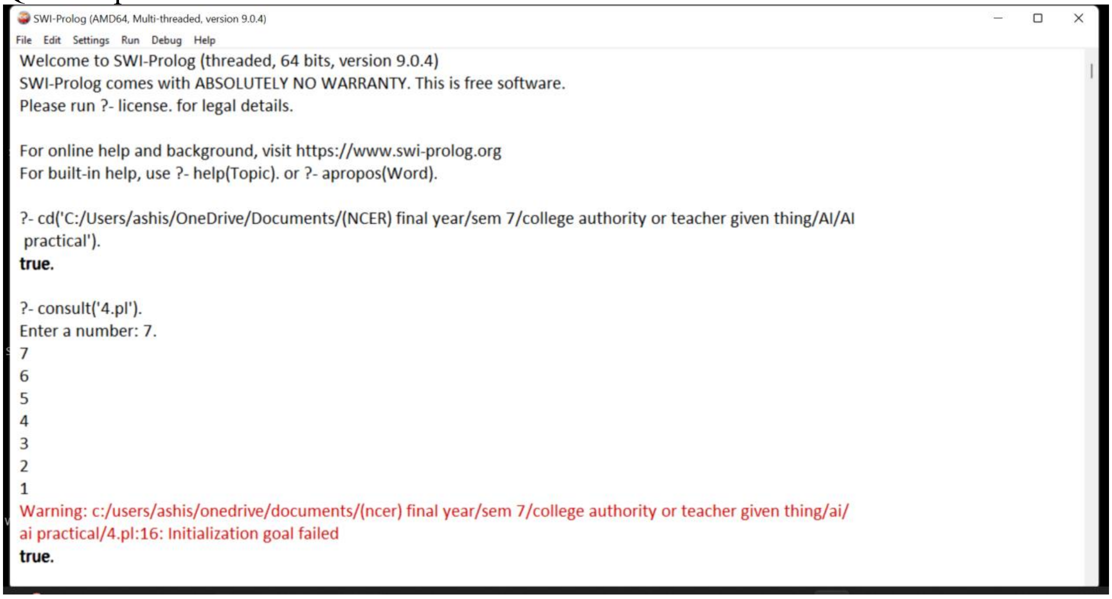
```
?- add(5, 3, Result).Result = 8.

?- consult('maximum.pl').true.

?-  maximum(5, 3, Result).Result = 5.

?- consult('3.pl').Enter a number (N), or enter 0 to exit: 3.
3
4
5
6
7
8
9
10
Enter a number (N), or enter 0 to exit: |: 0.
Exiting the program.
true.

?- consult('3(1).pl').
Enter a number (N), or enter 0 to exit: 15.
15
14
13
12
11
10
Enter a number (N), or enter 0 to exit: |: 0.
Exiting the program.
Warning: c:/users/ashis/onedrive/documents/(ncer) final year/sem 7/college authority or teacher given thing/ai/ai practical/3(1).pl:22: Initialization go
al failed
true.
```

Q.4. Answer:

```
% Define a predicate to count from N to 1
count_down(N) :-
```

```
    N >= 1,       % Stop when N reaches 1
    writeln(N),   % Print the current value of N
    Next is N - 1, % Decrement N by 1
    count_down(Next).

% Main predicate to read N and start counting
main :-
    write('Enter a number: '),
    read(N),      % Read a number from the user
    count_down(N),% Start counting from N to 1
    halt.         % Halt the Prolog interpreter (may vary depending on your Prolog
system).

% Run the program by calling the main predicate
:- initialization(main).
```

Q.4. Output:

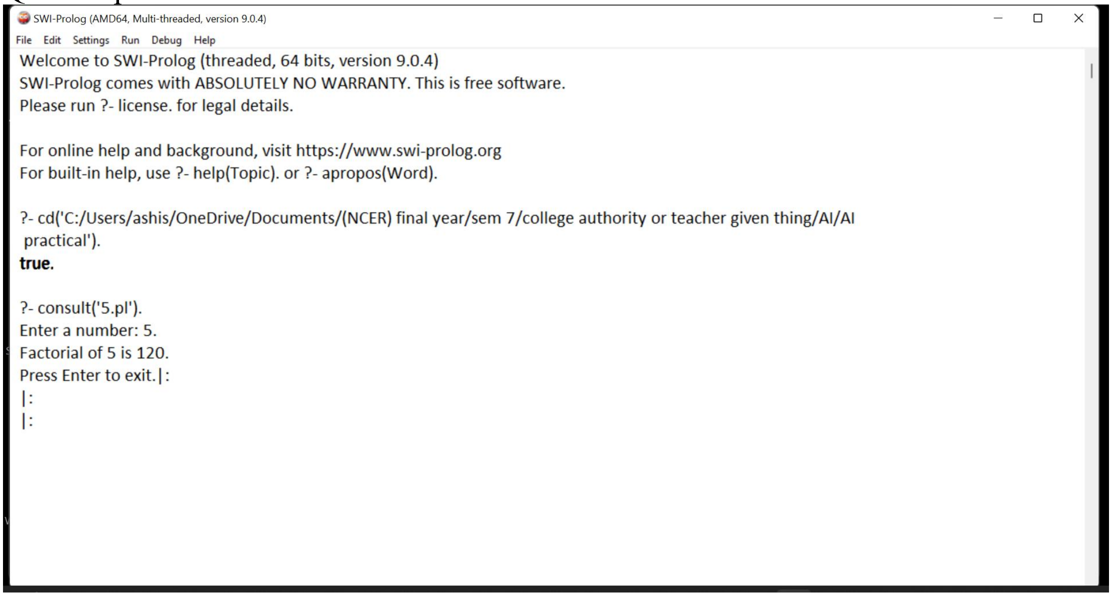

SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
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).

?- cd('C:/Users/ashis/OneDrive/Documents/(NCER) final year/sem 7/college authority or teacher given thing/AI/AI
 practical').
true.

?- consult('4.pl').
Enter a number: 7.
7
6
5
4
3
2
1
Warning: c:/users/ashis/onedrive/documents/(ncer) final year/sem 7/college authority or teacher given thing/ai/
ai practical/4.pl:16: Initialization goal failed
true.
```

Q.5. Answer:

```
% Define a predicate to calculate the factorial of N
factorial(0, 1). % Base case: The factorial of 0 is 1.

factorial(N, Result) :-
    N > 0,          % Ensure N is a positive integer
    Prev is N - 1,  % Calculate N - 1
    factorial(Prev, SubResult), % Recursively compute (N-1)!
    Result is N * SubResult.    % Calculate N! using (N-1)!

% New main predicate to read N and calculate factorial
calculate_factorial :-
    write('Enter a number: '),
    read(N),        % Read a number from the user
    factorial(N, Result), % Calculate factorial
    format('Factorial of ~d is ~d.~n', [N, Result]), % Print the result
    wait_for_input. % Wait for user input before exiting

wait_for_input :-
    write('Press Enter to exit.'),
    read(_), % Read a line and discard it (waiting for Enter key)
    halt.        % Halt the Prolog interpreter

% Run the program by calling the new main predicate
:- initialization(calculate_factorial).
```

Q.5. Output:


SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help

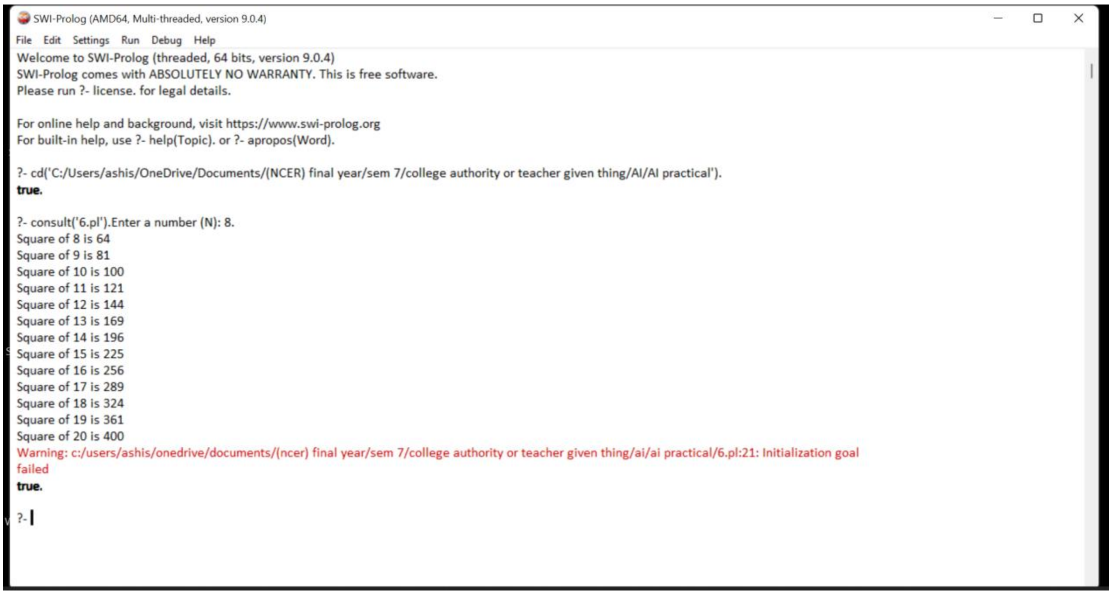
```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- cd('C:/Users/ashis/OneDrive/Documents/(NCER) final year/sem 7/college authority or teacher given thing/AI/AI
 practical').
true.

?- consult('5.pl').
Enter a number: 5.
Factorial of 5 is 120.
Press Enter to exit.|:
|:
|:
```

Q.6. Answer:

```
% Define a predicate to calculate the square of a number
square(X, X_Squared) :-
    X_Squared is X * X.

% Define a predicate to calculate and display squares from N to 20
calculate_and_display_squares(N) :-
    N =< 20, % Stop when N reaches 20 or exceeds it
    square(N, Square), % Calculate the square of N
    format('Square of ~d is ~d~n', [N, Square]), % Display the result
    Next is N + 1, % Increment N by 1
    calculate_and_display_squares(Next).

% Main predicate to read N and start the calculation
main :-
    write('Enter a number (N): '),
    read(N), % Read a number from the user
    calculate_and_display_squares(N), % Start the calculation and display
    halt. % Halt the Prolog interpreter (may vary depending on your Prolog system).
```

```
% Run the program by calling the main predicate
:- initialization(main).
```

Q.6. Output:



SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
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).

?- cd('C:/Users/ashis/OneDrive/Documents/(NCER) final year/sem 7/college authority or teacher given thing/AI/AI practical').
true.

?- consult('6.pl').Enter a number (N): 8.
Square of 8 is 64
Square of 9 is 81
Square of 10 is 100
Square of 11 is 121
Square of 12 is 144
Square of 13 is 169
Square of 14 is 196
Square of 15 is 225
Square of 16 is 256
Square of 17 is 289
Square of 18 is 324
Square of 19 is 361
Square of 20 is 400
Warning: c:/users/ashis/onedrive/documents/(ncer) final year/sem 7/college authority or teacher given thing/ai/ai practical/6.pl:21: Initialization goal
failed
true.

?-
```