# Principles of Programming Language (CS302)

## Assignment - 8

## U19CS012

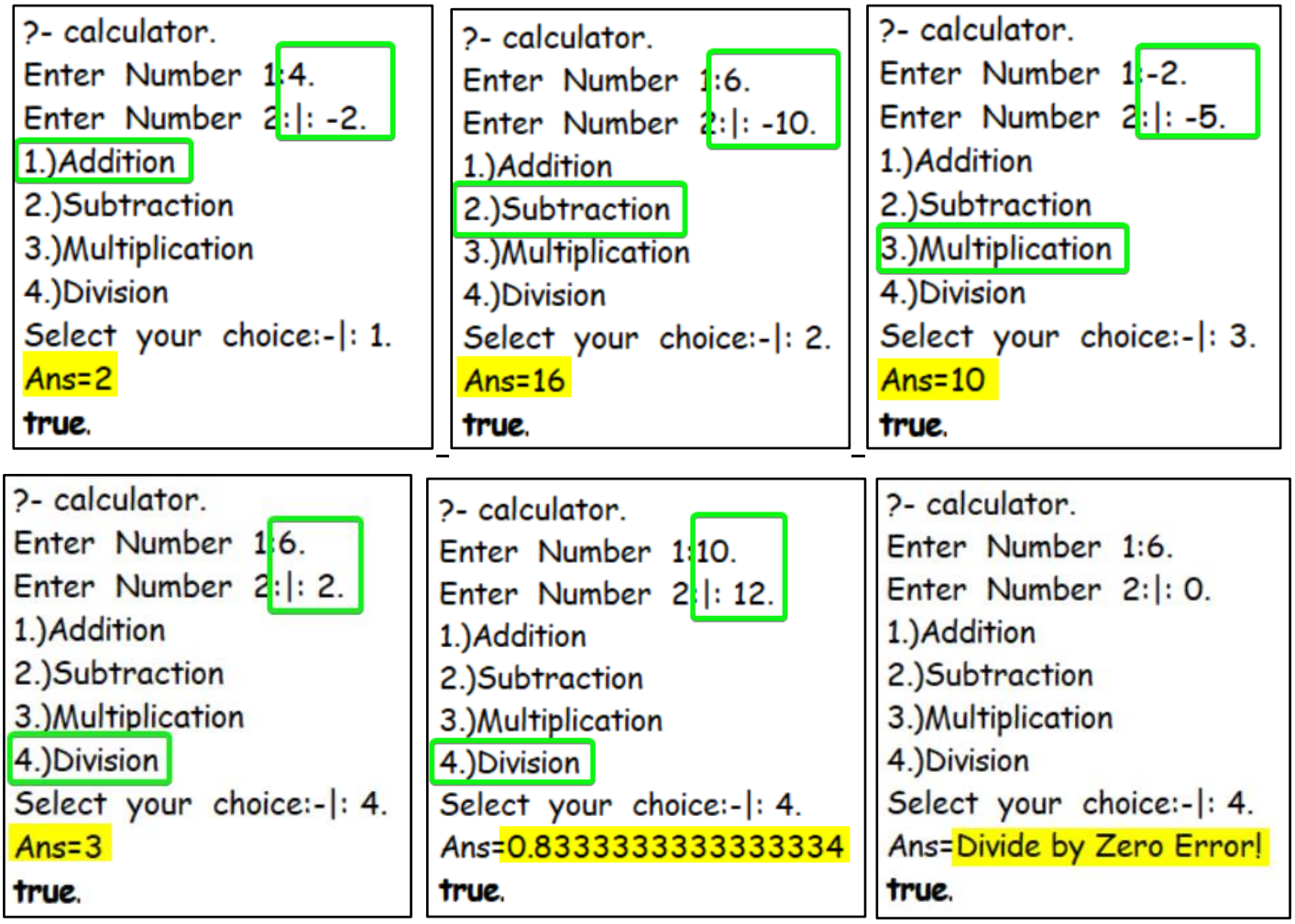
1.) Write a Prolog program to implement a Menu Driven **Calculator**.

**PROLOG Code**

```
calculator:-
    write("Enter  Number  1:"), read(A),
    write("Enter  Number  2:"), read(B),
    write("1.)Addition  \n2.)Subtraction  \n3.)Multiplication  \n4.)Division\n"),
    write("Select  your  choice:-"),
    read(Choice),
    write("Ans="),
    calculate(A,B,Choice).

    calculate(A,B,Choice):-
    (
        Choice = 1,
        Ans  is  A+B,
        write(Ans),!
    ).

    calculate(A,B,Choice):-
    (
        Choice = 2,
        Ans  is  A-B,
        write(Ans),!
    ).

    calculate(A,B,Choice):-
    (
        Choice = 3,
        Ans  is  A*B,
        write(Ans),!
    ).

    calculate(A,B,Choice):-
    (
        Choice = 4,
        (
            B =:= 0,
            write("Divide by Zero Error!\n"),
            !
        );
        (
```

```
            Ans  is  A/B,
            write(Ans),
            !
        )
    ).
```

**Output**




2.) Write a Prolog program to find **Maximum** and **Minimum** salaries of given 3 employees.

**PROLOG Code**

```
min_max_salary(X,Y,Z,Min,Max):-
    (
        (X>Y) ->
        (
            (Y>Z) ->
                (Max=X,Min=Z)
            ;
                (X>Z) ->
                    (Max=X,Min=Y)
```

```
            ;
                (Max=Z,Min=Y)
    )
    ;
    (
        (X>Z) ->
            (Max=Y,Min=Z)
        ;
            (Y>Z) ->
                (Max=Y,Min=X)
            ;
                (Max=Z,Min=X)
    )

).
```

**Output**

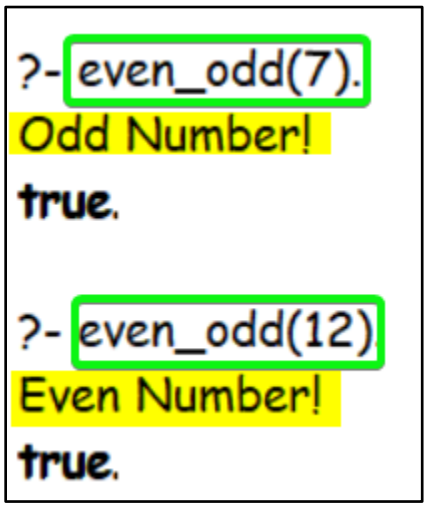
```
?- min_max_salary(10,20,30,Min,Max).
Min = 10,
Max = 30.

?- min_max_salary(10000,15000,25000,Min,Max).
Min = 10000,
Max = 25000.

?- min_max_salary(100000,5000,20000,Min,Max).
Min = 5000,
Max = 100000.

?- min_max_salary(100000,5000,200,Min,Max).
Min = 200,
Max = 100000.

?- min_max_salary(100,50000,200,Min,Max).
Min = 100,
Max = 50000.
```

3.) Write a Prolog program to check whether a given number is **Odd** or **Even**.

**PROLOG Code**

```
even_odd(X):-
    Mod is X mod 2,
    (
        Mod  =  0,
        write('Even Number!'),
        !
        ;
        write('Odd Number!')
    ).
```

**Output**




4.) Write a prolog program to check whether a given year is a **Leap year** or **not**.

**PROLOG Code**

```
is_leap_year(Year):-
    write(Year),
    (0 is Year  mod  400) ->
         write(" is a Leap Year!")
    ;
    (
        (0 is Year mod 100) ->
             write(" is Not a Leap Year!\n")
        ;
        (
            (0 is Year mod 4) ->
                 write(" is a Leap Year!\n")
            ;
             write(" is Not a Leap Year!\n")
        )
    ).
```

**Output**

```
?- is_leap_year(1600).
1600 is a Leap Year!
true.

?- is_leap_year(1900).
1900 is Not a Leap Year!
true.

?- is_leap_year(2020).
2020 is a Leap Year!
true.

?- is_leap_year(2023).
2023 is Not a Leap Year!
true.
```

5.) Write a prolog program to give **Grade** to a student based on total marks given:

| Marks Range | Grade |
|---|---|
| **80-100** | A |
| **60-79** | B |
| **36-59** | C |
| **1-35** | D |

**PROLOG Code**

```
grade(Marks,Grade):-
    Marks >= 80,
    Grade  =  'A',!.

grade(Marks,Grade):-
    Marks >= 60,
    Grade  =  'B',!.

grade(Marks,Grade):-
    Marks >= 36,
    Grade  =  'C',!.

grade(Marks,Grade):-
```

```
    Marks >= 1,
    Grade  =  'D',!.
```

**Output**

```
?- grade(95,Grade).
Grade = 'A'.

?- grade(75,Grade).
Grade = 'B'.

?- grade(55,Grade).
Grade = 'C'.

?- grade(35,Grade).
Grade = 'D'.
```

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**.

**PROLOG Code**

```
is_square(L,L).
```

**Output**

```
?- is_square(5,5).
true.

?- is_square(2,3).
false.
```

7.) Write a Prolog program to calculate the **Roots of Quadratic Equation**. Consider all possibilities **Real, Equal, Imaginary**.

# PROLOG Code

```
solve(A,B,C,L):-
    write("For  Equation  "),
    write(A),
    write("*x^2  +  "),
    write(B),
    write("x  +"),
    write(C),
    write("  =0\n"),
    D  is  (B^2)  -  (4*A*C),
    write("D = "),write(D),nl,
    roots(A,B,C,D,L).

    roots(A,B,_C,D,L):-
        (
            D == 0,
            write("Equal  Real  Roots!"),nl,
            X  is  (-B/(2*A)),
            L=[X],!
        ).

    roots(A,B,_C,D,L):-
        (
            D > 0,
            write("Real  Roots!"),nl,
            S is sqrt(D),
            X1  is  (-B  +  S)/(2*A),
            X2  is  (-B  -  S)/(2*A),
            L=[X1,X2],!
        ).

    roots(A,B,_C,D,L):-
        (
            D<0,
            write("Imaginary  Roots!"),nl,
            S  is  sqrt(-D)/(2*A),
            X1  is  -B/(2*A),
            L=[[X1,S],[X1,-S]],!
        ).
```

## Output

```
?- solve(1,-7,10,Roots).
For Equation 1*x^2 + -7x +10 =0
D = 9
Real Roots!
Roots = [5.0, 2.0].

?- solve(10,7,10,Roots).
For Equation 10*x^2 + 7x +10 =0
D = -351
Imaginary Roots!
Roots = [[-0.35, 0.9367496997597596], [-0.35, - 0.9367496997597596]].

?- solve(1,-10,25,Roots).
For Equation 1*x^2 + -10x +25 =0
D = 0
Equal Real Roots!
Roots = [5].
```

a = 1, b = -7, c = 10 (Example)

$1x^2 + -7x + 10 = 0$

$x^2 - 7x + 10 = 0$

Roots:

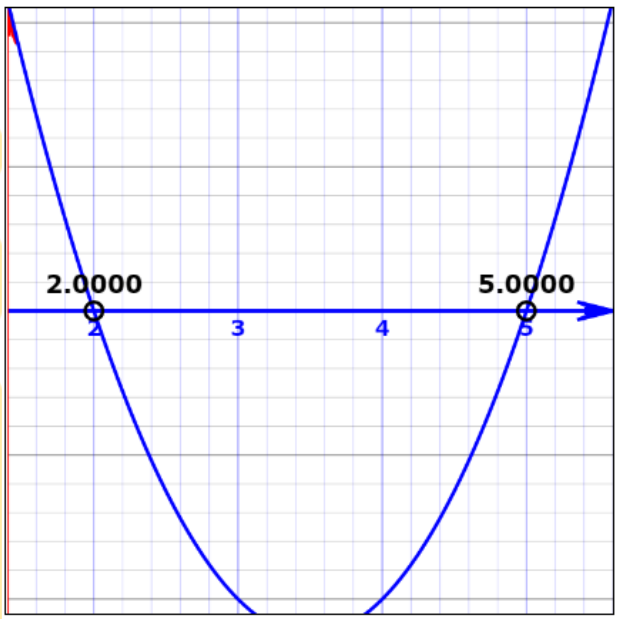
5
2

Details:

$x^2 - 7x + 10 = 0$

Roots: 5, 2
Root Pair: 7/2 ± 3/2
Factored: f(x) = (x - 5)(x - 2)

Discriminant: 9
Vertex: (3.5, -2.25)
Sum of Roots (-b/a): 7
Product of Roots (c/a): 10




Values accurate to 12 places only, and 4 places in graph.

a = 10, b = 7, c = 10 Example

$10x^2 + 7x + 10 = 0$

Roots:

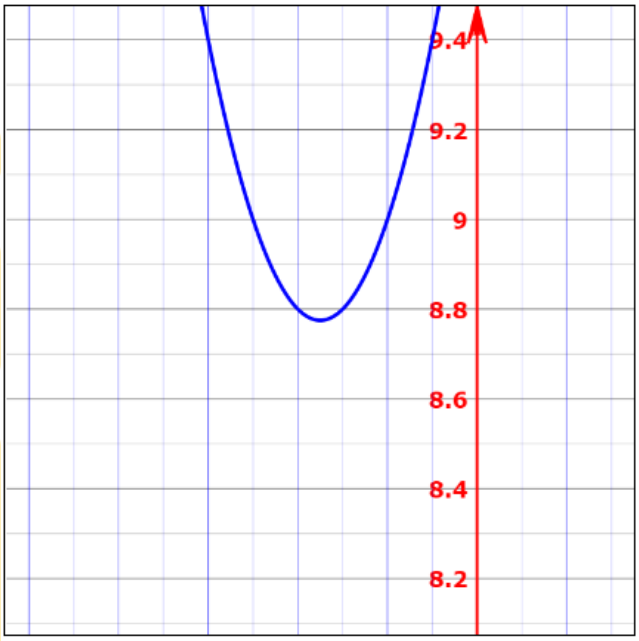
**-0.35 + 0.93674969975976i**
**-0.35 - 0.93674969975976i**
**It has Complex Roots !**

Details:

**$10x^2 + 7x + 10 = 0$**
**It has Complex Roots !**

**Roots: -0.35 + 0.93674969975976i, -0.35 - 0.93674969975976i**
**Root Pair: $-0.35 \pm 3\sqrt{(39)}/20i$**
**Factored: f(x) = 10(x + 0.35 + 0.93674969975976i)(x + 0.35 - 0.93674969975976i)**

**Discriminant: -351**
**Vertex: (-0.35, 8.775)**
**Sum of Roots (-b/a): -0.7**
**Product of Roots (c/a): 1**




Values accurate to 12 places only, and 4 places in graph.

a = 1, b = -10, c = 25 Example

$x^2 - 10x + 25 = 0$

Roots:

**5**

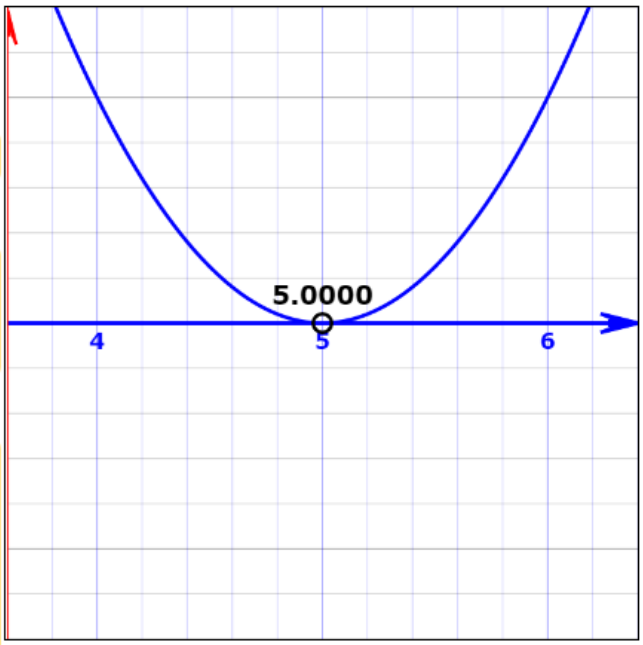
**Only one root.**

Details:

**$x^2 - 10x + 25 = 0$**
**Only one root.**

**Root: 5**
**Factored: $f(x) = (x - 5)^2$**

**Discriminant: 0**
**Vertex: (5, 0)**
**Sum of Roots (-b/a): 10**
**Product of Roots (c/a): 25**




Values accurate to 12 places only, and 4 places in graph.

8.) Write a Prolog program to find the number whether the number is **Positive, Negative or Zero**.

**PROLOG Code**

```
positive_negative(0):-
    write('Zero'),!.

positive_negative(X):-
    X > 0,
    write('Positive'),!.

positive_negative(X):-
    X < 0,
    write('Negative'),!.
```

**Output**

```
?- positive_negative(10).
Positive
true.

?- positive_negative(0).
Zero
true.

?- positive_negative(-10).
Negative
true.
```

**SUBMITTED BY**: U19CS012

BHAGYA VINOD RANA