Computer Programming Paradigms Lab

PROLOG Lab 2

1. Write a program to check whether a given integer is odd or even:

```
checkeven(N):- M is N//2, N=:=2*M.

?-checkeven(12).
True

?-checkeven(23).
False
```

2. Write a program to output integers from a specified value down to 1:

```
loop(0).
loop(N):-N>0, write(N),nl,M is N-1,loop(M)

?-loop(6).
6
5
4
3
2
1
True.
```

3. Write a program to output integers from 1 to a specified value:

```
loop(0).
loop(N):-N>0,M is N-1,loop(M),write(N),nl.

?-loop(6).
1
2
3
4
5
6
True
```

4. Write a program that outputs integers from First to Last both inclusive.

```
output_values(Last, Last):-write(Last),nl.
output_values(First, Last):- First=\=Last, write(First),nl, N is First
+ 1, output_values(N, Last)

?-output_values(5,12).
5
6
7
8
9
10
11
12
True
```

5. Define a predicate to find the sum of the integers from 1 to N.

```
sumto(1,1).
sumto(N,S):-N>1, N1 is N-1, sumto(N1, S1), S is S1 + N.

?-sumto(100,N).
N = 5050
```

6. Output the first N squares, one per line.

```
writesquares(1):-write(1),nl.
writesquares(N):-N>1, N1 is N-1, writesquares(N1), Nsq is N*N,
write(Nsq),nl.

?-writesquares(6).
1
4
9
16
25
36

True
```

7. Write a program to loop until a condition is satisfied.

```
loop:-write('Type end to end'),read(Word),write('Input was
'),write(Word),nl,(Word=end;loop).
```

8. Write a program to find factorial using recursion.

```
factorial(1,1).
factorial(N,Nfact):-N1 is N-1,factorial(N1,Nfact1), Nfact is N*Nfact1.

?-factorial(6,N).

N = 720
```

9. Define and test a predicate **palindrome** that checks whether a list reads the same way forwards and backwards, e.g.

```
?- palindrome([a,b,c,b,a]).
true
?- palindrome([a,b,c,d,e]).
false

palindrome(A):-reverse(A,A).
```

10. Define and test a predicate which takes two arguments, both numbers, and calculates and outputs the following values: (a) their average, (b) the square root of their product and (c) the larger of (a) and (b).

```
pred(A,B):-X is (A+B)/2, write('Average is: '), write(X), nl, Y is
sqrt(A*B), write('Square root of product is: '), write(Y), nl, Z is
max(X,Y),write('Larger is: '),write(Z),nl.
```