CSE341 – Programming Languages (Fall 2020) Homework #4

REPORT

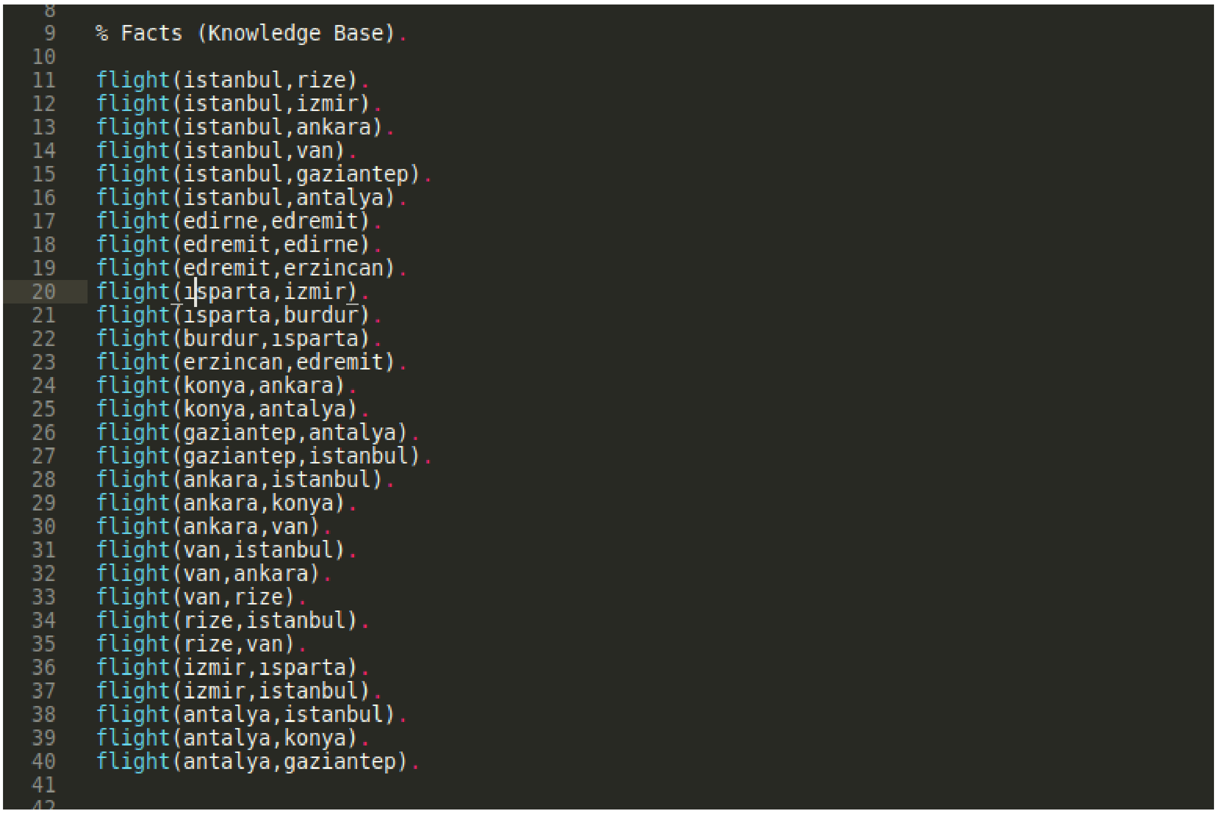
MİRAY YILDIZ

161044023

PART 1

I created possible flight graph. These are facts. For instance flight(istanbul, ankara) means that there is a flight from istanbul to ankara .

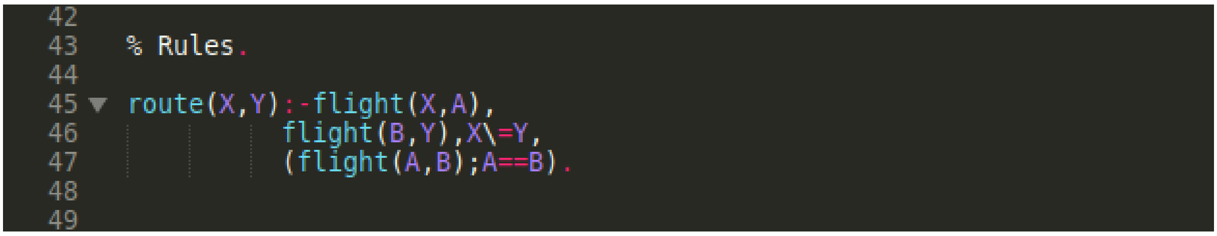
All facts :



```
% Facts (Knowledge Base).

flight(istanbul,rize).
flight(istanbul,izmir).
flight(istanbul,ankara).
flight(istanbul,van).
flight(istanbul,gaziantep).
flight(istanbul,antalya).
flight(edirne,edremit).
flight(edremit,edirne).
flight(edremit,erzincan).
flight(ısparta,izmir).
flight(ısparta,burdur).
flight(burdur,ısparta).
flight(erzincan,edremit).
flight(konya,ankara).
flight(konya,antalya).
flight(gaziantep,antalya).
flight(gaziantep,istanbul).
flight(ankara,istanbul).
flight(ankara,konya).
flight(ankara,van).
flight(van,istanbul).
flight(van,ankara).
flight(van,rize).
flight(rize,istanbul).
flight(rize,van).
flight(izmir,ısparta).
flight(izmir,istanbul).
flight(antalya,istanbul).
flight(antalya,konya).
flight(antalya,gaziantep).
```

I checked direct rout between 2 given cities.

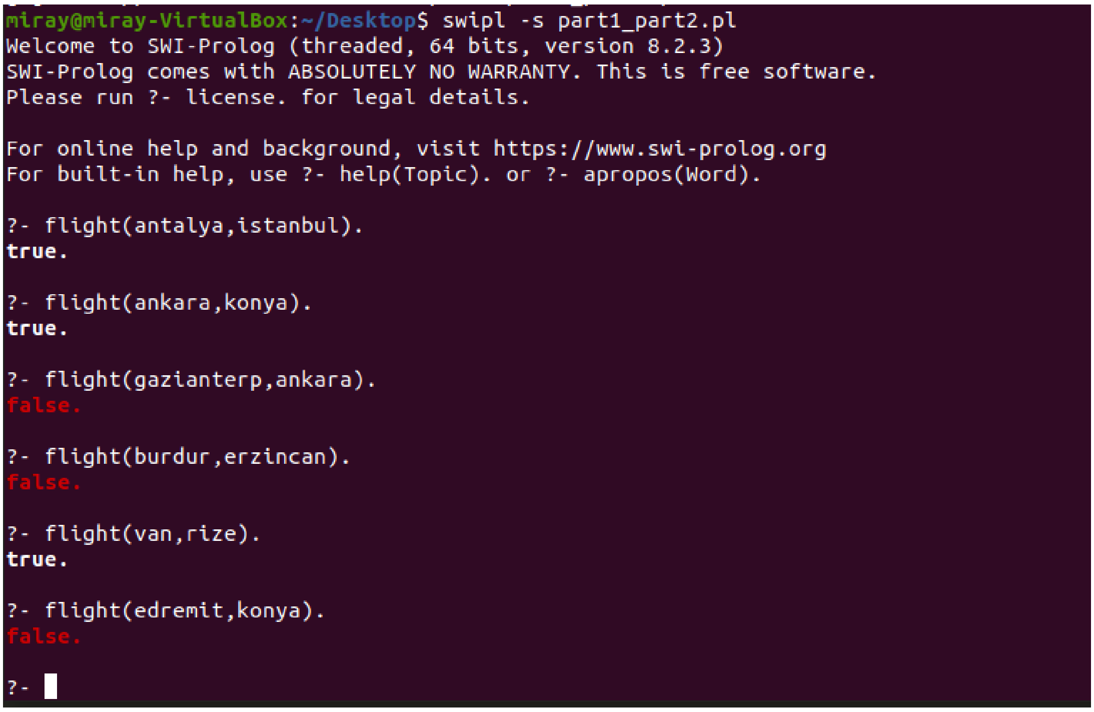
These rules :



```
% Rules.

route(X,Y):-flight(X,A),
            flight(B,Y),X\=Y,
            (flight(A,B);A==B).
```

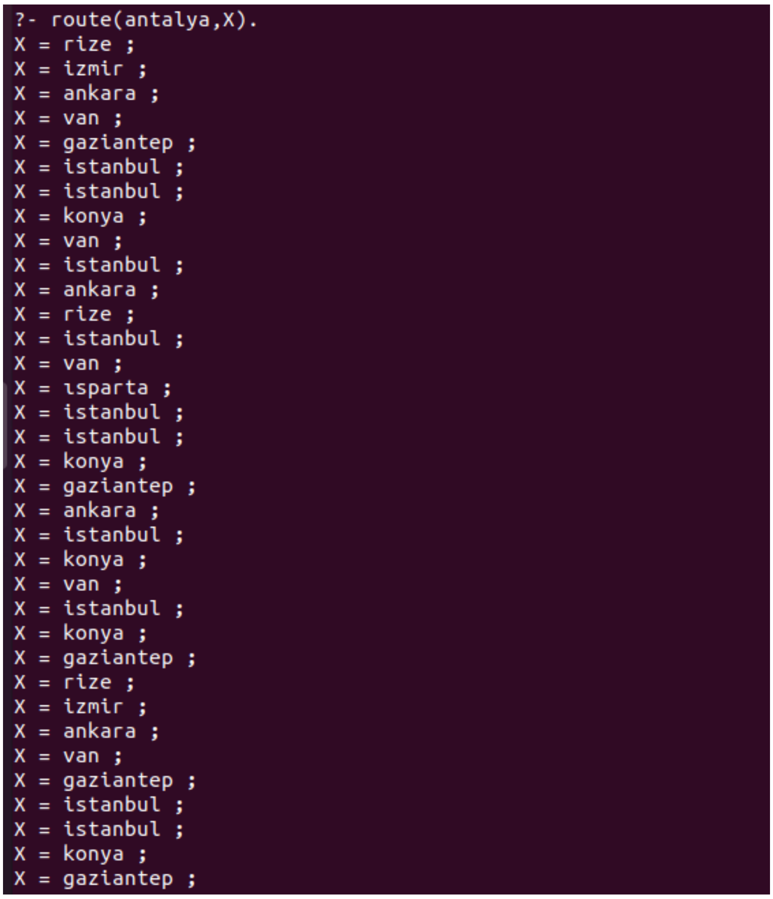
TESTS:

Check directed and undirected flights:

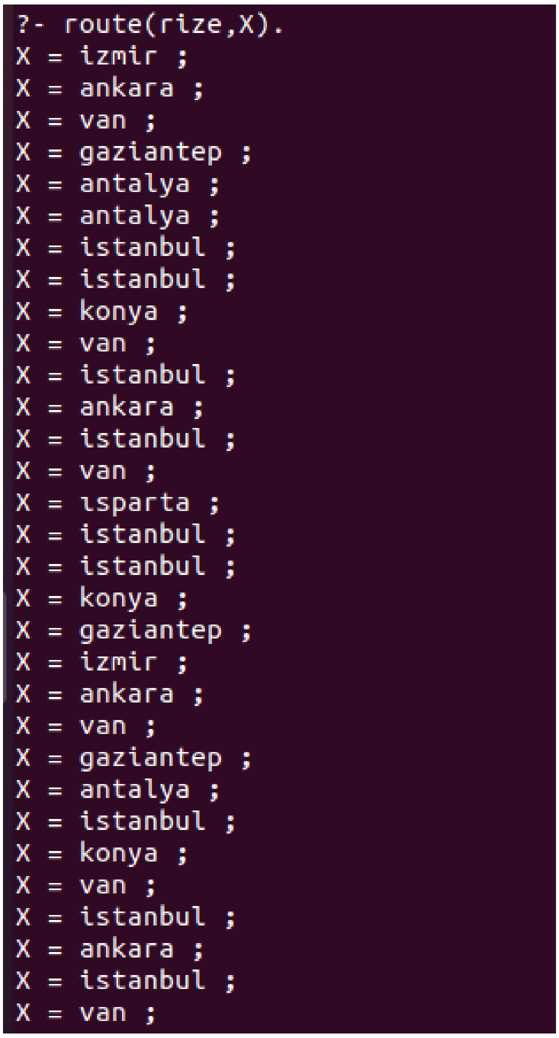


```
miray@miray-VirtualBox:~/Desktop$ swipl -s part1_part2.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- flight(antalya,istanbul).
true.

?- flight(ankara,konya).
true.

?- flight(gazianterp,ankara).
false.

?- flight(burdur,erzincan).
false.

?- flight(van,rize).
true.

?- flight(edremit,konya).
false.

?- 
```

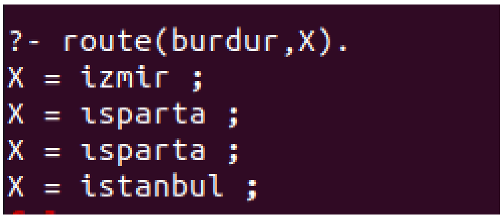
Check routes :



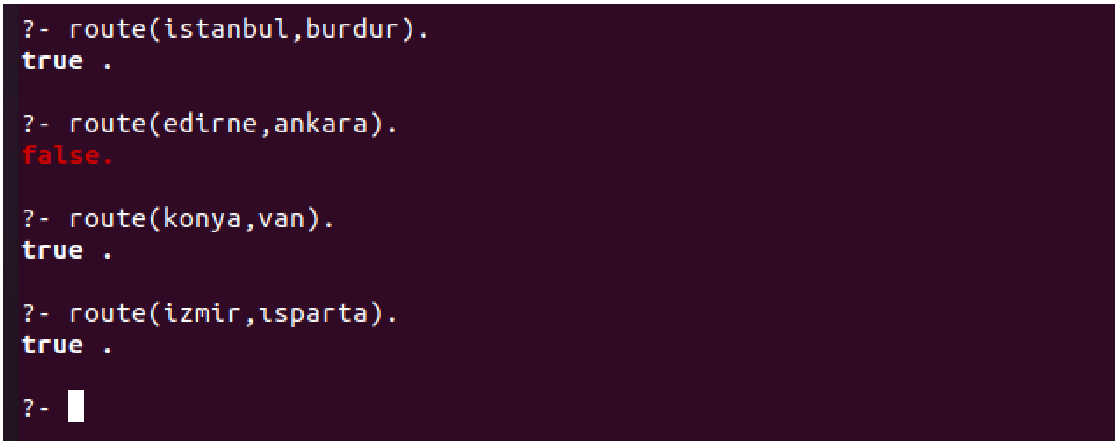
```
?- route(antalya,X).
X = rize ;
X = izmir ;
X = ankara ;
X = van ;
X = gaziantep ;
X = istanbul ;
X = istanbul ;
X = konya ;
X = van ;
X = istanbul ;
X = ankara ;
X = rize ;
X = istanbul ;
X = van ;
X = isparta ;
X = istanbul ;
X = istanbul ;
X = konya ;
X = gaziantep ;
X = ankara ;
X = istanbul ;
X = konya ;
X = van ;
X = istanbul ;
X = konya ;
X = gaziantep ;
X = rize ;
X = izmir ;
X = ankara ;
X = van ;
X = gaziantep ;
X = istanbul ;
X = istanbul ;
X = konya ;
X = gaziantep ;
```



```
?- route(rize,X).
X = izmir ;
X = ankara ;
X = van ;
X = gaziantep ;
X = antalya ;
X = antalya ;
X = istanbul ;
X = istanbul ;
X = konya ;
X = van ;
X = istanbul ;
X = ankara ;
X = istanbul ;
X = van ;
X = ısparta ;
X = istanbul ;
X = istanbul ;
X = konya ;
X = gaziantep ;
X = izmir ;
X = ankara ;
X = van ;
X = gaziantep ;
X = antalya ;
X = istanbul ;
X = konya ;
X = van ;
X = istanbul ;
X = ankara ;
X = istanbul ;
X = van ;
```



```
?- route(burdur,X).
X = izmir ;
X = ısparta ;
X = ısparta ;
X = istanbul ;
```

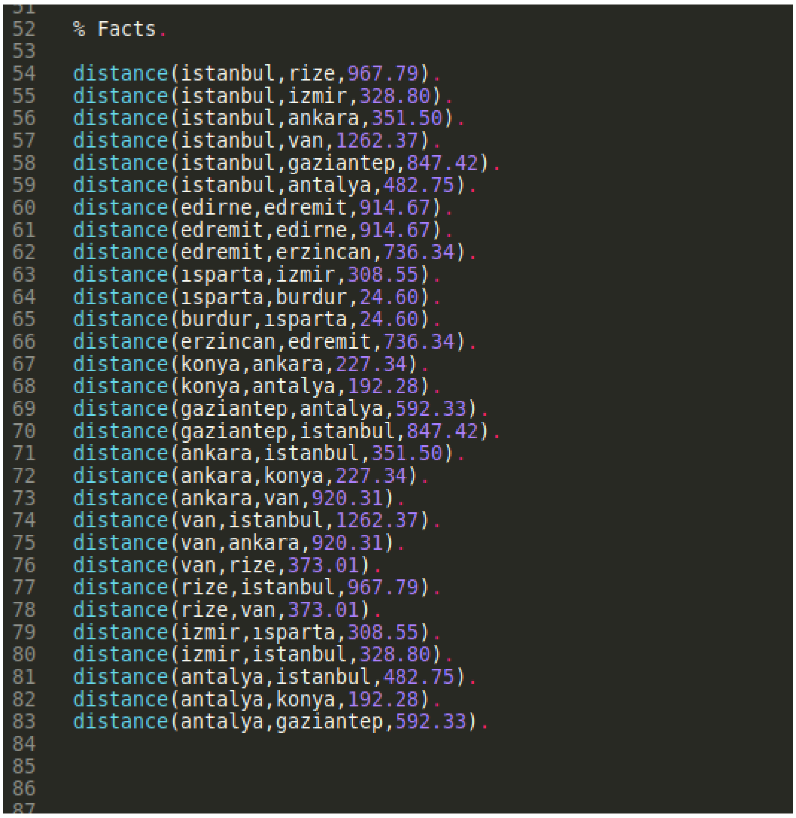


```
?- route(istanbul,burdur).
true .

?- route(edirne,ankara).
false.

?- route(konya,van).
true .

?- route(izmir,ısparta).
true .

?-
```

PART 2

In this part, I expanded flights with distances(Distances in given link)

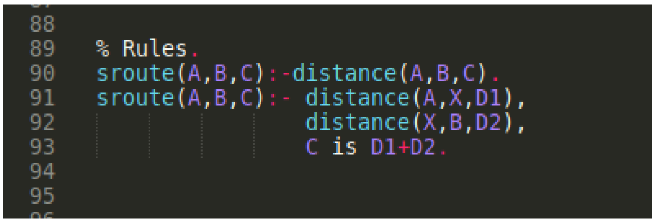
Facts:



```
% Facts.

distance(istanbul,rize,967.79).
distance(istanbul,izmir,328.80).
distance(istanbul,ankara,351.50).
distance(istanbul,van,1262.37).
distance(istanbul,gaziantep,847.42).
distance(istanbul,antalya,482.75).
distance(edirne,edremit,914.67).
distance(edremit,edirne,914.67).
distance(edremit,erzincan,736.34).
distance(ısparta,izmir,308.55).
distance(ısparta,burdur,24.60).
distance(burdur,ısparta,24.60).
distance(erzincan,edremit,736.34).
distance(konya,ankara,227.34).
distance(konya,antalya,192.28).
distance(gaziantep,antalya,592.33).
distance(gaziantep,istanbul,847.42).
distance(ankara,istanbul,351.50).
distance(ankara,konya,227.34).
distance(ankara,van,920.31).
distance(van,istanbul,1262.37).
distance(van,ankara,920.31).
distance(van,rize,373.01).
distance(rize,istanbul,967.79).
distance(rize,van,373.01).
distance(izmir,ısparta,308.55).
distance(izmir,istanbul,328.80).
distance(antalya,istanbul,482.75).
distance(antalya,konya,192.28).
distance(antalya,gaziantep,592.33).
```

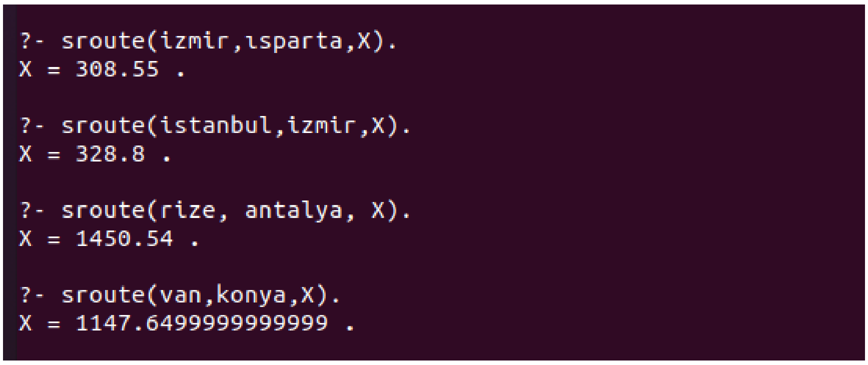
Then, I calculate directed or undirected flights.

Rules :



```
% Rules.
sroute(A,B,C):-distance(A,B,C).
sroute(A,B,C):- distance(A,X,D1),
                distance(X,B,D2),
                C is D1+D2.
```

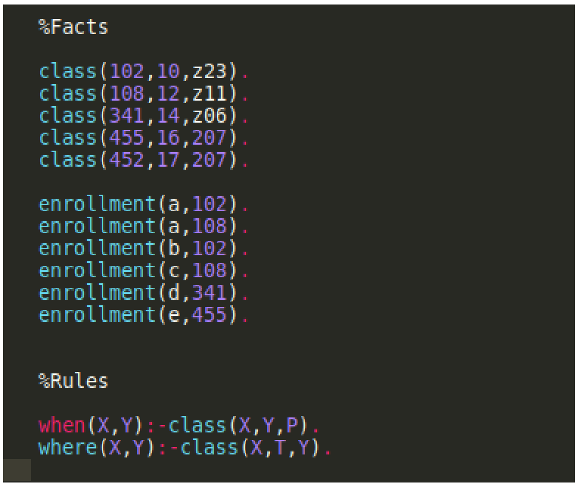
TEST:



```
?- sroute(izmir,ısparta,X).
X = 308.55 .

?- sroute(istanbul,izmir,X).
X = 328.8 .

?- sroute(rize, antalya, X).
X = 1450.54 .

?- sroute(van,konya,X).
X = 1147.6499999999999 .
```

PART 3

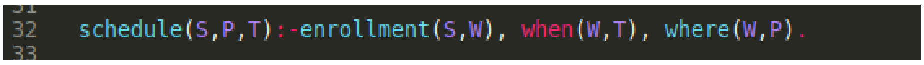
First, I write all “classes” and “enrollment” facts. Then I write “when” and “while” rules.



```
%Facts

class(102,10,z23).
class(108,12,z11).
class(341,14,z06).
class(455,16,207).
class(452,17,207).

enrollment(a,102).
enrollment(a,108).
enrollment(b,102).
enrollment(c,108).
enrollment(d,341).
enrollment(e,455).


%Rules

when(X,Y):-class(X,Y,P).
where(X,Y):-class(X,T,Y).
```

3.1

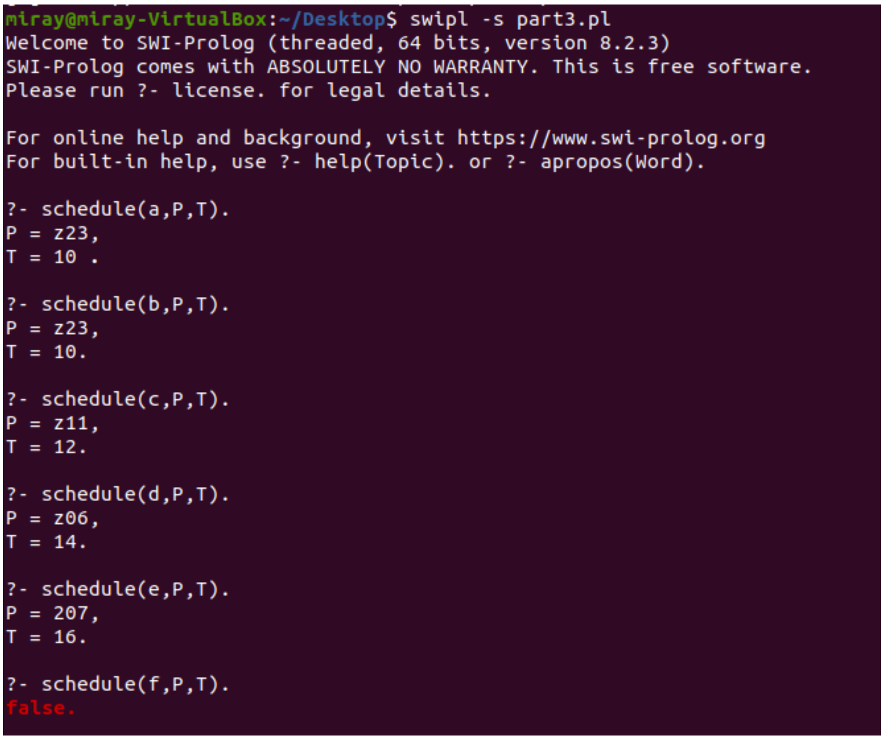
Predicate schedule(S,P,T) associates a student to a place and time of class. I write it using “enrollments”, “when” and “write” predicates.



```
schedule(S,P,T):-enrollment(S,W), when(W,T), where(W,P).
```

S -> Student number, P -> Classroom, T -> Time

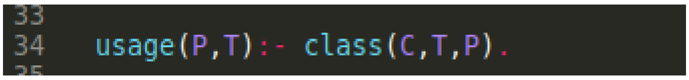
TEST :



```
miray@miray-VirtualBox:~/Desktop$ swipl -s part3.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- schedule(a,P,T).
P = z23,
T = 10 .

?- schedule(b,P,T).
P = z23,
T = 10.

?- schedule(c,P,T).
P = z11,
T = 12.

?- schedule(d,P,T).
P = z06,
T = 14.

?- schedule(e,P,T).
P = 207,
T = 16.

?- schedule(f,P,T).
false.
```

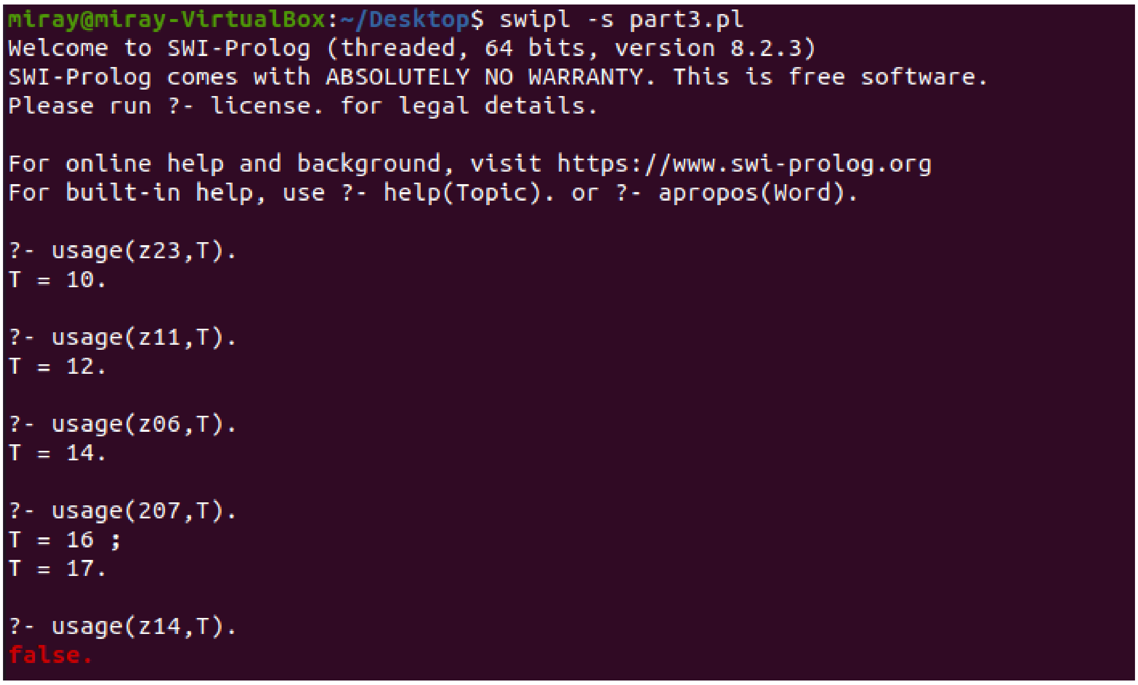
3.2

I write predicate “usage(P,T)” with “class” fact.



```
usage(P,T):- class(C,T,P).
```

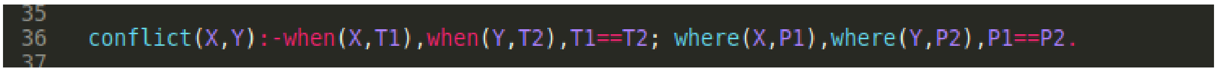
TEST :



```
miray@miray-VirtualBox:~/Desktop$ swipl -s part3.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- usage(z23,T).
T = 10.

?- usage(z11,T).
T = 12.

?- usage(z06,T).
T = 14.

?- usage(207,T).
T = 16 ;
T = 17.

?- usage(z14,T).
false.
```

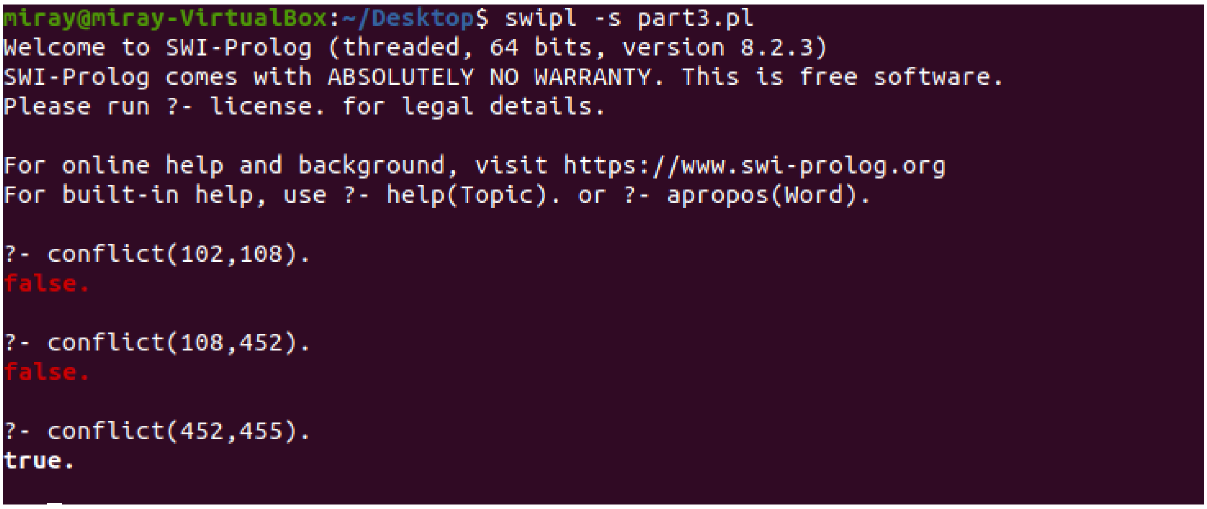
3.3

I used "when" and "where" rules while writing "conflict(X,Y)" predicate. This predicate returns true if place or time of courses overlap. Else, return false.



```
conflict(X,Y):-when(X,T1),when(Y,T2),T1==T2; where(X,P1),where(Y,P2),P1==P2.
```

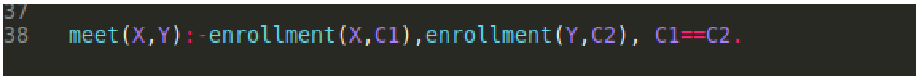
TEST:



```
miray@miray-VirtualBox:~/Desktop$ swipl -s part3.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- conflict(102,108).
false.

?- conflict(108,452).
false.

?- conflict(452,455).
true.
```

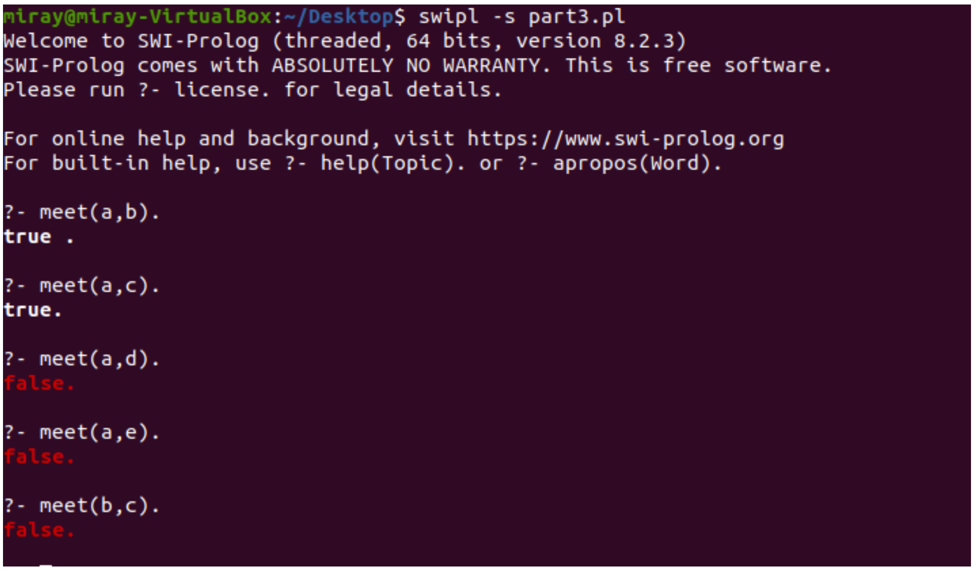
3.4

I write predicate “meet(X,Y)”, it returns true if student X and Y are present in the same classroom at the same time.



```
meet(X,Y):-enrollment(X,C1),enrollment(Y,C2), C1==C2.
```

TEST :

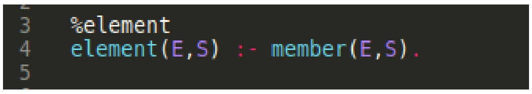


```
miray@miray-VirtualBox:~/Desktop$ swipl -s part3.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- meet(a,b).
true .

?- meet(a,c).
true.

?- meet(a,d).
false.

?- meet(a,e).
false.

?- meet(b,c).
false.
```

PART 4

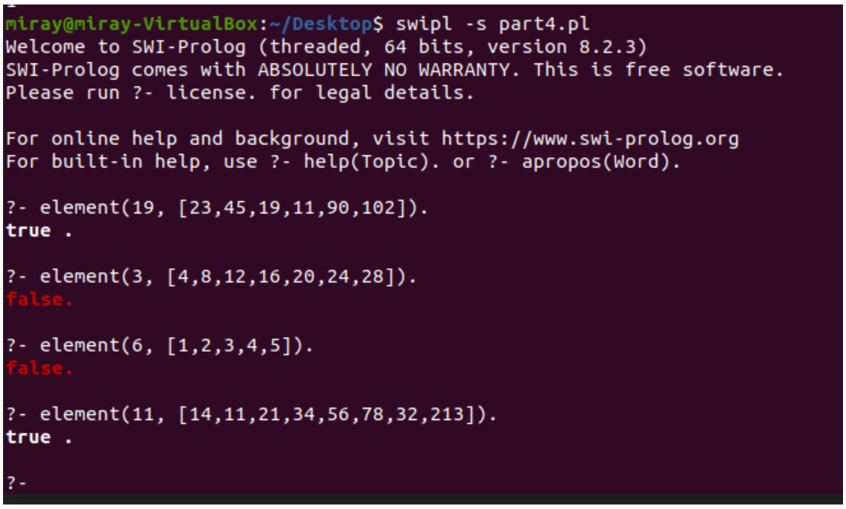
4.1

I define a Prolog predicate “element(E,S)” . It returns true if E is in S.



```
%element
element(E,S) :- member(E,S).
```

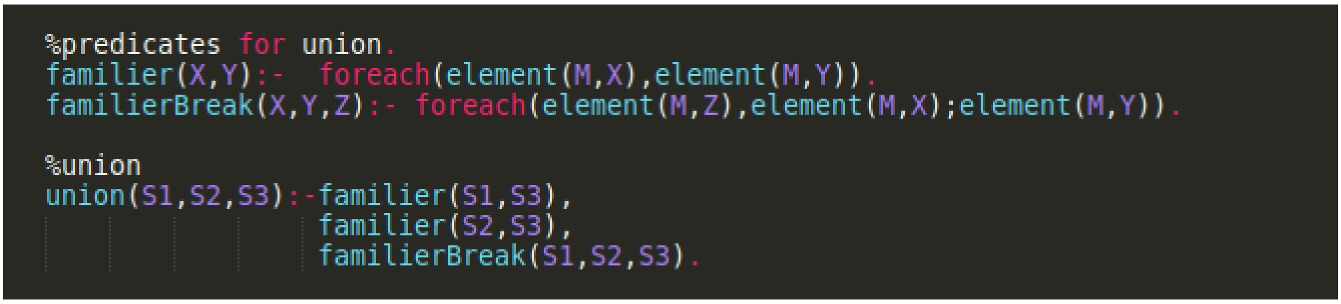
TEST :



```
miray@miray-VirtualBox:~/Desktop$ swipl -s part4.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- element(19, [23,45,19,11,90,102]).
true .

?- element(3, [4,8,12,16,20,24,28]).
false.

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

?- element(11, [14,11,21,34,56,78,32,213]).
true .

?-
```

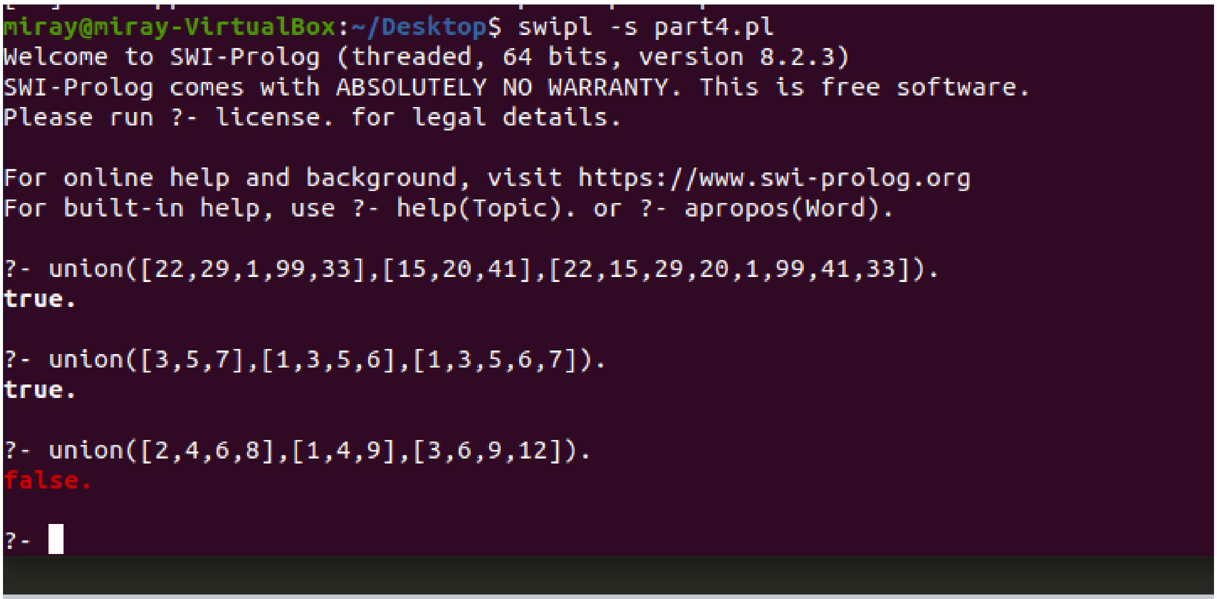
4.2

I define a predicate “union(S1,S2,S3)” . It returns true if S3 is the union of S1 and S2. I write some predicates for create union predicate. This predicates are familier and familierBreak.



```
%predicates for union.
familier(X,Y):-  foreach(element(M,X),element(M,Y)).
familierBreak(X,Y,Z):- foreach(element(M,Z),element(M,X);element(M,Y)).

%union
union(S1,S2,S3):-familier(S1,S3),
                 familier(S2,S3),
                 familierBreak(S1,S2,S3).
```

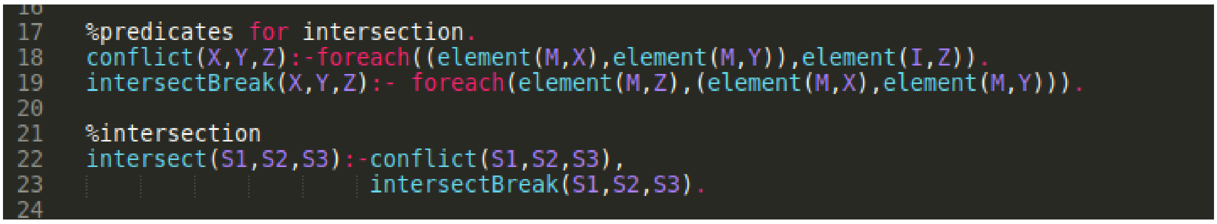
TEST :



```
miray@miray-VirtualBox:~/Desktop$ swipl -s part4.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- union([22,29,1,99,33],[15,20,41],[22,15,29,20,1,99,41,33]).
true.

?- union([3,5,7],[1,3,5,6],[1,3,5,6,7]).
true.

?- union([2,4,6,8],[1,4,9],[3,6,9,12]).
false.

?- 
```

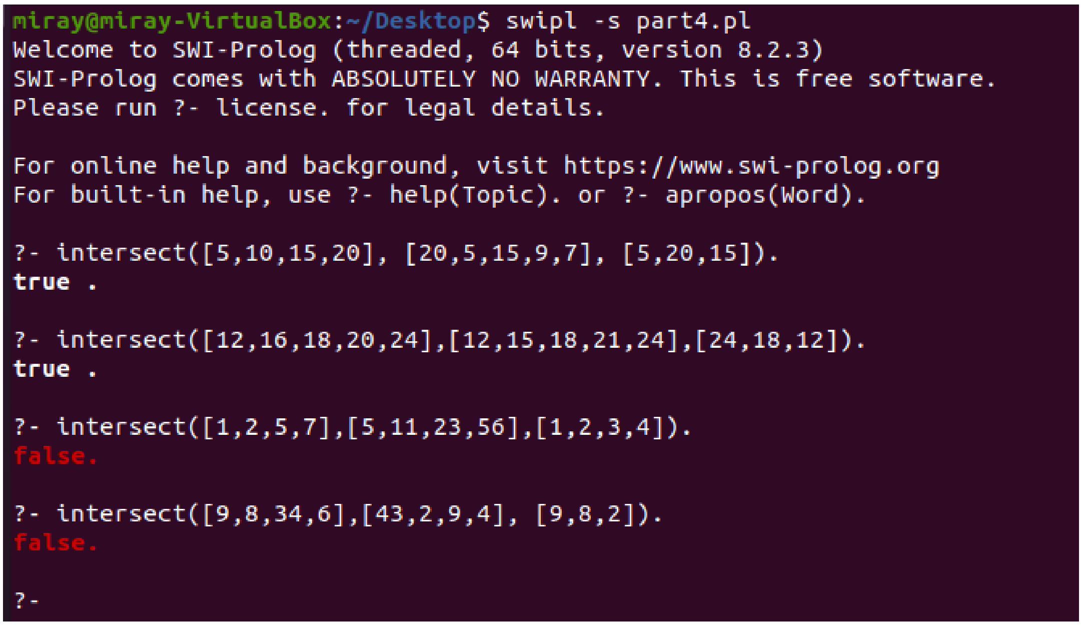
4.3

I define a predicate “intersect(S1,S2,S3)” . It returns true if S3 is the intersection of of S1 and S2. I write some predicates for create intersect predicate. This predicates are conflict and intersectBreak.



```
%predicates for intersection.
conflict(X,Y,Z):-foreach((element(M,X),element(M,Y)),element(I,Z)).
intersectBreak(X,Y,Z):- foreach(element(M,Z),(element(M,X),element(M,Y))).

%intersection
intersect(S1,S2,S3):-conflict(S1,S2,S3),
                     intersectBreak(S1,S2,S3).
```

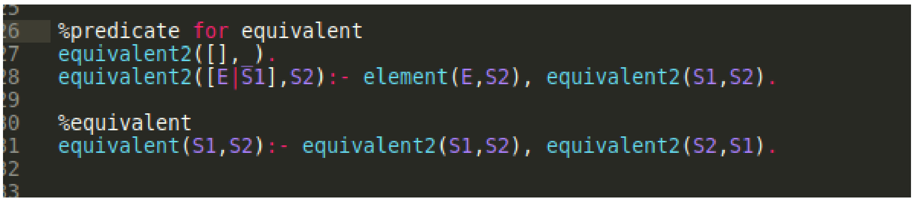
TEST :



```
miray@miray-VirtualBox:~/Desktop$ swipl -s part4.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- intersect([5,10,15,20], [20,5,15,9,7], [5,20,15]).
true .

?- intersect([12,16,18,20,24],[12,15,18,21,24],[24,18,12]).
true .

?- intersect([1,2,5,7],[5,11,23,56],[1,2,3,4]).
false.

?- intersect([9,8,34,6],[43,2,9,4], [9,8,2]).
false.

?-
```

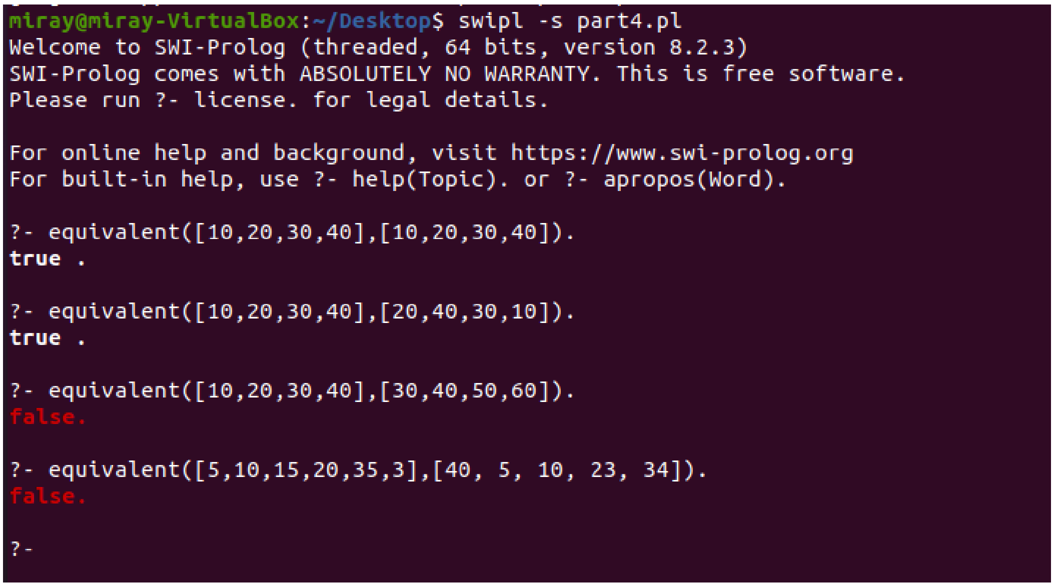
4.4

I define a predicate “equivalent(S1,S2)” . It returns true if S1 and S2 are equivalent sets. . I write a predicate for create equivalent predicate. This predicate is equivalent2.



```
%predicate for equivalent
equivalent2([],_).
equivalent2([E|S1],S2):- element(E,S2), equivalent2(S1,S2).

%equivalent
equivalent(S1,S2):- equivalent2(S1,S2), equivalent2(S2,S1).
```

TEST :



```
miray@miray-VirtualBox:~/Desktop$ swipl -s part4.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- equivalent([10,20,30,40],[10,20,30,40]).
true .

?- equivalent([10,20,30,40],[20,40,30,10]).
true .

?- equivalent([10,20,30,40],[30,40,50,60]).
false.

?- equivalent([5,10,15,20,35,3],[40, 5, 10, 23, 34]).
false.

?-
```

PART 5

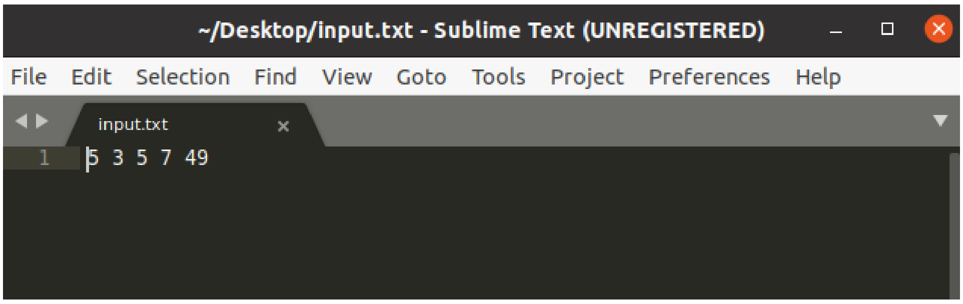
In this part, first I read numbers into input.txt file. Than, I create a list. Then I parsed the list. I determined the right and left terms. After that, I evaluate and compare the terms. I evalute terms in binary fact. In this fact, I did addition, subtraction, multiplication and division. After evaluations, I write operations into a output.txt. I wrote the query at the and. (?-lists(“input.txt”))

TEST :

İnput.txt contains :

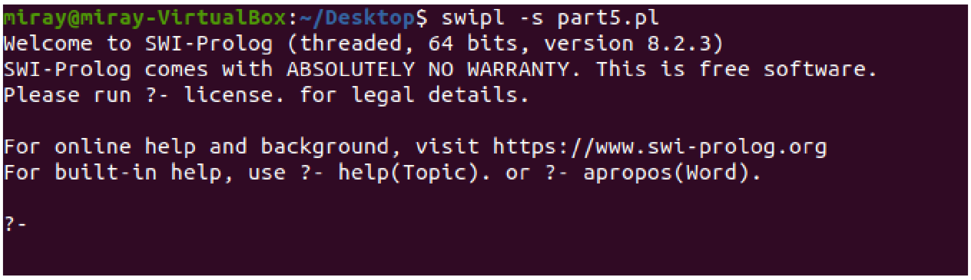
[5 3 5 7 49]

İnput.txt :


~/Desktop/input.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
input.txt

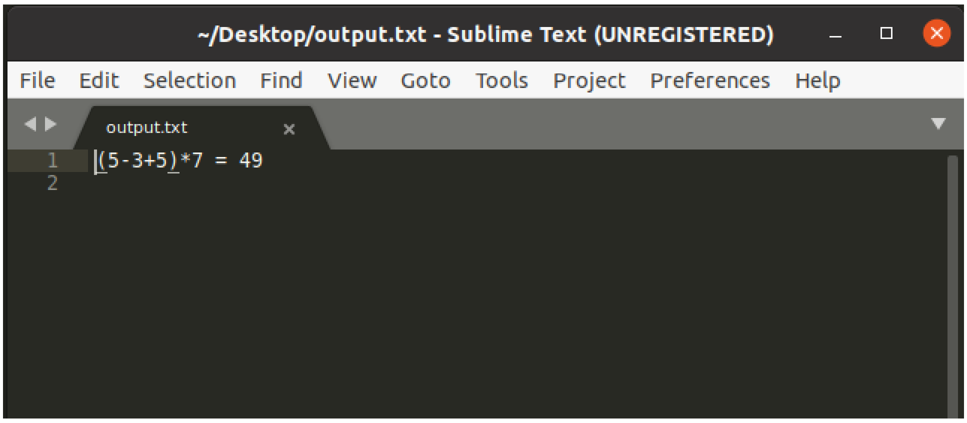
```
5 3 5 7 49
```

Compiler :



```
miray@miray-VirtualBox:~/Desktop$ swipl -s part5.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
```

Output.txt :


~/Desktop/output.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
output.txt

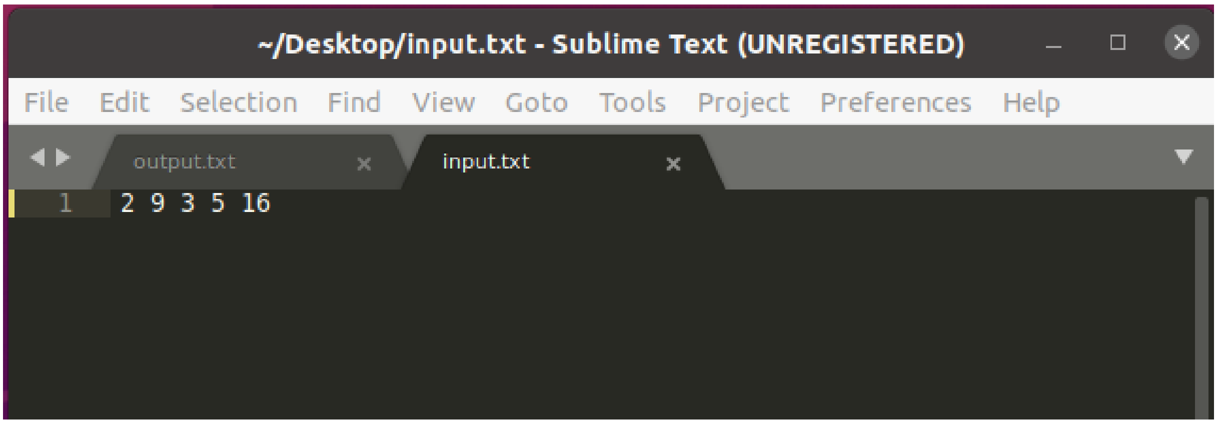
```
(5-3+5)*7 = 49

```

Another example :

İnput.txt :

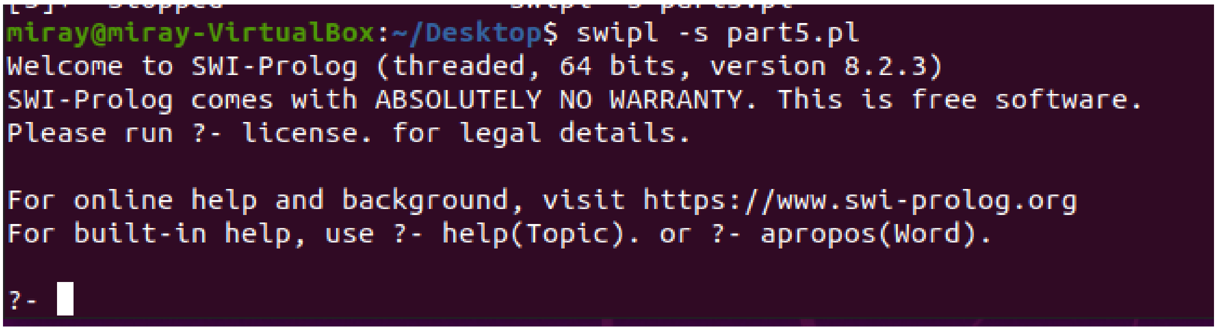
[2 9 3 5 16]

İnput.txt :


~/Desktop/input.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
output.txt
input.txt

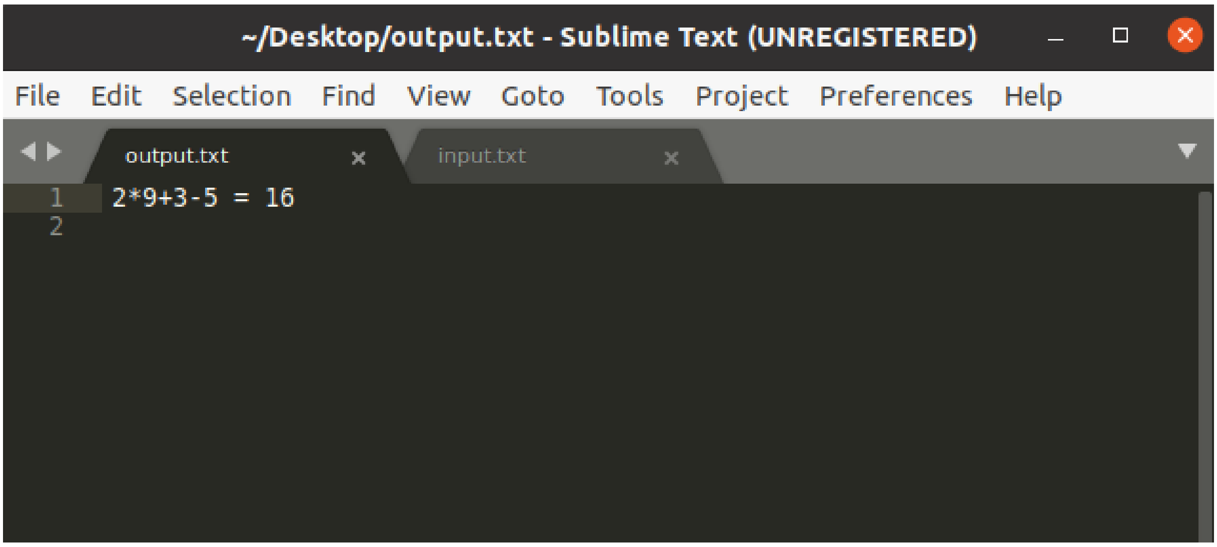
```
2 9 3 5 16
```

Compiler :



```
[3]+  Stopped                 swipl -s part5.pl
miray@miray-VirtualBox:~/Desktop$ swipl -s part5.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- 
```

Output.txt :


~/Desktop/output.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
output.txt
input.txt

```
2*9+3-5 = 16

```