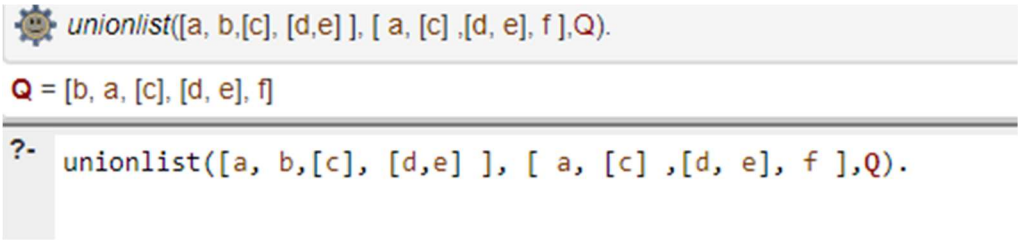
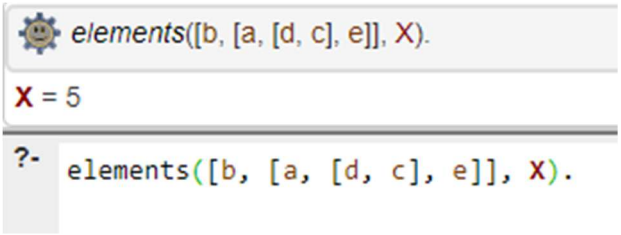
Karan Sahu, kxs190007, cs4337.503

1.



2.



Karan Sahu, kxs190007, cs4337.503

e = exit
c = call
el = elements

Karan Sahu | kxs190007

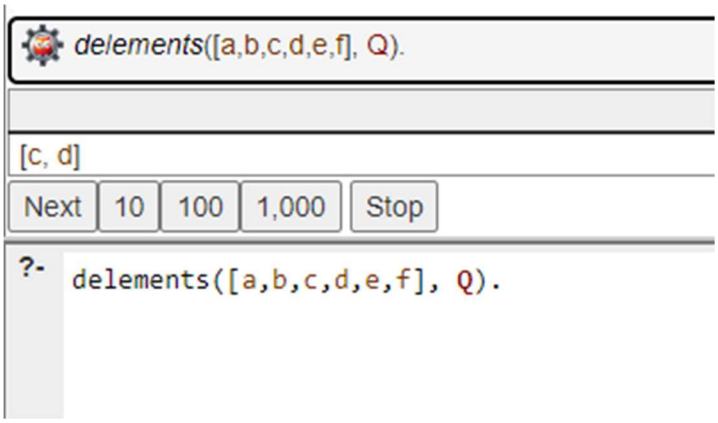
(1)	1	c	el([6,[a,[d,c],e]], -5030)
(2)	2	c	el(6, -5348)
(2)	2	e	el(6, 1)
(3)	2	c	el([[a, [d,c], e]], -5350)
(4)	3	c	el([a, [d,c], e]], -5356)
(5)	4	c	el(a, -5354)
(5)	4	e	el(a, 1)
(6)	4	c	el([d,c], e], -5358)
(7)	5	c	el([d,c], e], -5356)
(8)	6	c	el([d,c], -5360)
(8)	6	e	el(d, 1)
(9)	6	c	el([c], -5362)
(10)	7	c	el(c, -5364)
(10)	7	e	el(c, 1)
(11)	7	c	el([], -5366)
(11)	7	e	el([], 0)
(12)	7	c	-5362 is 1+0
(12)	7	e	1 is 1+0
(9)	6	e	el([c], 1)
(13)	6	c	-5358 is 1+1
(13)	6	e	2 is 1+1
(7)	5	e	el([d,c], 2)
(14)	5	c	el([e], -5380)
(15)	6	c	el(e, -5382)
(15)	6	e	el(e, 1)
(16)	6	c	el([], -5384)
(16)	6	e	el([], 0)
(17)	6	c	-5380 is 1+0
(17)	6	e	1 is 1+0
(14)	5	e	el([e], 1)
(18)	5	c	-5356 is 2+1
(18)	5	e	3 is 2+1
(6)	4	e	el([[d,c],e], 3)
(19)	4	c	-5352 is 3+1
(19)	4	e	4 is 3+1
(4)	3	e	el([a[d,c],e], 4)
(20)	3	c	el([], -5404)
(20)	3	e	el([], 0)
(21)	3	c	-5350 is 4+0
(21)	3	e	4 is 4+0
(3)	2	e	el([[a, [d, c], e]], 4)
(22)	2	c	-5030 is 1+4
(22)	2	e	5 is 1+4
(1)	1	e	el([6, [a, [d, c], e]], 5)

X = 5

(1)	1	call	elements([b, [a, [d, c], e]], -5030)
(2)	2	call	elements(b, -5348)
(2)	2	exit	elements(b, 1)
(3)	2	call	elements([[a, [d, c], e]], -5350)
(4)	3	call	Elements([a, [d, c], e]], -5356)
(5)	4	call	elements(a, -5354)
(5)	4	exit	elements(a, 1)
(6)	4	call	elements([d, c], e], -5356)
(7)	5	call	elements([d, c], -5358)
(8)	6	call	elements(d, -5360)
(8)	6	exit	elements(d, 1)
(9)	6	call	elements([c], -5362)
(10)	7	call	elements(c, -5364)
(10)	7	exit	elements(c, 1)
(11)	7	call	elements([], -5366)
(11)	7	exit	elements([], 0)
(12)	7	call	-5362 is 1+0
(12)	7	exit	1 is 1+0
(9)	6	exit	elements([c] , 1)
(13)	6	call	-5358 is 1+1
(13)	6	exit	2 is 1+1
(7)	5	exit	elements([d,c], 2)
(14)	5	call	elements([e] , -5380)
(15)	6	call	elements(e, -5382)
(15)	6	exit	elements(e, 1)
(16)	6	call	elements([], -5384)
(16)	6	exit	elements([], 0)
(17)	6	call	-5380 is 1+0
(17)	6	exit	1 is 1+0
(14)	5	exit	elements([e], 1)
(18)	5	call	-5356 is 2+1
(18)	5	exit	3 is 2+1
(6)	4	exit	elements([[d,c],e] , 3)
(19)	4	call	-5352 is 1+3
(19)	4	exit	4 is 3+1
(4)	3	exit	elements([a[d,c],e] , 4)
(20)	3	call	elements([] ,-5404)
(20)	3	exit	elements([] ,0)
(21)	3	call	-5350 is 4+0
(21)	3	exit	4 is 4+0
(3)	2	exit	elements([[a, [d, c], e]], 4)
(22)	2	call	-5030 is 1+4
(22)	2	exit	5 is 1+4
(1)	1	exit	elements([b, [a, [d, c], e]], 5)

X=5

3.


delements([a,b,c,d,e,f], Q).
[c, d]
Next
10
100
1,000
Stop
?-
delements([a,b,c,d,e,f], Q).

4.

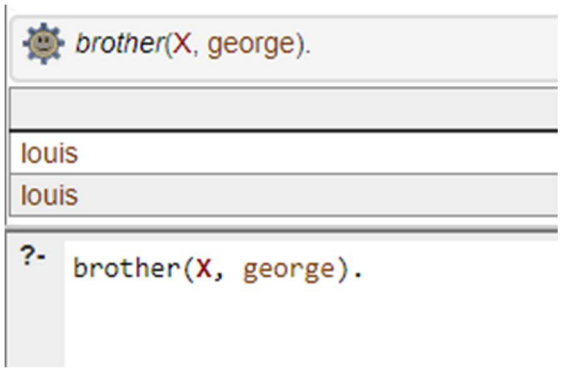
a.


mother(X,george).
X = catherine
?-
mother(X,george).

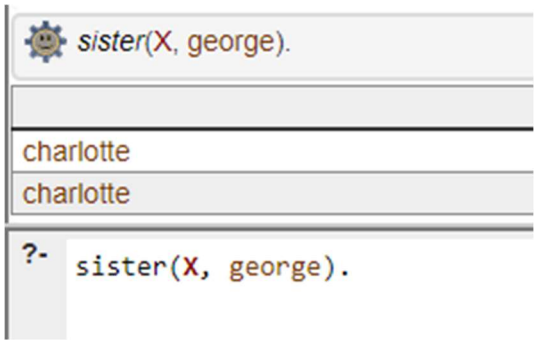
b.


father(X, george).
X = william
?-
father(X, george).

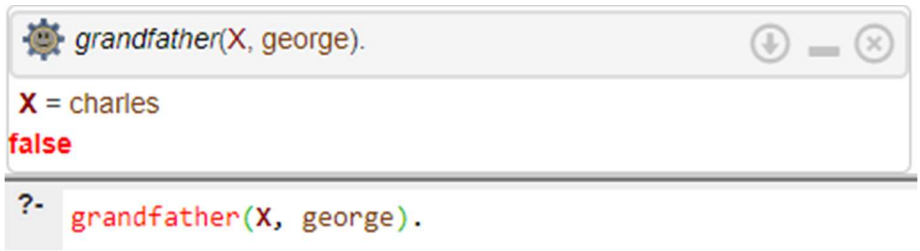
c.


brother(X, george).
louis
louis
?- brother(X, george).

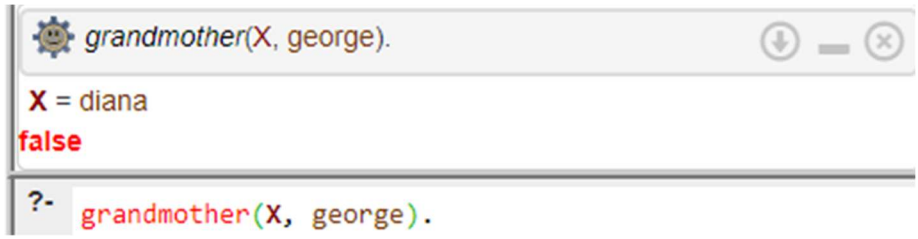
d.


sister(X, george).
charlotte
charlotte
?- sister(X, george).

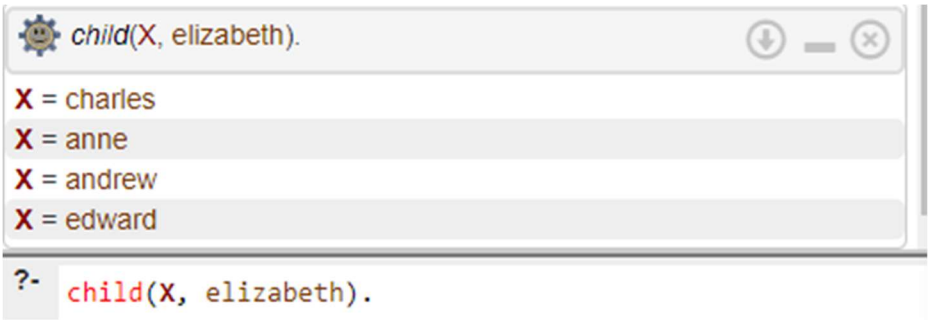
e.


grandfather(X, george).
X = charles
false
?- grandfather(X, george).

f.


grandmother(X, george).
X = diana
false
?- grandmother(X, george).

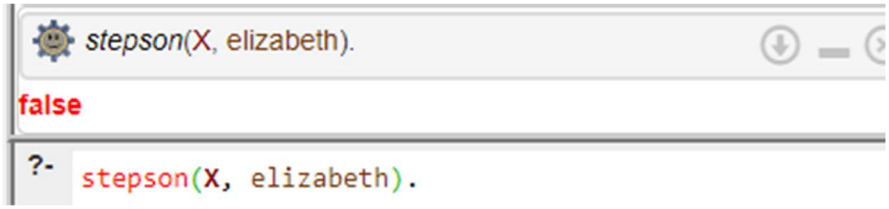
g.


child(X, elizabeth).
X = charles
X = anne
X = andrew
X = edward
?- child(X, elizabeth).

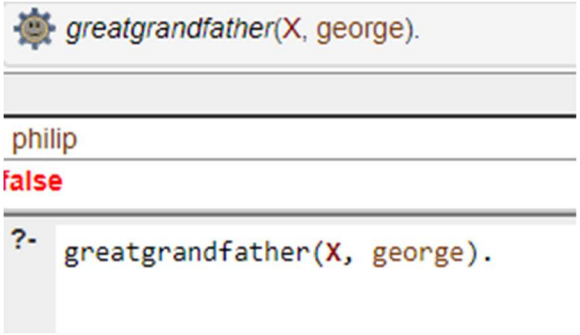
h.


stepdaughter(X, elizabeth).
false
?- stepdaughter(X, elizabeth).

i.


stepson(X, elizabeth).
false
?- stepson(X, elizabeth).

j.


greatgrandfather(X, george).
philip
false
?- greatgrandfather(X, george).

k.

Karan Sahu, kxs190007, cs4337.503

aggregate_all(count, female(X), Count).

X	Count	
X	17	1

```
?- aggregate_all(count, female(X), Count).
```

l.

equal_to_family(X, 1).

harryfamily

false

```
?- equal_to_family(X, 1).
```

m.

equal_to_family(X, 2).

andrewfamily
edwardfamily
peterfamily
zarafamily
charlesfamily
annefamily

```
?- equal_to_family(X, 2).
```

n.

bigger_than_family(X, 1).

andrewfamily
edwardfamily
williamfamily
peterfamily
zarafamily
elizabethfamily
charlesfamily
annefamily

```
?- bigger_than_family(X, 1).
```

5.

map(Waller, Harris, Walker, Montgomery, Liberty, FortBend, Colorado, Austin, Chambers, Galveston, Brazoria, Matagora, Wharton)

Waller	Harris	Walker	Montgomery	Liberty	FortBend	Colorado	Austin	Chambers	Galveston	Brazoria	Matagora	Wharton	
blue	red	blue	yellow	blue	white	yellow	red	yellow	blue	yellow	red	blue	1

```
?- map( Waller, Harris, Walker, Montgomery, Liberty, FortBend, Colorado, Austin, Chambers, Galveston, Brazoria, Matagora,
Wharton)
```

6.

a.

caseslist(L), *sum*(L,X).

L	X	
[20, 39, 310, 307, 434, 200]	1310	1

```
?- caseslist(L), sum(L,X).
```

b.

lesscasesthan(X,100).

groupA
groupB

```
?- lesscasesthan(X,100).
```

c.

deathlist(L), *maxlist*(L,X), *deaths*(Groups, X)

L	X	Groups	
[0, 2, 4, 6, 23, 24]	24	groupF	1

```
?- deathlist(L), maxlist(L,X), deaths(Groups, X)
```

d.

morecasesthan(X,100), *lessdeathsthan*(X, 5).

groupC

false

```
?- morecasesthan(X,100), lessdeathsthan(X, 5).
```

e.

deathlist(L), *maxlist*(L, X), *deaths*(LargestDeath, X), *caseslist*(E), *maxlist*(E, Y), *cases*(LargestCases, Y).

L	X	LargestDeath	E	Y	LargestCases	
[0, 2, 4, 6, 23, 24]	24	groupF	[20, 39, 310, 307, 434, 200]	434	groupE	1

```
?- deathlist(L), maxlist(L, X), deaths(LargestDeath, X), caseslist(E), maxlist(E, Y), cases(LargestCases, Y).
```

f.

deathlist(L), *minlist*(L, X), *deaths*(LowestDeath, X), *caseslist*(E), *minlist*(E, Y), *cases*(LowestCases, Y), *hospitalizedlist*(N), *minlist*(N, Z), *hospitalized*(Lowesthospitalized, Z).

L	X	LowestDeath	E	Y	LowestCases	N	Z	Lowesthospitalized	
[0, 2, 4, 6, 23, 24]	0	groupA	[20, 39, 310, 307, 434, 200]	20	groupA	[5, 6, 22, 55, 79, 80]	5	groupA	1

```
?- deathlist(L), minlist(L, X), deaths(LowestDeath, X), caseslist(E), minlist(E, Y), cases(LowestCases, Y), hospitalizedlist(N), minlist(N, Z),
hospitalized(Lowesthospitalized, Z).
```

g.

cases(groupA,X), *cases*(groupB,Y),*cases*(groupC,Z),*sum*([X,Y,X], Sum).

X	Y	Z	Sum	
20	39	310	79	1

```
?- cases(groupA,X), cases(groupB,Y),cases(groupC,Z),sum([X,Y,X], Sum).
```

h.

Karan Sahu, kxs190007, cs4337.503

deaths(groupD, X), *moredeathsthan*(ANSWER, X)

X	ANSWER	
6	groupE	1
6	groupF	2

```
?- deaths(groupD, X), moredeathsthan(ANSWER, X)
```

i.

X	Y	LargerDeath	LargerCases	
6	307	groupE	groupE	1

```
?- deaths(groupD, X), cases(groupD, Y), moredeathsthan(LargerDeath, X), morecasesthan(LargerCases, Y),
LargerDeath = LargerCases.
```

j.

X	Y	LargerDeath	LargerCases	
6	307	groupE	groupC	1
6	307	groupE	groupE	2
6	307	groupF	groupC	3
6	307	groupF	groupE	4

```
?- deaths(groupD, X), cases(groupD, Y), moredeathsthan(LargerDeath, X), morecasesthan(LargerCases, Y).
```

k.

deathlist(L), *sum*(L,X), *aggregate_all*(count, *deaths*(_,Y), Count), Average is X/Count.

L	X	Y	Count	Average	
[0, 2, 4, 6, 23, 24]	59	Y	6	9.833333333333334	1

```
?- deathlist(L), sum(L,X), aggregate_all(count, deaths(_,Y), Count), Average is X/Count.
```

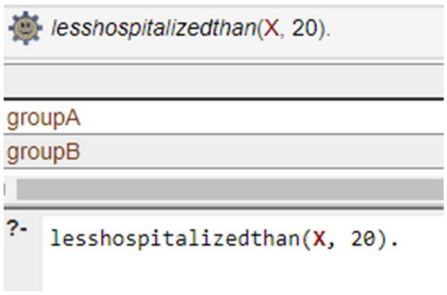
L.

X	Y	
groupC	groupC	1
groupD	groupD	2

```
?- morehospitalizedthan(X, 20), lesshospitalizedthan(Y, 60), X=Y.
```

m.

Karan Sahu, kxs190007, cs4337.503



n.