Decision, Branching, and Loop

Example 1:

Prolog code for logicand.pl:

```
logicand-:
 write('Logic And Gate'),nl،
 write('A     B     ->   Z'),nl،
 write('0     0          0'),nl،
 write('0     1          0'),nl،
 write('1     0          0'),nl،
 write('1     1           1'),nl،
 write('Enter input A[0 or 1]: '),read(A),nl،
 write('Enter input B[0 or 1]: '),read(B),nl،
 processand(A,B.(
processand(A,B-:(
 A=0 , B=0 -> Z=0, write('A= '),write(A),write(' B= '),write(B،(
 write(' Z= '),write(Z),nl,askme؛
 A=0 , B=1 -> Z=0, write('A= '),write(A),write(' B= '),write(B،(
 write(' Z= '),write(Z),nl,askme؛
 A=1 , B=0 -> Z=0,write('A= '),write(A),write(' B= '),write(B،(
 write(' Z= '),write(Z),nl,askme؛
 A=1 , B=1 -> Z=1,write('A= '),write(A),write(' B= '),write(B،(
 write(' Z= '),write(Z),nl,askme.
askme:-
```

```
 write('to continue enter key c or any other key to exit: '),read(CH),

 processloop(CH).

processloop(CH):-

 CH=c -> logicand;

 write('Thank you').
```

Consulting and running:

```
?- ['logicand.pl'].

% logicand.pl compiled 0.00 sec, 264 bytes

true.

?- logicand.

Logic And Gate

A     B     ->   Z

0     0          0

0     1          0

1     0          0

1     1           1

Enter input A[0 or 1]: 1.

Enter input B[0 or 1]: 0.

A= 1 B= 0 Z= 0

to continue enter key c or any other key to exit: c.

Logic And Gate

A     B     ->   Z

0     0          0

0     1          0
```

1 0 0

1 1 1

Enter input A[0 or 1]: 1.

Enter input B[0 or 1]: 1.

A= 1 B= 1 Z= 1

to continue enter key c or any other key to exit: x.

Thank you

true.

Example 2:

Prolog code of mathop.pl:

```
mathop:-
 write('Select one form the following Math operations: '),nl,
 write('   1. Addition.'),nl,
 write('   2. Subtraction.'),nl,
 write('   3. Multiplication.'),nl,
 write('   4. Division.'),nl,
 write('   5. Mean.'),nl,
write('Select an operation [1, 2, 3, 4, 5]: '),read(N),processop(N).
processop(N):-
 N=1->write('Enter A value: '), read(A),
      write('Enter B value: '),read(B),
      write(A),write('+'),write(B),write('='),C is A+B,write(C),askme;
 N=2->write('Enter A value: '), read(A),
      write('Enter B value: '),read(B),
```

```
    write(A),write('-'),write(B),write('='),C is A-B,write(C),askme;
 N=3->write('Enter A value: '), read(A),
    write('Enter B value: '),read(B),
    write(A),write('*'),write(B),write('='),C is A*B,write(C),askme;
 N=4->write('Enter A value: '), read(A),
    write('Enter B value: '),read(B),
    write(A),write('/'),write(B),write('='),C is A/B,write(C),askme;
 N=5->write('Enter A value: '), read(A),
    write('Enter B value: '),read(B),
    write('The mean of '),write(A),write(' and '),write(B),
    write(' is '),C is (A+B)/2,write(C),askme;
 N<1->mathop;
 N>5->mathop.
askme:-
 nl,write('Enter key c/C to continue or any other key to exit: '),
 read(CH),processmath(CH).
processmath(CH):-
 CH=c->mathop;
 CH='C'->mathop;
 write('Thank you.').
```

Consulting and running:

```
?- ['mathop.pl'].
% mathop.pl compiled 0.00 sec, 8 bytes
true.
```

?- mathop.

Select one form the following Math operations:

1. Addition.
2. Subtraction.
3. Multiplication.
4. Division.
5. Mean.

Select an operation [1, 2, 3, 4, 5]: 4.

Enter A value: 2.

Enter B value: 3.

2/3=0.666667

Enter key c/C to continue or any other key to exit: C.

Select one form the following Math operations:

1. Addition.
2. Subtraction.
3. Multiplication.
4. Division.
5. Mean.

Select an operation [1, 2, 3, 4, 5]: 4.

Enter A value: 2.

Enter B value: 3.

2/3=0.666667

Enter key c/C to continue or any other key to exit: x.

Thank you.

```
true.
```