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,

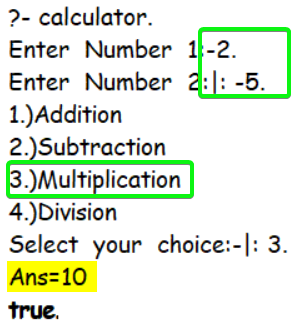
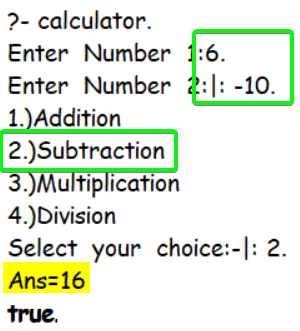
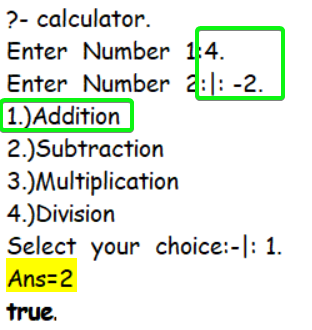
            write(Ans),

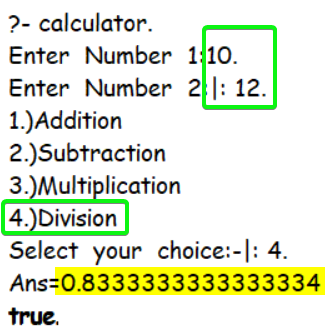
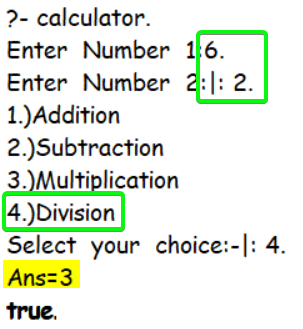
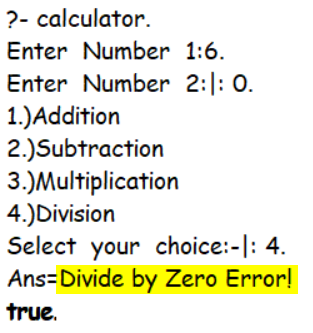
*!*

        )

    )*.*

**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)

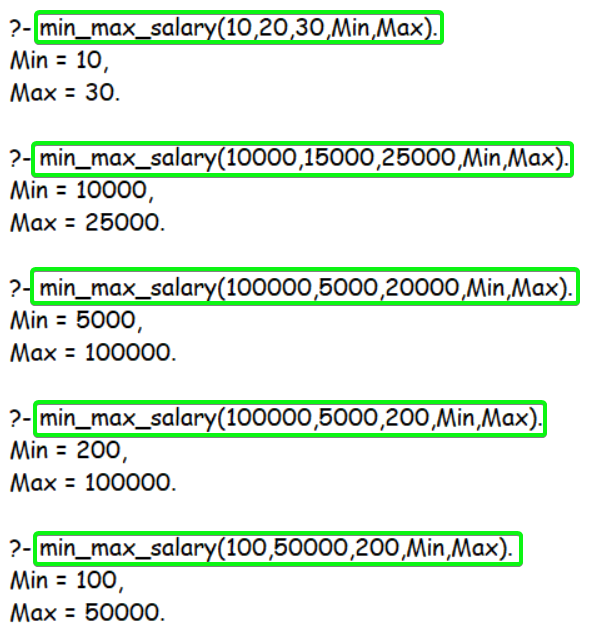
*;*

                    (Max=Z,Min=X)

        )

    )*.*

**Output**



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!'),

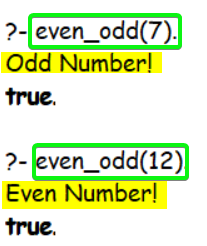
*!*

        ;

        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")

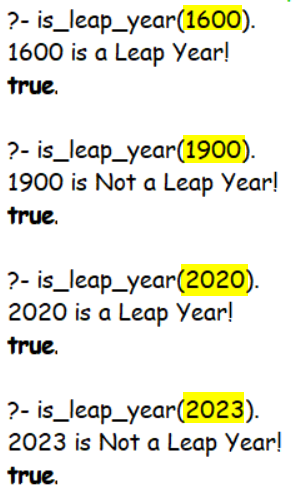
            ;

             write(" is Not a Leap Year!\n")

        )

    )*.*

**Output**



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,

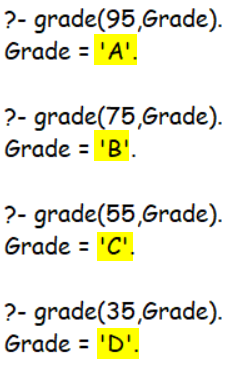
    Grade  =  'C',*!.*

grade(Marks,Grade)*:-*

    Marks >= 1,

    Grade  =  'D',*!.*

**Output**

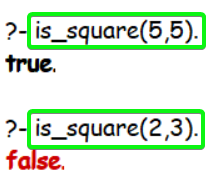


6.) Write a Prolog program to take values of Length and Breadth of a Rectangle from the user and check if it is **square or not**.

**PROLOG Code**

is\_square(L,L)*.*

**Output**



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,

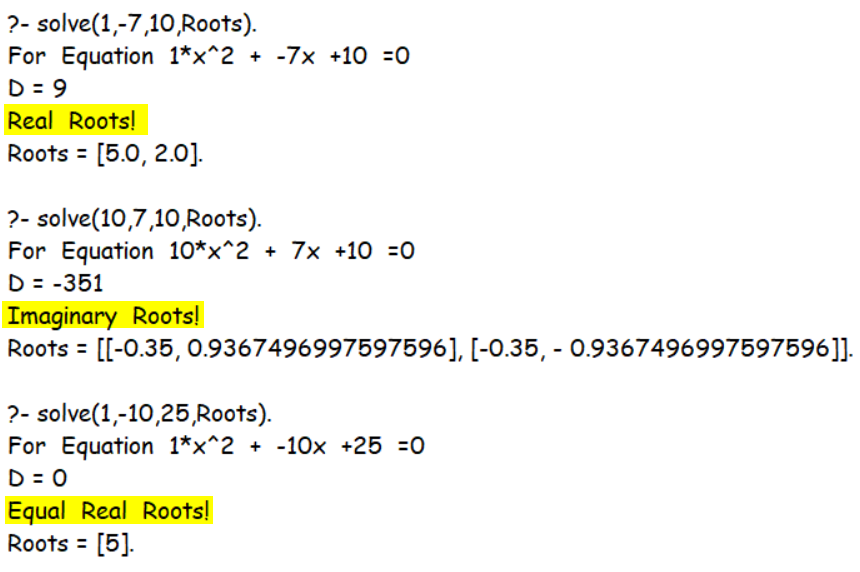
            S  is  sqrt(-D)/(2\*A),

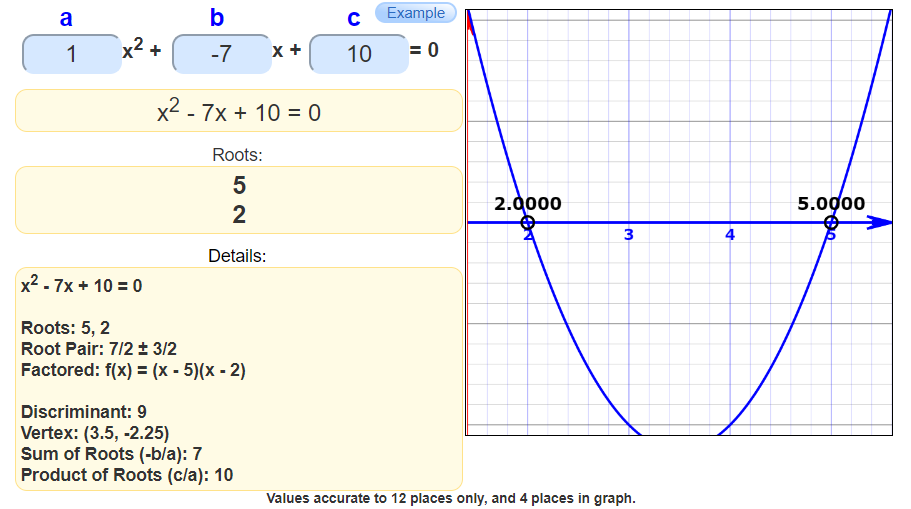
            X1  is  -B/(2\*A),

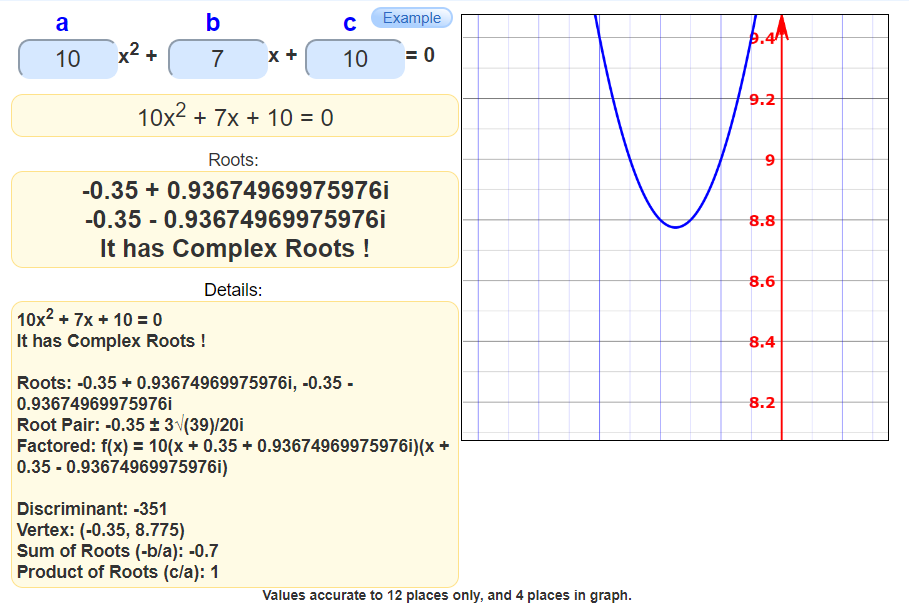
            L=[[X1,S],[X1,-S]],*!*

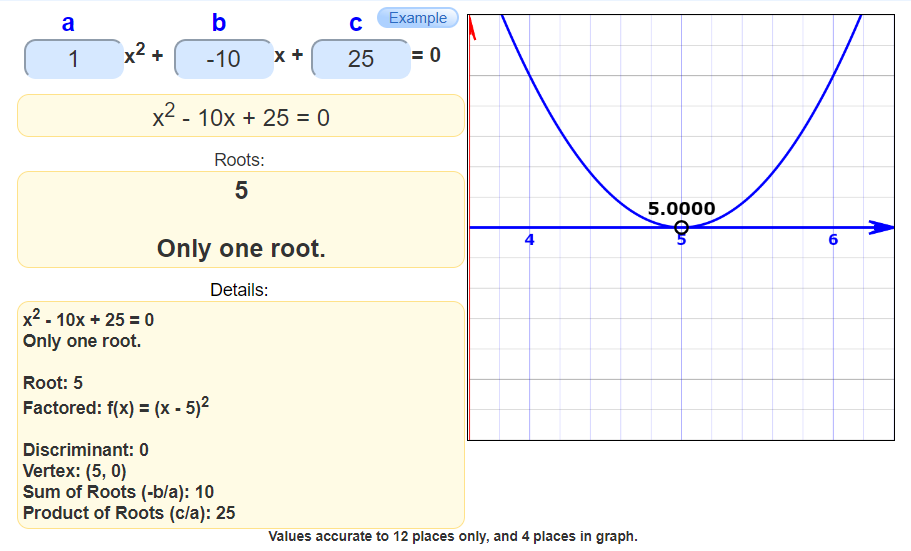
        )*.*

**Output**









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,

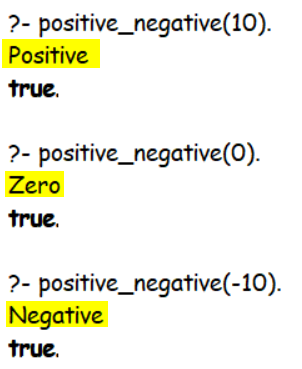
    write('Positive'),*!.*

positive\_negative(X)*:-*

    X < 0,

    write('Negative'),*!.*

**Output**



**SUBMITTED BY**: U19CS012

BHAGYA VINOD RANA