practical 01

SWISH File Edit Examples Help

246 users online

Search

Program

```
/*AI practical program-1 sum of two numbers*/

sum(X,Y,Z):- Z is X+Y.

/* And sum of elements of a list*/

sumlist([],0).

sumlist([First | Rest], Sum) :-
    sumlist(Rest, SumRest),
    Sum is First + SumRest.
```

sum(8,20,Z).

Z = 28

sumlist([5,9,7],Sum).

Sum = 21

?- sumlist([5,9,7],Sum).

Examples History Solutions table results Run!

practical 02

SWISH File Edit Examples Help

206 users online

Search

Program Program +

```
1  /* pract2. write a prolog program to impliment max(X, Y, M) so that M is the
2    of two numbers X and Y
3  */
4
5  max(X,Y,M):- X>Y, M is X.
6  max(X,Y,M):- Y>=X, M is Y.
7
8  /*minimum of two numbers
9  */
10 min(X,Y,Mi):- X<Y, Mi is X.
11 min(X,Y,Mi):- Y<X, Mi is Y.
```

max(7,9,M).

M = 9

max(9,7,M).

M = 9

false

min(9,7,M).

M = 7

min(7,9,M).

M = 7

false

?- min(7,9,M).

Examples History Solutions table results Run!

practical 03

SWISH File Edit Examples Help

253 users online

Search

Program Program Program

```
/*3.write a program in prolog to impliment factorial(N, F) where F represents
 * the factorial of a number N.
*/

fact(0,1):-!.
fact(N,R):- N>0,
     N1 is N-1,
    fact(N1,R1),
    R is N*R1.
```

fact(9,R).

R = 362880

?- fact(9,R).

Examples History Solutions table results Run!

practical 04

SWISH File Edit Examples Help

108 users online

Search

Program Program Program Program

```
/* 4. write a program in prolog to impliment generate_fib(N,T)
 * where T represents the Nth term of the fibonacci series.
 * here N :- index of the fibonacci series element
*/

fib(1,0):-!.
fib(2,1):-!.
fib(N,T):- N>0,
    N1 is N-1,
    N2 is N-2,
    fib(N1,T1),
    fib(N2,T2),
    T is T1+T2.
```

T = 2

fib(5,T).

T = 3

fib(8,T).

T = 13

fib(6,T).

T = 5

fib(7,T).

T = 8

fib(9,T).

T = 21

?- fib(9,T).

Examples History Solutions table results Run!

practical 05

SWISH File Edit Examples Help

119 users online

Search

Program Program Program Program

Program +

```
1  /* pract5 . write a prolog program to impliment
2   * GCD(Greatest Common Divisor) of two numbers.
3  */
4
5  gcd(0,X,X):-!.
6  gcd(X,0,X):-!.
7  gcd(X,Y,R):- Y1 is mod(X,Y), gcd(Y, Y1,R).
```

gcd(89,75,R).

R = 1

gcd(88,75,R).

R = 1

gcd(88,77,R).

R = 11

?- gcd(88,77,R).

Examples History Solutions table results Run!

practical 06

SWISH File Edit Examples Help

158 users online

Search

Program Program Program Program

Program Program +

```
1  /* pract6. Write a prolog program to implement
2   * power(Num, Pow, Ans):where Num is raised to the power Pow to get Ans.
3  */
4
5  power(x,0):-!.
6  power(Num, Pow, Ans):-Ans is Num^Pow.
```

power(7,3,Ans).

Ans = 343

power(9,5,Ans).

Ans = 59049

?- power(9,5,Ans).

Examples History Solutions table results Run!

practical 07

SWISH File Edit Examples Help

151 users online

Search

Program Program Program Program

Program Program Program

```
/* pract7. Write a prolog program to implement multi(N1,N2,R): where N1
 * and N2 denotes the numbers to be multiplied and R represents the result.
*/

/*multi(X,0).*/
multi(N1, N2, R) :- R is N1*N2.
```

multi(9,25,R)

R = 225

multi(77,20,R)

R = 1540

?- `multi(77,20,R)`

Examples History Solutions table results Run!

practical 08

SWISH File Edit Examples Help

321 users online

Search

Program Program Program Program

Program Program

```
1 /*pract8. Write a prolog program to implement memb(X,L): to check whether
2  *X(element) is a member of L (list) or not.
3 */
4
5 is_list_member(X,[X|_]).
6 is_list_member(X,[_|TAIL]):- is_list_member(X,TAIL).
```

is_list_member(2,[1,2,3,4,5]).

true 1

false

is_list_member(6,[1,2,3,4,5]).

false

?- is_list_member(6,[1,2,3,4,5]).

Examples History Solutions table results Run!

practical 09

SWISH File Edit Examples Help 315 users online Search

Program Program Program Program Program +

```
1 /*pract9. Write a Prolog Program to implement conc(L1,L2,L3) where L2
2  *is the list to be appended with L1 to get the resulted list L3.
3 */
4
5 concat([H|T_L1],L2,[H|T_L3]) :- concat(T_L1, L2, T_L3).
6 concat([],L,L).
7
```

concat([1,2,3],[a,b,c],[1,2,3,a,b,c]).

true 1

concat([1,2,3],[a,b,c],[1,2,3,a,b,c,d]).

false

concat([1,2,3],[a,b,c],L).

L = [1, 2, 3, a, b, c]

concat([1,2,3],L,[1,2,3,a,b,c,d]).

L = [a, b, c, d]

concat(L,[a,b,c],[1,2,3,a,b,c]).

L = [1, 2, 3]
false

concat(L, X, [1,2,3,a,b,c]).

L = [1, 2, 3, a, b, c],
X = []
L = [1, 2, 3, a, b],
X = [c]

?- concat(L, X, [1,2,3,a,b,c]).

Examples History Solutions table results Run!

practical 10

SWISH File Edit Examples Help

366 users online

Search

Program Program Program Program

Program Program

```
/*pract10. Write a Prolog Program to implement reverse (L,R) where List L
 *is original and list R is reserved list.
*/

reverse_list([],[]).
reverse_list([H_L|T_L],Reversed):-
    reverse_list(T_L, ReversedTail),
    concatination(ReversedTail, [H_L], Reversed).

concatination([H_L1|T_L1],L2,[H_L1|T_L3]):-
    concatination(T_L1, L2, T_L3).
concatination([],L1, L1).
```

reverse_list([1,2,3],R).

R = [3, 2, 1]

reverse_list([1,2,3,4],R).

R = [4, 3, 2, 1]

reverse_list([1,2,3,4],[4,3,2,1]).

true 1

reverse_list([1,2,3,4],[4,3,1]).

false

?- reverse_list([1,2,3,4],[4,3,1]).

Examples History Solutions table results Run!

practical 11

SWISH File Edit Examples Help 327 users online Search

Program

```
/*pract11. Write a Prolog Program to implement palindrome (L) which
 *checks whether a list L is a palindrome or not.
*/

/* If we can read a it same in bacward and forward direction then we call it

is_palindrome(List) :- reverse_list(List,List).

reverse_list([],[]).
reverse_list([H_L|T_L],Reversed):-
    reverse_list(T_L, ReversedTail),
    concatination(ReversedTail, [H_L], Reversed).

concatination([H_L1|T_L1],L2,[H_L1|T_L3]):-
    concatination(T_L1, L2, T_L3).
concatination([],L1, L1).
```

```
is_palindrome([1,2,3,2,1]).
true
is_palindrome([1,2,3,2]).
false
```

```
?- is_palindrome([1,2,3,2]).
```

Examples History Solutions table results Run!

practical 12

SWISH File Edit Examples Help 354 users online Search

Program

```
1  /*pract12. Write a Prolog Program to implement sumlist(L,S) so that S is
2   *the Sum Of a given list L.
3  */
4
5  sumlist([],0).
6
7  sumlist([H_L | T_L], Sum) :-
8      sumlist(T_L, SumTail),
9      Sum is H_L + SumTail.
10
```

sumlist([2,4,6,20],X).

X = 32

sumlist([],Sum).

Sum = 0

sumlist([34,56,84,92],Sum).

Sum = 266

?- sumlist([34,56,84,92],Sum).

Examples History Solutions table results Run!

https://swish.swi-prolog.org/#tabbed-tab-0

practical 13

SWISH File Edit Examples Help

284 users online

Search

Program Program Program Program Program Program +

```
1  /* Pract13. Write a Prolog program to implement two predicates evenlength(Lis
2   * and oddlength(List) so that they are true if their argument is a
3   * list of even or odd length respectively.
4  */
5
6  evenlength([]).
7  evenlength([_|T]):-
8      oddlength(T).
9  oddlength([_|T]):-
10     evenlength(T).
11 evenoddlength:-
12     write("Enter the list to be checked: "),
13     read(L),
14     (evenlength(L)
15     ->write("The entered list is even length");
16     write("The entered list is odd length")),!.
```

```
false

evenlength([e,r,u,h]).
true 1

evenoddlength.
Enter the list to be checked:
    oddlength([a,b,c,d,e]).
The entered list is odd length
true 1

evenoddlength.
Enter the list to be checked:
    ([a,b,c,d]).
The entered list is even length
true 1

evenoddlength.
Enter the list to be checked:
    ([a,b,c,d,e]).
The entered list is odd length
true 1

?- evenoddlength.
```

Examples History Solutions table results Run!

practical 14

SWISH File Edit Examples Help 268 users online Search

Program Program Program Program Program Program +

```
1  /* pract14. write a prolog program to implement nth_element (N,L,X)
2   * where N is desired position, L is a list and X represents the Nth
3   * element of L.
4  */
5
6  nth_element(1, [X|_], X).
7  nth_element(K,[_|L],X):-
8      nth_element(K1,L,X),
9      K is K1+1.
10 nth_element:-
11     write("Enter the list: "),
12     read(L),
13     write("Enter the position of the element"),
14     read(N),
15     nth_element(N,L,X),
16     write("The element at position "),
17     write(N),write(" in the list is: "),
18     write(X),!.
```

https://swish.swi-prolog.org

nth_element.

Enter the list:

([2,4,3,6,1,8]).

Enter the position of the element

3

The element at position 3 in the list is: 3

true 1

nth_element.

Enter the list:

([a,b,h,i,s,h,e,k]).

Enter the position of the element

7

The element at position 7 in the list is: e

true 1

?- nth_element.

Examples History Solutions table results Run!

practical 15

SWISH File Edit Examples Help

277 users online

Search

Program Program Program Program

Program Program

```
1  /*Pract15. Write a Prolog program to implement maxlist(L, M) so that M is
2   * the maximum number in the list.
3  */
4
5
6  maxlist([X],X).
7  maxlist([H|T],M):-
8      maxlist(T,M1),
9      H<M1 -> M is M1;
10     M is H.
11 maxlist:-
12     write("Enter the list: "),
13     read(L),
14     maxlist(L,X),
15     write("The maximum element in the given list is: "),
16     write(X),!.
17
```

maxlist.

Enter the list:

([2,4,6,1,23,5,61,8,10]).

The maximum element in the given list is: 61

true 1

maxlist.

Enter the list:

([9,7,8,3,20,5,6]).

The maximum element in the given list is: 20

true 1

?- maxlist.

Examples History Solutions table results Run!

practical 16

SWISH File Edit Examples Help 274 users online Search

Program Program Program Program Program Program +

```
/* Pract16. Write a prolog program to implement insert_nth (I, N, L, R)
 * that inserts an item I into Nth position of list L to generate a list R.
*/

insert_nth(I, 1, L, [I|L]).
insert_nth(I, N, [H|T], [H|R]):-
    N1 is N-1,
    insert_nth(I, N1, T, R).
insert_nth:-
    write("Enter the list: "),
    read(L),
    write("Enter the position of the element to be inserted: "),
    read(N),
    write("Enter the element to be inserted: "),
    read(I),
    insert_nth(I,N,L,R),
    write("Final list after insertion of "),
    write(I),write(" at "),
    write(N),write(" position in the list is: "),
    write(R),!.
```

insert_nth.

```
Enter the list:
([a,b,c,d,e]).
Enter the position of the element to be inserted:
2
Enter the element to be inserted:
h
Final list after insertion of h at 2 position in the list is: [a, h, b, c, d, e]
true
```

insert_nth.

```
Enter the list:
([2,4,3,5,7,6]).
Enter the position of the element to be inserted:
5
Enter the element to be inserted:
90
Final list after insertion of 90 at 5 position in the list is: [2, 4, 3, 5, 90, 7, 6]
true
```

?- insert_nth.

Examples History Solutions table results Run!

practical 17

SWISH File Edit Examples Help 289 users online Search

Program Program Program Program Program Program +

```
1  /* Pract17 Write a Prolog program to implement delete_nth (N, L, R) that
2   * removes the element on Nth position from a list L to generate a list R.
3  */
4
5  delete_nth(1, [_|T], T).
6  delete_nth(N, [H|T], [H|R]):-
7      N1 is N-1,
8      delete_nth(N1, T, R).
9  delete_nth:-
10     write("Enter the list: "),
11     read(L),
12     write("Enter the position of the element to be deleted: "),
13     read(N),
14     delete_nth(N,L,R),
15     write("Final list after deletion of element at "),
16     write(N),write(" position in the list is: "),
17     write(R),!.
```

delete_nth.

Enter the list:

([3,5,4,8,6,9]).

Enter the position of the element to be deleted:

3

Final list after deletion of element at 3 position in the list is: [3, 5, 8, 6, 9]

true 1

delete_nth.

Enter the list:

([d,e,f,g,h,i,k]).

Enter the position of the element to be deleted:

5

Final list after deletion of element at 5 position in the list is: [d, e, f, g, i, k]

true 1

?- delete_nth.

Examples History Solutions table results Run!

practical 18

SWISH File Edit Examples Help 285 users online Search

Program Program Program Program Program Program

```
/* Pract18. Write a program in PROLOG to implement merge (L1, L2, L3) where
 * L1 is first ordered list and L2 is second ordered list and L3 represents
 * the merged list.
*/

merge([],L2,L2).
merge(L1,[],L1).
merge([H1|T1],[H2|T2],[H1|T3]):-
    H1=<H2,
    merge(T1, [H2|T2], T3).
merge([H1|T1],[H2|T2],[H2|T3]):-
    merge([H1|T1], T2, T3).
merge:-
    write("Enter the first ordered list: "),
    read(L1),
    write("Enter the second ordered list: "),
    read(L2),
    merge(L1,L2,L3),
    write("The final list after merging given two ordered list: "),
    write(L3),!.
```

merge.

Enter the first ordered list:
([1,2,3,4]).
Enter the second ordered list:
([5,6,7,8]).
The final list after merging given two ordered list: [1, 2, 3, 4, 5, 6, 7, 8]
true 1

merge.

Enter the first ordered list:
([2,4,6,8]).
Enter the second ordered list:
([8,10,12,14]).
The final list after merging given two ordered list: [2, 4, 6, 8, 8, 10, 12, 14]
true 1

?- merge.

Examples History Solutions table results Run!