

PPL Assignment 8

Name: Himani Verma

Admission No: U19CS075

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

Source Code:

```
menu :-
    nl,
    write(' 1. Addition '),nl,
    write(' 2. Subtraction '),nl,
    write(' 3. Multiplication '),nl,
    write(' 4. Division '),nl,
    write(' 5. Exit '),nl,
    write('Enter your choice: '),
    read(Ch),
    case(Ch),
    menu.

case(1):-
    write('Enter the first number: '),read(A),
    write('Enter the second number: '), read(B),nl,
    X is A+B,
    write('Sum is : '),write(X),nl.

case(2):-
    write('Enter the first number: '),read(A),
    write('Enter the second number: '), read(B),nl,
    X is A-B,
    write('Difference is : '),write(X),nl.

case(3):-
    write('Enter the first number: '),read(A),
    write('Enter the second number: '), read(B),nl,
    X is A*B,
    write('Product is : '),write(X),nl.

case(4):-
    write('Enter the first number: '),read(A),
    write('Enter the second number: '), read(B),nl,
    B =\= 0,
    X is A/B,
    write('Division is : '),write(X),nl.

case(exit):-
    halt(0).
```

Output:

```
himani@Himani:~/Desktop/PPL$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
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).

?- consult('assgn8q1.pl').
true.

?- menu.

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 1.
Enter the first number: |: 10.
Enter the second number: |: 2.

Sum is : 12

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: |: 2.
Enter the first number: |: 10.
Enter the second number: |: 11.

Difference is : -1

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: |: 3.
Enter the first number: |: 4.
Enter the second number: |: 2.

Product is : 8

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: |: 4.
Enter the first number: |: 16.
Enter the second number: |: 4.

Division is : 4

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: |: exit.
himani@Himani:~/Desktop/PPL$
```

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

Source Code:

run:-

```

write('Enter Salary of Person A: '),
read(A),
write('Enter Salary of Person B: '),
read(B),
write('Enter Salary of Person C: '),
read(C),
max(A,B,C),
min(A,B,C).

max(A,B,C):-
( A>=B, A>=C -> write('A has max salary'),nl;
  B>=A, B>=C -> write('B has max salary'),nl;
  write('C has max salary'),nl).

min(A,B,C):-
( A<=B, A<=C -> write('A has min salary'),nl;
  B<=A, B<=C -> write('B has min salary'),nl;
  write('C has min salary'),nl).

```

Output:

```

himani@Himani:~/Desktop/PPL$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
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).

?- consult('assgn8q2.pl').
true.

?- run.
Enter Salary of Person A: 22000.
Enter Salary of Person B: |: 55000.
Enter Salary of Person C: |: 100000.
C has max salary
A has min salary
true.

```

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

Source Code:

```

run:-
write('Enter a Number: '),
read(N),
( N mod 2:=0 -> write('Even '),nl;
  write('Odd '),nl).

```

Output:

```

himani@Himani:~/Desktop/PPL$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
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).

?- consult('assgn8q3.pl').
true.

?- run.
Enter a Number: 3.
Odd
true.

?- run.
Enter a Number: 22.
Even
true.

?-

```

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

Source Code:

```

leapYear :-
    write('Enter the year: '),
    read(Year),
    check(Year).

check(Y):-
    (
        Y mod 4 == 0, Y mod 100 /= 0 ->
            write('Is a leap year'),nl;
        Y mod 400 == 0 ->
            write('Is a leap year'),nl;
        % Else
            write('Not a leap year'),nl).

```

Output:

```
himani@Himani:~/Desktop/PPL$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
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).

?- consult('ass8q4.pl').
true.

?- leapYear.
Enter the year: 2012.
Is a leap year
true.

?- leapYear.
Enter the year: 1900.
Not a leap year
true.

?- leapYear.
Enter the year: 1600.
Is a leap year
true.
```

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

00 - 80 Grade A

60 - 79 Grade B

35 - 59 Grade C

1 - 35 Grade D

Source Code:

```
run:-
write('Enter Total Marks: '),
read(M),
( M>=80, M<100 -> write('Grade A'),nl;
  M>=60, M<79 -> write('Grade B'),nl;
  M>=35, M<59 -> write('Grade C'),nl;
  M>=1, M<35 -> write('Grade D'),nl).
```

Output:

```
himani@Himani:~/Desktop/PPL$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
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).

?- consult('assgn8q5.pl').
true.

?- run.
Enter Total Marks: 74.
Grade B
true.

?- run.
Enter Total Marks: 96.
Grade A
true.

?- run.
Enter Total Marks: 48.
Grade C
true.

?- run.
Enter Total Marks: 2.
Grade D
true.

?- 
```

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.

Source Code:

```
run:-
write('Enter length: '),
read(L),
write('Enter breadth: '),
read(B),
(L == B ->
    write(' It is a square! '),nl
    ;

    write(' It is a rectangle! '),nl).
```

Output:

```

himani@Himani:~/Desktop/PPL$ gedit assgn8q6.pl
^C
himani@Himani:~/Desktop/PPL$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
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).

?- consult('assgn8q6.pl').
true.

?- run.
Enter length: 2.
Enter breadth: |: 3.
It is a rectangle!
true.

?- run.
Enter length: 2.
Enter breadth: |: 2.
It is a square!
true.

```

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

Source Code:

```

run:-
    write(" ***** Ax^2 + Bx + C = 0 ***** "),nl,
    write("Enter the value of A : " ),
    read(A),
    write("Enter the value of B : " ),
    read(B),
    write("Enter the value of C : " ),
    read(C),
    B1 is B*B,
    B2 is 4*A*C,
    D is B1-B2,
    root(A,B,C,D).

root(A,B,C,D):-
    D:=0,
    A\=0,
    write("Both roots are same: "),
    ANS is (-B/2*A),
    format('~2f', [ANS]),nl
    ;

    D>0,
    A \= 0,
    Q is (2*A),
    SQ1 is sqrt(D),
    ANS is (-B - SQ1) /Q,

```

```

ANS1 is  $(-B + \text{SQ1}) / Q$ ,
write("First root is : "),
format('~2f', [ANS]),nl,
write("Second root is : "),
format('~2f', [ANS1]),nl
;

D<0,
A \= 0,
SQ2 is  $\text{sqrt}(-D)$ ,
REAL is  $(-B/2*A)$ ,
IMG is  $\text{SQ2} / (2*A)$ ,
write("Real root is : "),
format('~2f', [REAL]),nl,
write("Imaginary root is : "),
format('~2f', [IMG]),write("i"),nl
;

A:=0,
write("Only one root exists : "),
ANS is  $(-C/B)$ ,
format('~2f', [ANS]),nl.

```

Output:


```
himani@Himani:~/Desktop/PPL$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
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).

?- consult('assgn8q7.pl').
true.

?- run.
***** Ax^2 + Bx + C = 0 *****
Enter the value of A : 0.
Enter the value of B : |: 1.
Enter the value of C : |: 1.
Only one root exists : -1.00
true.

?- run.
***** Ax^2 + Bx + C = 0 *****
Enter the value of A : 1.
Enter the value of B : |: -2.
Enter the value of C : |: 1.
Both roots are same: 1.00
true .

?- run.
***** Ax^2 + Bx + C = 0 *****
Enter the value of A : 1.
Enter the value of B : |: -4.
Enter the value of C : |: 2.
First root is : 0.59
Second root is : 3.41
true .

?- run.
***** Ax^2 + Bx + C = 0 *****
Enter the value of A : 1.
Enter the value of B : |: 1.
Enter the value of C : |: 1.
Real root is : -0.50
Imaginary root is : 0.87i
true .
```

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

Source Code:

```
pos_neg(X) :- X>0, write("Number is Positive \n").
pos_neg(X) :- X:=0, write("Number is Zero \n").
pos_neg(_) :- write("Number is Negative \n").
```

Output:

```
himani@Himani:~/Desktop/PPL$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
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).

?- consult('ass8q8.pl').
true.

?- pos_neg(8).
Number is Positive
true .

?- pos_neg(-2).
Number is Negative
true.

?- pos_neg(0).
Number is Zero
true .
```

□