

Indian Institute of Engineering Science & Technology, Shibpur
Department of Computer Science & Technology
8th Semester Artificial Intelligence Laboratory 2025
CS 4271
ASSIGNMENT – 6

Write PROLOG Programs:

1. To find last element of a list.
2. To append two lists in a third list.
3. To reverse a list in another list.
4. To determine whether a list is a palindrome.

The structure of predicate:

`Palindrome(L)`

5. To find the kth element of a list.

Example:

```
?- element_at(X, [a,b,c,d,e], 3).  
X=c
```

6. To find the sum and average of all elements of a list using sum and length.
7. To find gcd of two integers.
8. To determine whether a given integer number is prime or not.

Example:

```
?- is_prime(7).  
true
```

9. To determine the prime factors of a given positive integer.

Construct a flat list containing the prime factors in ascending order.

Example:

```
?- prime_factors(315, L).  
L = [3,3,5,7]
```

10. To determine Goldbach's conjecture

Goldbach's conjecture says that every positive even number greater than 2 is the sum of two prime numbers. Example: $28 = 5 + 23$. It is one of the most famous facts in number theory that has not been proved to be correct in the general case. It has been numerically confirmed up to very large numbers. Write a predicate to find the two prime numbers that sum up to a give even integer.

Example:

```
?- goldbach(28, L).  
L = [5, 23]
```

11. To generate first N Fibonacci numbers.

12. Consider a database of facts that describe *parent* relationships as well as *gender* relationships. The predicate `parent(john, ann)` is interpreted as: "John is a parent of Ann". The predicate `male(john)` is interpreted as: "John is a man". The predicate `female(ann)` is interpreted as: "Ann is a woman". So an example database of facts is:

```
parent(john, ann).  
parent(jim, john).  
parent(jim, keith).  
parent(mary, ann).  
parent(mary, sylvia).  
parent(brian, sylvia).  
male(keith). male(jim).  
female(sylvia).  
female(ann).  
male(brian).
```

Note that some things are not specified in the database above (e.g., `male(john)`).

a) Write a Prolog predicate `uncle(X, Y)` that is true if `x` is `y`'s uncle. Note that we are not considering uncles "by marriage", meaning that for `x` to be `y`'s uncle the two must be related by blood. For instance (user input is in red):

```
?- uncle(keith, ann).  
Yes  
?- uncle(ann, mary).  
No  
?- uncle(keith, X).  
X = ann ;  
No  
?- uncle(john, ann).  
No  
?- uncle(X, Y).  
X = keith  
Y = ann;  
No
```

b) Write a Prolog predicate `halfsister(X, Y)` that is true if `x` is `y`'s half-sister. For instance (user input is in red):

```
?- halfsister(ann, sylvia).  
Yes  
?- halfsister(X, sylvia).  
X=ann ;  
No  
?- halfsister(X, Y).  
X=ann  
Y=sylvia ;
```

```
X=sylvia  
X=ann ;  
No
```