```
add(X, Y, Z):- Z is X + Y.
```

```
sub(X, Y, Z):- Z is X - Y.
```

```
multi(X, Y, Z):- Z is X * Y.
```

```
div(X, Y, Z):- Z is X / Y.
```



```
is_even(X):- X mod 2 =:= 0.
```

```
is_odd(X):- X mod 2 \= 0.
```

```
largest_of_2(A, B, C):- C is max(A, B).
```

```
largest_of_3(A, B, C, D):-
    A > B, A > C ->  D is A;
    B > A, B > C ->  D is B;
    D is C.
```

```
smallest_of_3(A, B, C, D):-
    A < B, A < C ->  D is A;
    B < A, B < C ->  D is B;
    D is C.
```

```
largest([X],X).
largest([X, Y | Rest], Z):-
    X >= Y,
    largest([X | Rest], Z).
largest([X, Y | Rest], Z):-
    X < Y,
    largest([Y | Rest], Z).
```

```
smallest([X],X).
smallest([X, Y | Rest], Z):-
    X < Y,
    smallest([X | Rest], Z).
smallest([X, Y | Rest], Z):-
    X >= Y,
    smallest([Y | Rest], Z).
```

```
gcd(X, 0, X):- !.
gcd(X, Y, Z):-
    R is X mod Y,
```

```
    gcd(Y, R, Z).

lcm(X, Y, Z):-
    R is X * Y,
    gcd(X, Y, R1),
    Z is R / R1.
```

```
factorial(0, 1).
factorial(X, Z):-
    X > 0,
    R is X - 1,
    factorial(R, Z1),
    Z is X * Z1.
```

```
is_prime(2).
is_prime(3).
is_prime(N) :-
    N > 3,
    \+ has_divisor(N, 2).

has_divisor(N, D) :-
    N mod D =:= 0.

has_divisor(N, D) :-
    D < N - 1,
    D1 is D + 1,
    has_divisor(N, D1).
```

```
parent_of(john,james).
parent_of(john,mary).
parent_of(jack,john).
parent_of(jack,hina).
parent_of(jill,john).
parent_of(jill,hina).
male(james).
male(john).
male(jack).
female(mary).
female(hina).
female(jill).
father_of(X, Y):- male(Y),
    parent_of(X,Y).
mother_of(X, Y):- female(Y),
    parent_of(X,Y).
```