**Name: Brijesh Rameshbhai Rohit**

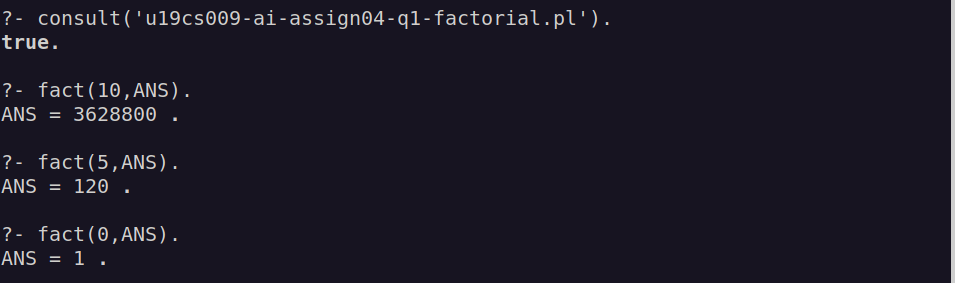
**Admission number: U19CS009**

**ARTIFICIAL INTELLIGENCE**

**ASSIGNMENT - 04**

**1. W.A.P.P to find factorial of a number.**

|  |
| --- |
| %U19CS009  %BRIJESH ROHIT  fact(0,1).  fact(N,F):-  N1 is N-1,  fact(N1,F1),  F is N\*F1. |



**2. W.A.P.P to print Fibonacci series.**

**The Fibonacci sequence f (1), f (2), f (3)...is:**

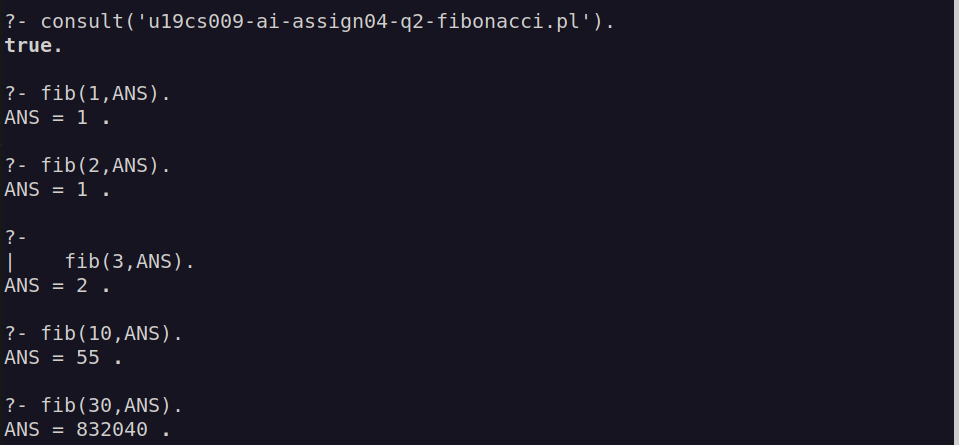
**1, 1, 2, 3, 5, 8, 13, 21, 34, 55....**

**Example:**

**?- fib (6, R).**

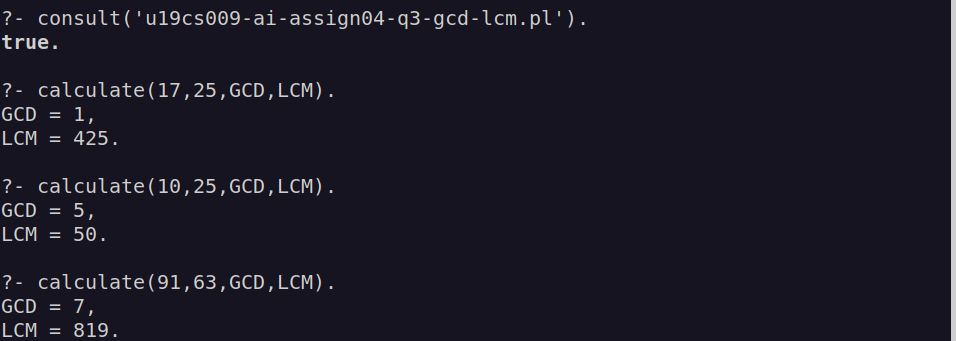
**R = 8**

|  |
| --- |
| %U19CS009  %BRIJESH ROHIT  fib(1,1).  fib(2,1).  fib(N,X):-  N1 is N-1,  N2 is N-2,  fib(N1,X1),  fib(N2,X2),  X is X1+X2. |



**3. W.A.P.P to finding the greatest common divider (GCD) and the least common multiple (LCM) of two integers.**

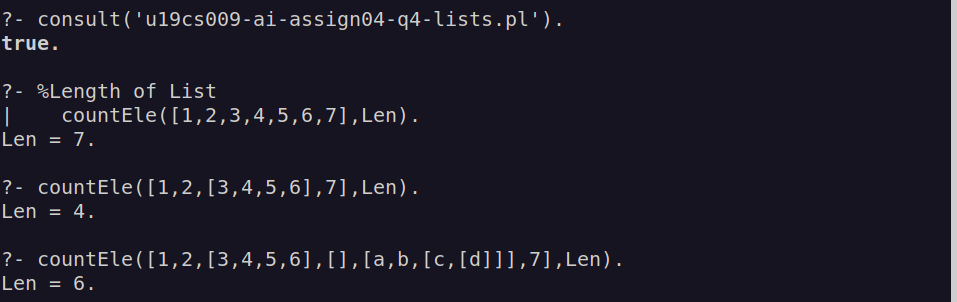
|  |
| --- |
| %U19CS009  %BRIJESH ROHIT  lcm(A,B,R):-  gcd(A,B,Res),  R is A\*B/Res.  gcd(0,A,A):-!.  gcd(A,0,A):-!.  gcd(A,B,R):-  B1 is mod(A,B),  gcd(B,B1,R).  calculate(A,B,GCD,LCM):-  gcd(A,B,GCD),  lcm(A,B,LCM). |



**4. W.A.P.P.**

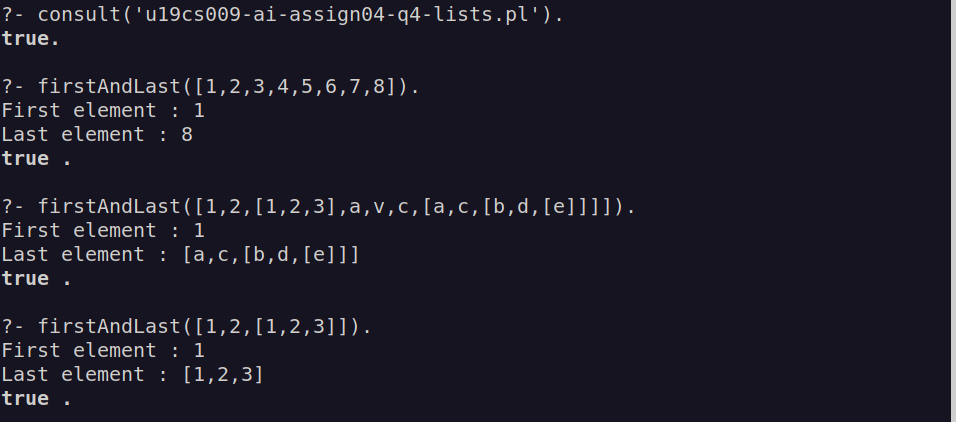
**A. To find length of the list.**

|  |
| --- |
| %U19CS009  %BRIJESH ROHIT  %count number of elements in list  countEle([],0).  countEle([\_|T],N):-  countEle(T,N1),  N is N1 + 1. |



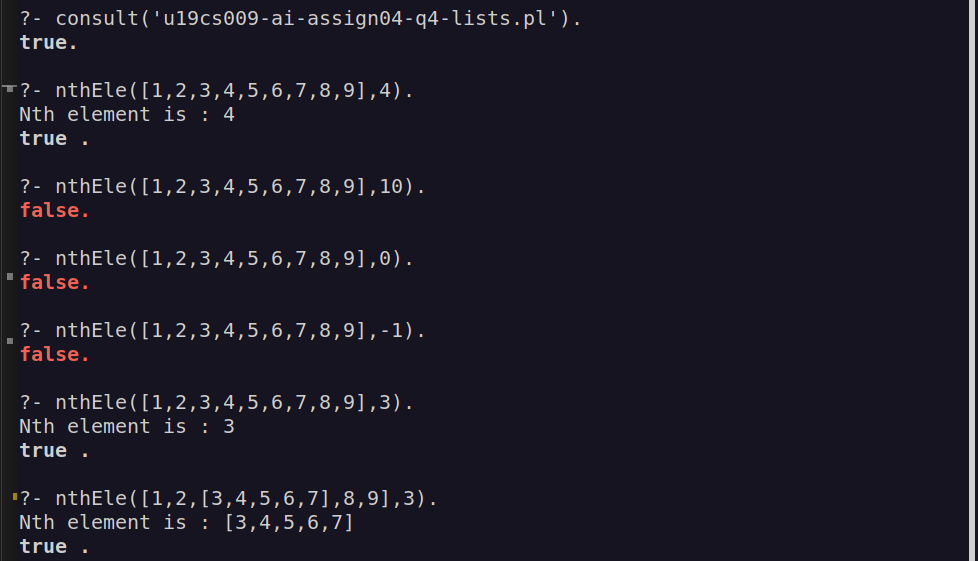
**B. To find first and last element of the list.**

|  |
| --- |
| %first element of list  firstEle([]).  firstEle(X,[H|\_]):-  X is H.  %last element  lastEle(X,[X]).  lastEle(X,[\_|L]):-  lastEle(X,L).  firstAndLast([H|T]):-  write("First element : "),  firstEle(X,[H|T]),  write(X),nl,  write("Last element : "),  lastEle(Y,[H|T]),  write(Y). |



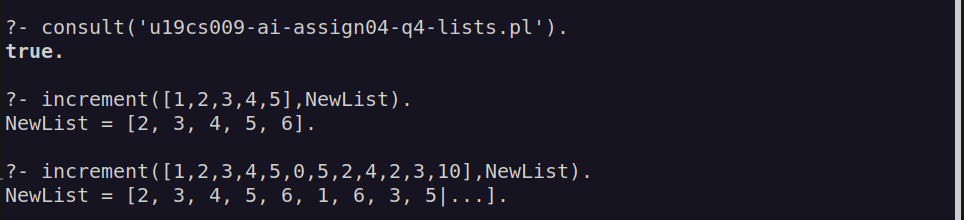
**C. To find the nth element of the list.**

|  |
| --- |
| %nth element in list  nthEle([],0):-  write("Invalid Argument!!"),nl.  nthEle([H|\_],1):-  write("Nth element is : "),  write(H).  nthEle([\_|T],N):-  N1 is N-1,  nthEle(T,N1). |



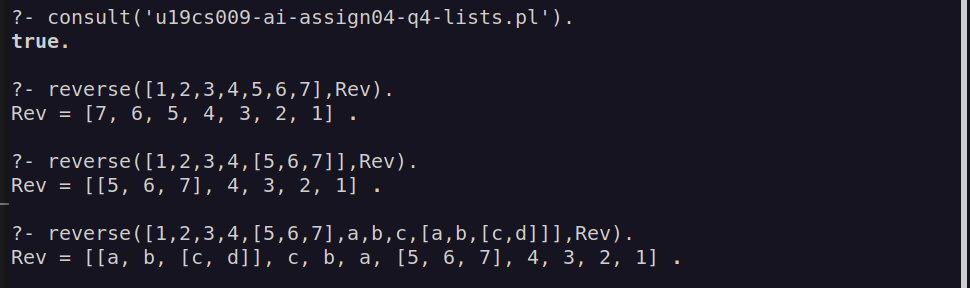
**D. To increment each number in the list.**

|  |
| --- |
| %increment each number in list  increment([],[]).  increment([H|T],[H1|T1]):-  increment(T,T1),  H1 is H+1. |



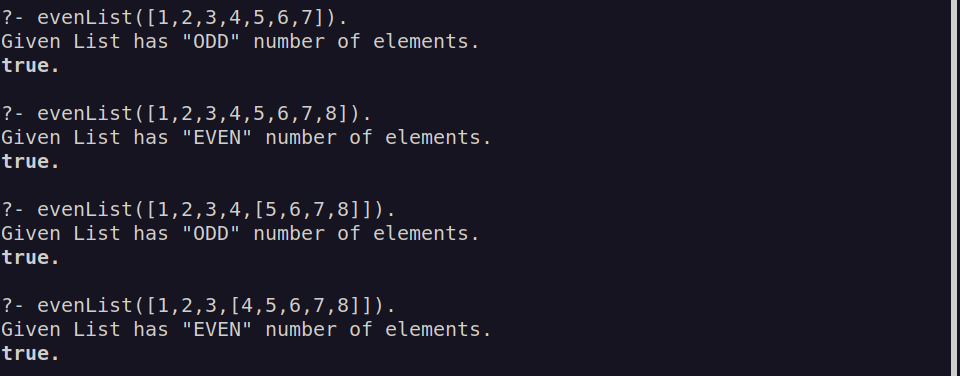
**E. To reverse the list.**

|  |
| --- |
| %add elemenet at end  addAtEnd(H,[],[H]).  addAtEnd(X,[H|T1],[H|T2]):-  addAtEnd(X,T1,T2).  %reverse a list  reverse([],[]).  reverse([H|T],Y):-  reverse(T,T1),  addAtEnd(H,T1,Y). |



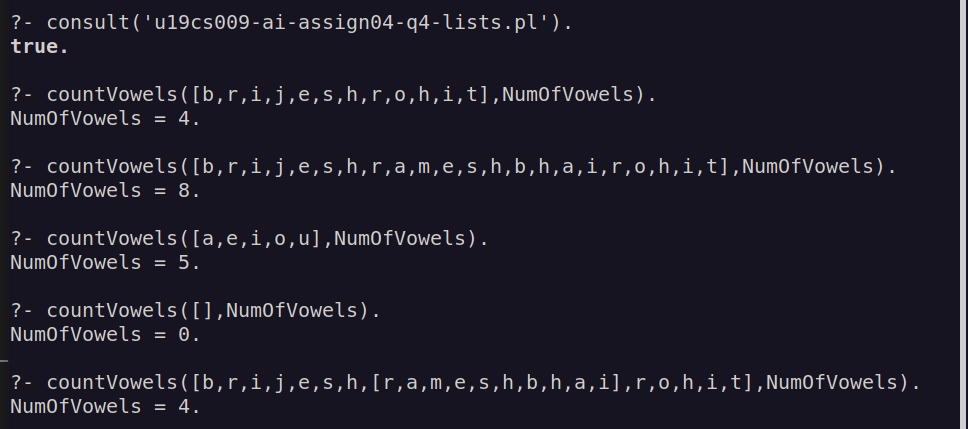
**F. To verify if a list has an even number of elements.**

|  |
| --- |
| %check even or odd integer  check\_even(N):-  Y is N//2,Y\*2=:=N  ->format('Given List has \"EVEN\" number of elements.~n');  format('Given List has \"ODD\" number of elements.~n').  %check if number of elements are even or odd  evenList([H|T]):-  countEle([H|T],N), %from first ques  check\_even(N). |



**G. To count vowels in the list**

|  |
| --- |
| %check vowels in list  vowels(X):-member(X,[a,e,i,o,u]).  countVowels([],0).  countVowels([H|T],N):-  vowels(H),  countVowels(T,N1),  N is N1+1,  !.  countVowels([\_|T],N):-  countVowels(T,N). |



**H. To remove duplicates from the list.**

|  |
| --- |
| %remove duplicates  remove\_duplicates([],[]).  remove\_duplicates([H | T], List) :-  member(H, T),  remove\_duplicates( T, List).  remove\_duplicates([H | T], [H|T1]) :-  \+member(H, T),  remove\_duplicates( T, T1). |

