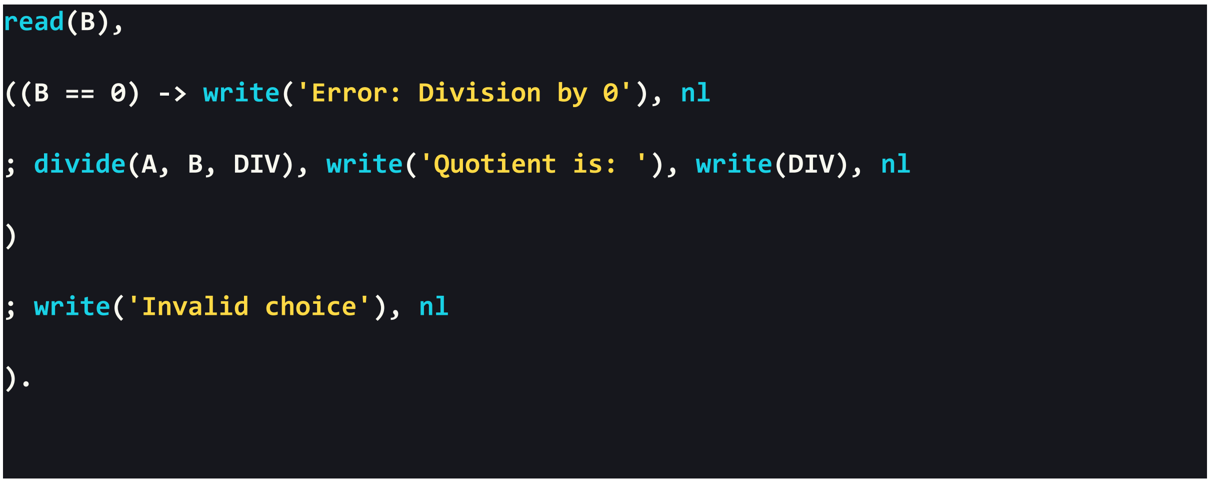
PPL LAB 3

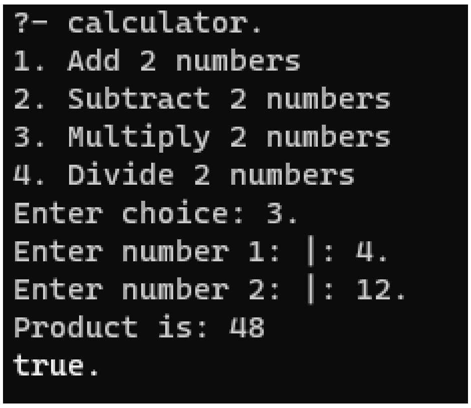
SAHIL BONDRE: U18CO021

1. Write a prolog program to implement a Menu Driven Calculator.

```
add(A, B, SUM) :- SUM is A + B.
subtract(A, B, DIF) :- DIF is A - B.
multiply(A, B, MUL):- MUL is A * B.
divide(A, B, DIV):- DIV is A / B.

calculator :- write('1. Add'), nl, write('2. Subtract'), nl,
write('3. Multiply'), nl, write('4. Divide'), nl,
write('Enter choice: '), read(X),
((X == 1)
-> write('Enter number 1: '), read(A), write('Enter number 2: '),
read(B),
add(A, B, SUM), write('Sum is: '), write(SUM), nl
; (X == 2)
-> write('Enter number 1: '), read(A), write('Enter number 2: '),
read(B),
subtract(A, B, DIF), write('Difference is: '), write(DIF), nl
; (X == 3)
-> write('Enter number 1: '), read(A), write('Enter number 2: '),
read(B),
multiply(A, B, MUL), write('Product is: '), write(MUL), nl
; (X == 4)
-> write('Enter number 1: '), read(A), write('Enter number 2: '),
```

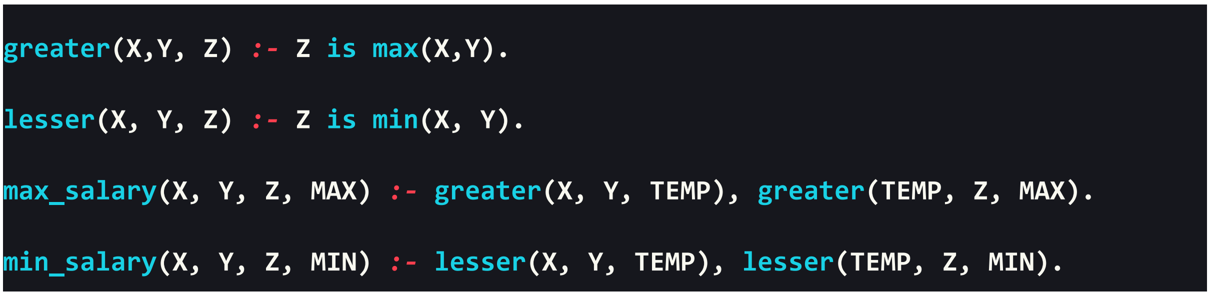


```
read(B),

((B == 0) -> write('Error: Division by 0'), nl

; divide(A, B, DIV), write('Quotient is: '), write(DIV), nl

)

; write('Invalid choice'), nl

).
```

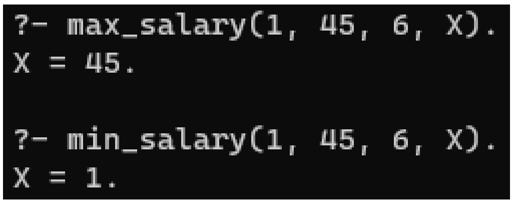


```
?- calculator.
1. Add 2 numbers
2. Subtract 2 numbers
3. Multiply 2 numbers
4. Divide 2 numbers
Enter choice: 3.
Enter number 1: |: 4.
Enter number 2: |: 12.
Product is: 48
true.
```

2. Write a prolog program to find maximum and minimum of salaries of given 3 employees.



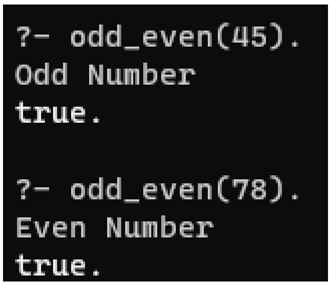
```
greater(X,Y, Z) :- Z is max(X,Y).

lesser(X, Y, Z) :- Z is min(X, Y).

max_salary(X, Y, Z, MAX) :- greater(X, Y, TEMP), greater(TEMP, Z, MAX).

min_salary(X, Y, Z, MIN) :- lesser(X, Y, TEMP), lesser(TEMP, Z, MIN).
```



```
?- max_salary(1, 45, 6, X).
X = 45.

?- min_salary(1, 45, 6, X).
X = 1.
```

3. Write a prolog program to check whether given number is odd or even.

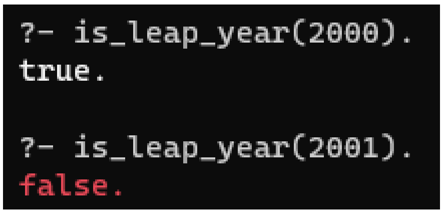
```
odd_even(X) :- (

0 is mod(X, 2)-> write('Even Number'), nl

; write('Odd Number'), nl

).
```



```
?- odd_even(45).
Odd Number
true.

?- odd_even(78).
Even Number
true.
```

4. Write a prolog program to check whether a given year is leap year or not.

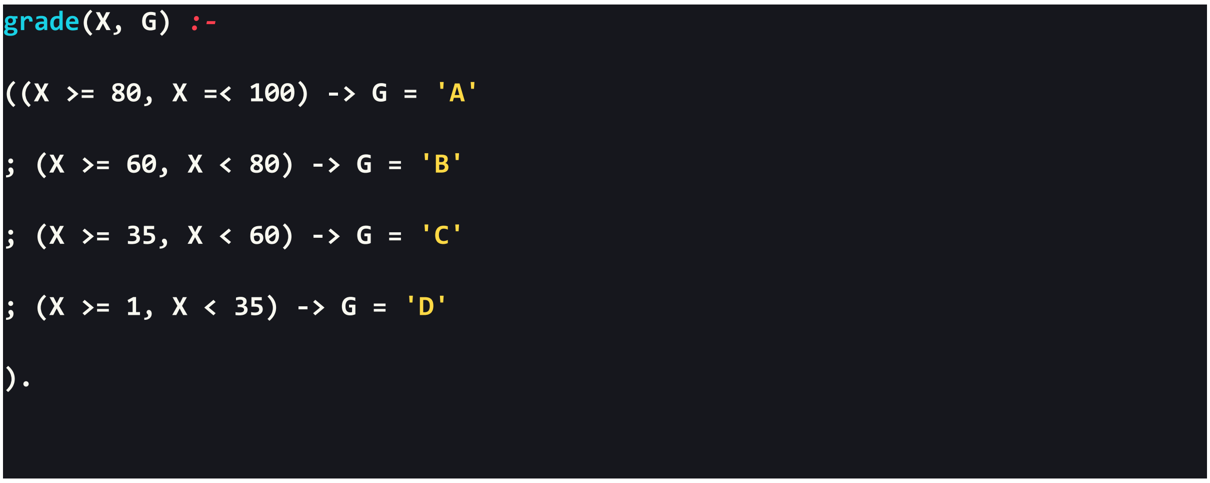
```
is_leap_year(Year) :-

R4 is Year mod 4,

R100 is Year mod 100,

R400 is Year mod 400,

((R4 = 0, R100 \= 0); R400 = 0).
```



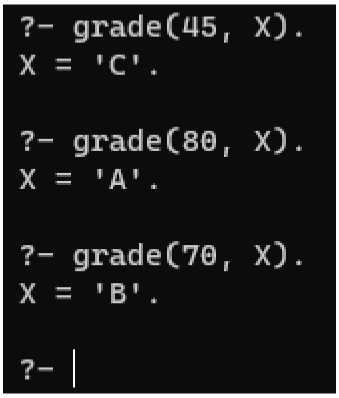
```
?- is_leap_year(2000).
true.

?- is_leap_year(2001).
false.
```

5. Write a prolog program to give grade to a student based on total marks given:

- 100 - 80 Grade A
- 60 - 79 Grade B
- 35 - 59 Grade C
- 1 - 35 Grade D

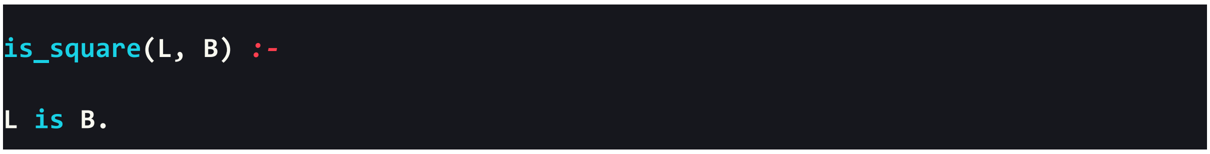


```
grade(X, G) :-

((X >= 80, X =< 100) -> G = 'A'

; (X >= 60, X < 80) -> G = 'B'

; (X >= 35, X < 60) -> G = 'C'

; (X >= 1, X < 35) -> G = 'D'

).
```

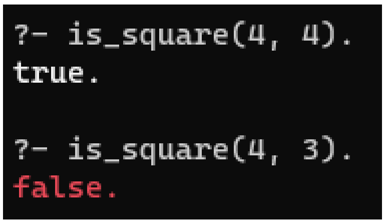


```
?- grade(45, X).
X = 'C'.

?- grade(80, X).
X = 'A'.

?- grade(70, X).
X = 'B'.

?- 
```

6. Write a prolog program to take values of length and breadth of a rectangle from the user and check if it is square or not.

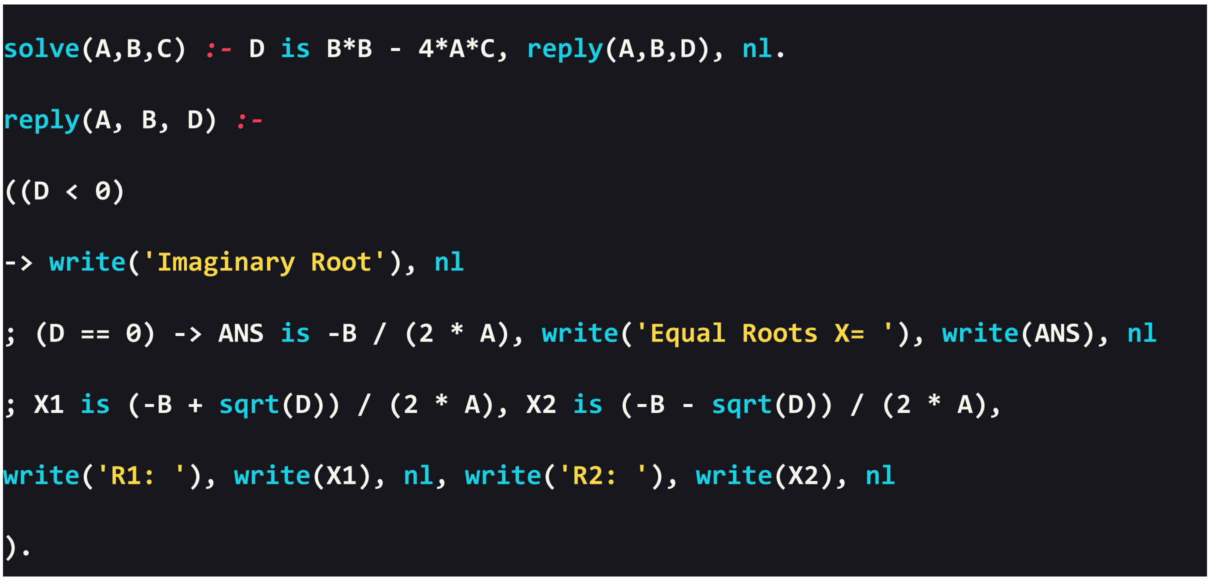


```
is_square(L, B) :-

L is B.
```

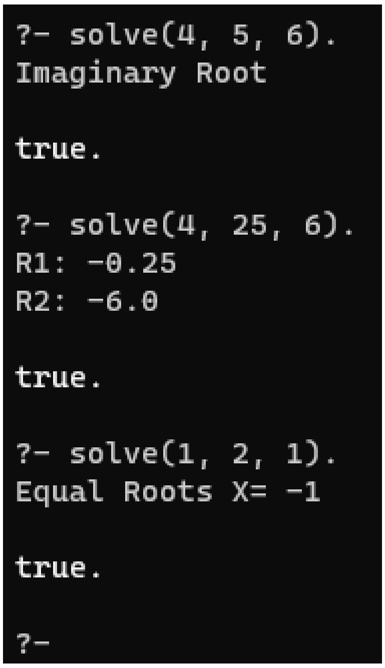


```
?- is_square(4, 4).
true.

?- is_square(4, 3).
false.
```

7. Write a PROLOG program to calculate the roots of quadratic equation Consider all possibilities real, equal, imaginary.

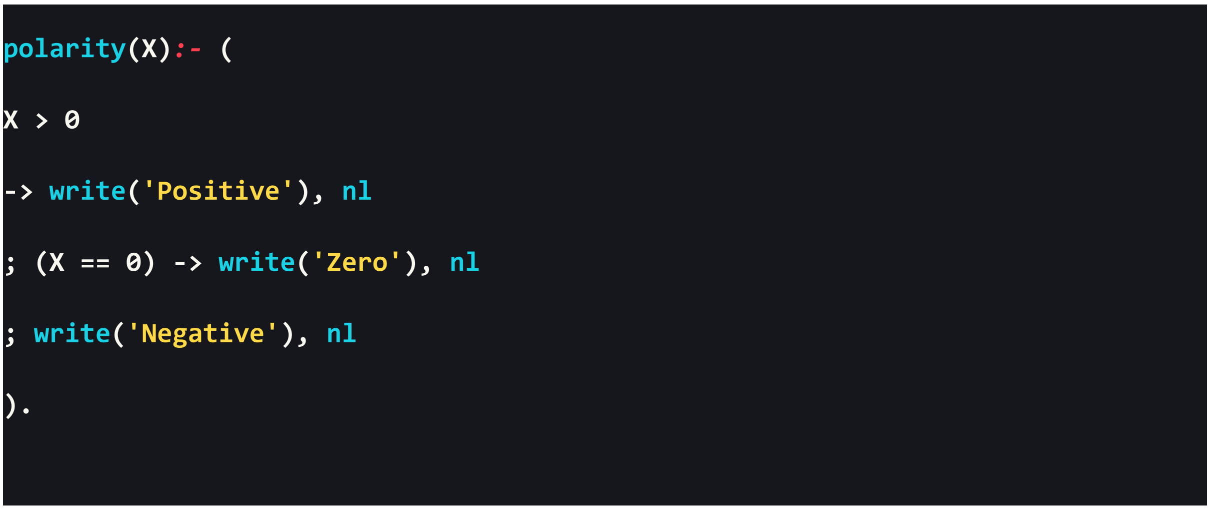


```
solve(A,B,C) :- D is B*B - 4*A*C, reply(A,B,D), nl.
reply(A, B, D) :-
((D < 0)
-> write('Imaginary Root'), nl
; (D == 0) -> ANS is -B / (2 * A), write('Equal Roots X= '), write(ANS), nl
; X1 is (-B + sqrt(D)) / (2 * A), X2 is (-B - sqrt(D)) / (2 * A),
write('R1: '), write(X1), nl, write('R2: '), write(X2), nl
).
```

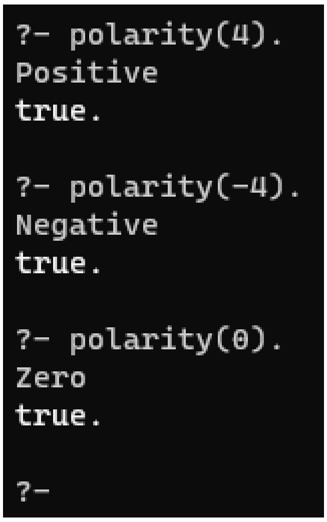


```
?- solve(4, 5, 6).
Imaginary Root

true.

?- solve(4, 25, 6).
R1: -0.25
R2: -6.0

true.

?- solve(1, 2, 1).
Equal Roots X= -1

true.

?-
```

8. Write a PROLOG program to find the number whether the number is positive, negative or Zero.



```
polarity(X):- (

X > 0

-> write('Positive'), nl

; (X == 0) -> write('Zero'), nl

; write('Negative'), nl

).
```



```
?- polarity(4).
Positive
true.

?- polarity(-4).
Negative
true.

?- polarity(0).
Zero
true.

?-
```