

**Gebze Technical University
Department of Computer
Engineering**

**CSE 341 – Homework 4 Report
(PROLOG)**

DİLARA KARAKAŞ

171044010

PART 1

In this part, the facts were defined according to the figure. Then the route rule was written.

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

```
[?- route(istanbul,izmir).  
[true .  
  
[?- route(istanbul,edirne).  
false.  
  
?-
```

PART 2

In this part, first of all the distance facts were written. For this, the distances between cities were taken as follows:

The sroute rule statement, which gives the shortest flight between cities, is required. Then complete all the flights and distance are calculated.

The testbenches between Edremit Erzincan 1026 km. The distance between Istanbul and Burdur is 660 km via Izmir and Isparta.

```

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).

?- sroute(edremit,erzincan,X).
X = 1026.

?- sroute(istanbul,burdur,X).
X = 660.

?- sroute(istanbul,burdur,660).
true.

?- sroute(istanbul,burdur,340).
false.

```

PART 3

According to the statement given in the first part of this party, the facts were defined.

Classes		
Class	Time	Room
102	10	z23
108	12	z11
341	14	z06
455	16	207
452	17	207

Enrollment	
Student	Class
a	102
a	108
b	102
c	108
d	341
e	455

PART 3.1

Predicate “schedule(S,P,T)” that associates a student to a place and time of class.

```

[Dilara-MacBook-Air:171044010_karakaş_dilara_hw4 dilarakarakas$ swipl 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).

[?- schedula(a,z23,10).
Correct to: "schedule(a,z23,10)"?
Please answer 'y' or 'n'? yes
>true .

[?- schedula(b,P,T).
Correct to: "schedule(b,P,T)"? yes
P = z23,
T = 10 .

?- 

```

PART 3.2

Another predicate “usage(P,T)” that gives the usage times of a classroom. See the example query and its result.

```

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

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

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

[?- usage(207,17).
true.

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

```

PART 3.3

Another predicate “conflict(X,Y)” that gives true if X and Y conflicts due to classroom or time. And I added class 402 (conflict 341) time 14 for only used in this part(3.3).

```
[?- conflict(102,108).  
false.  
  
[?- conflict(341,108).  
false.  
  
[?- conflict(455,452).  
false.  
  
[?- conflict(402,341).  
true.
```

PART 3.4

Another predicate “meet(X,Y)” that gives true if student X and student Y are present in the same classroom at the same time.

```
[?- meet(a,a).  
true .  
  
[?- meet(a,b).  
true .  
  
[?- meet(a,d).  
false.  
  
[?- meet(a,c).  
true.  
  
[?- meet(a,e).  
false.
```

PART 4.1

Define a Prolog predicate “element(E,S)” that returns true if E is in S.

```
[?- element(1,[1,2,3]).  
true.  
  
[?- element(5,[1,2,3]).  
false.  
  
?- ]
```

PART 4.2

Define a Prolog predicate “union(S1,S2,S3)” that returns true if S3 is the union of S1 and S2.

```
[?- union([3,5,2],[1,5],[3,2,1,5]).  
true .  
  
[?- union([3,5,2],[1,5],U).  
U = [3, 2, 1, 5] .  
  
[?- union([3,5,2],[1,5],[1,2,3,4]).  
false.
```

PART 4.3

Define a Prolog predicate “intersect(S1,S2,S3)” that returns true if S3 is the intersection of S1 and S2.

```
?- intersection([1,2,3,4],[2,3],[2,3]).
true.

?- intersection([1,2,3,4],[2,3],L).
L = [2, 3].

?- intersection([1,2,3,4],[2,3],[7,5]).
false.

?- intersection([7,7,5,2],[2,5],[5,2]).
true.

?- intersection([7,7,5,2],[2,5],L).
L = [5, 2].

?- intersection([7,7,5,2],[2,5],[7,13]).
false.
```

PART 4.4

Define a Prolog predicate “equivalent(S1,S2)” that returns true if S1 and S2 are equivalent sets.

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

?- equalivalent([1,2,3],[3,2,1]).
true.

?- equalivalent([1,2,3],[3,1,2]).
true.

?- equalivalent([1,2,3],[6,7,5]).
false.

?- █
```