برنامج لإيجاد ناتج تقاطع قائمتين من الأعداد الصحيحة

```
Domains
i=integer
l=i*
Predicates
member(i,l)
inter(l,l,l)
Clauses
member(X,[X|_]):-!.
member(X,[_|T]):-member(X,T).
inter([],_,[]):-!.
inter([H|T],L,[H|T1]):-member(H,L),inter(T,L,T1),!.
inter([_|T],L,L1):-inter(T,L,L1).
Goal
1- inter([2,1,7],[5,3,2,6],L).
2- inter([5,3,6],[2,1,7],L).
3- inter([],[5,3,2,6],L).
4- inter([2,1,7],[],L).

Solutions
1- L=[2]
1 Solution
2- L=[]
1 Solution
3- L=[]
1 Solution
4- L=[]
1 Solution
```

برنامج لإيجاد ناتج الفرق بين قائمتين من الأعداد الصحيحة

```
Domains
i=integer
l=i*
Predicates
member(i,l)
diff(l,l,l)
Clauses
member(X,[X|_]):-!.
member(X,[_|T]):-member(X,T).
diff([],_,[]):-!.
diff([H|T],L,[H|T1]):-not(member(H,L)),diff(T,L,T1),!.
diff([_|T],L,L1):-diff(T,L,L1).
Goal
1- diff([2,1,7],[5,3,2,6],L).
2- diff([5,3,2,6],[2,1,7],L).
3- diff([],[5,3,2,6],L).
4- diff([2,1,7],[],L).

Solutions
1- L=[1,7]
1 Solution
2- L=[5,3,6]
1 Solution
3- L=[]
1 Solution
4- L=[2,1,7]
1 Solution
```

برنامج لإضافة عنصر في بداية قائمة من الأعداد الصحيحة

```
Domains
i=integer
l=i*
Predicates
insert1st(i,l,l)
Clauses
insert1st(X,L,[X|L]).
Goal
1- insert1st(2,[4,6,8],L).
2- insert1st(2,[],L).

Solutions
1- L=[2,4,6,8]
1 Solution
2- L=[2]
1 Solution
```

برنامج لإضافة عنصر في نهاية قائمة من الأعداد الصحيحة

```
Domains
i=integer
l=i*
Predicates
insertlast(i,l,l)
Clauses
insertlast(X,[],[X]):-!.
insertlast(X,[H|T],[H|T1]):-insertlast(X,T,T1).
Goal
1- insertlast(10,[4,6,8],L).
2- insertlast(10,[],L).

Solutions
1- L=[4,6,8,10]
1 Solution
2- L=[10]
1 Solution
```

برنامج لإضافة عنصر في موقع معين في قائمة من الأعداد الصحيحة

```
Domains
i=integer
l=i*
Predicates
insertloc(i,l,i,l)
Clauses
insertloc(X,L,1,[X|L]):-!.
insertloc(X,[H|T],N,[H|T1]):-N1=N-1,insertloc(X,T,N1,T1).
Goal
1- insertloc(6,[2,4,8],1,L).
2- insertloc(6,[2,4,8],2,L).
3- insertloc(6,[2,4,8],3,L).
4- insertloc(6,[2,4,8],4,L).
5- insertloc(6,[2,4,8],5,L).

Solutions
1- L=[6,2,4,8]
1 Solution
2- L=[2,6,4,8]
1 Solution
3- L=[2,4,6,8]
1 Solution
4- L=[2,4,8,6]
1 Solution
5- No Solution
```

برنامج لحذف العنصر الأول من قائمة من الأعداد الصحيحة

Domains

i=integer

l=i*

Predicates

delete1st(l,l)

Clauses

delete1st([_|T],T).

Goal

1- delete1st ([2,4,6,8],L).

2- delete1st ([],L).

Solutions

1- L=[4,6,8]

1 Solution

2- No Solution

برنامج لحذف العنصر الأخير من قائمة من الأعداد الصحيحة

```
Domains
i=integer
l=i*
Predicates
deletelast(l,l)
Clauses
deletelast([_],[]):-!.
deletelast([H|T],[H|T1]):-deletelast(T,T1).
Goal
1- deletelast ( [2,4,6,8],L).
2- deletelast ([],L).

Solutions
1- L=[2,4,6]
1 Solution
2- No Solution
```

برنامج لحذف عنصر من موقع معين في قائمة من الأعداد الصحيحة

```
Domains
i=integer
l=i*
Predicates
deleteloc(l,i,l)
Clauses
deleteloc([_|T],1,T):-!.
deleteloc([H|T],N,[H|T1]):-N1=N-1, deleteloc(T,N1,T1).
Goal
1- deleteloc([2,4,6,8],1,L).
2- deleteloc([2,4,6,8],2,L).
3- deleteloc([2,4,6,8],3,L).
4- deleteloc([2,4,6,8],4,L).
5- deleteloc([2,4,6,8],5,L).

Solutions
1- L=[4,6,8]
1 Solution
2- L=[2,6,8]
1 Solution
3- L=[2,4,8]
1 Solution
4- L=[2,4,6]
1 Solution
5- No Solution
```

برنامج لحذف عنصر معين من قائمة من الأعداد الصحيحة

```
Domains
i=integer
l=i*
Predicates
deleteitem(i,l,l)
Clauses
deleteitem(X,[X|T],T):-!.
deleteitem(X,[H|T],[H|T1]):-deleteitem(X,T,T1).
Goal
1- deleteitem(2,[2,4,6],L).
2- deleteitem(6,[2,4,6],L).
3- deleteitem(4,[2,4,6],L).
4- deleteitem(8,[2,4,6],L).

Solutions
1- L=[4,6]
1 Solution
2- L=[2,4]
1 Solution
3- L=[2,6]
1 Solution
4- No Solution
```

برنامج لحذف تكرار عنصر معين من قائمة من الأعداد الصحيحة

```
Domains
i=integer
l=i*
Predicates
deleteitem(i,l,l)
deleterep(i,l,l)
Clauses
deleteitem(X,[X|T],T):-!.
deleteitem(X,[H|T],[H|T1]):-deleteitem(X,T,T1).
deleterep(X,L,L1):-deleteitem(X,L,L2),deleterep(X,L2,L1).
Goal
1- deleterep(2,[2,2,4,6],L).
2- deleterep(4,[2,4,4,6],L).
3- deleterep(6,[2,4,6,6],L).
4- deleterep(8,[2,4,6],L).

Solutions
1- L=[4,6]
1 Solution
2- L=[2,4]
1 Solution
3- L=[2,6]
1 Solution
4- No Solution
```

برنامج لجعل العنصر الأول هو العنصر الأخير في قائمة من الأعداد الصحيحة

```
Domains
i=integer
l=i*
Predicates
insertlast(i,l,l)
swap(l,l)
Clauses
insertlast(X,[],[X]):-!.
insertlast(X,[H|T],[H|T1]):-insertlast(X,T,T1).
swap([],[]):-!.
swap([H|T],L):-insertlast(H,T,L).
Goal
swap([8,2,4,6],L).

Solutions
L=[2,4,6,8]
1 Solution
```

برنامج لإستبدال عنصر معين بعنصر آخر في قائمة من الأعداد الصحيحة

```
Domains
i=integer
l=i*
Predicates
replaceitem(i,i,l,l)
Clauses
replaceitem(Y,X,[X|T],[Y|T]):-!.
replaceitem(Y,X,[H|T],[H|T1]):-replaceitem(Y,X,T,T1).
Goal
1- replaceitem(3,2,[8,2,4,6],L).
2- replaceitem(3,2,[8,10,4,6],L).
3- replaceitem(3,2,[],L).

Solutions
1- L=[8,3,4,6]
1 Solution
2- No Solution
3- No Solution
```

برنامج لاختبار وجود عنصر في نهاية قائمة من الأعداد الصحيحة

```
Domains
i=integer
li=i*
Predicates
last(i,li)
Clauses
last(X,[X]):-!.
last(X,[_|T]):-last(X,T).
Goal
1- last(3,[1,2,3]).
2- last(3,[3]).
3- last(3,[1,3,2]).
4- last(3,[3,1,2]).
5- last(3,[]).
```

Solutions

1- yes

2- yes

3- no

4- no

5- no

```
domains
c=char
lc=c*
s=string
predicates
convert(lc,s(
clauses
convert.!-:(""‹[])
convert([H|T],S):-
str_char(S1,H),convert(T,NS),concat(S1,NS,S.(
goal
convert(['H','a','m','i','d'],S).
```