# CSE341 – Programming Languages (Fall 2020)
# Homework #4 Report

Berk Pekgöz
171044041


SWI Prolog used for this Assignment.

## Part1:

**Logic:** There is a helper predicate which goes through possible routes and adds the city passed through. While going through checks from the list to know if already passed through or did not.

Example output:

```
% c:/Users/Berk/Desktop/PL HW4/part1.pl compiled 0.00 sec, 0 clauses
?- route(edirne,X).
X = erzincan ;
X = edremit.
```

## Part2:

**Logic:** There is a helper predicate route_distance which goes through possible routes and adds the city passed through and adds distence between previous city to “Distance”. While going through checks from the list to know if already passed through or did not. At the end route_distance returns distance between given cities with possible routes. And then to get all possible distance between two cities findall called with route_distance and DisList become list of all possible distances. Then find_min finds minimum distance in the list and returns.

Example output:

```
% c:/Users/Berk/Desktop/PL HW4/part2.pl compiled 0.00 sec, 68 clauses
?- sroute(edremit,erzincan,X).
X = 1066.
```

**Note**: This output different than the output in the Assignment pdf but Distance between Edremit and Erzincan is shown as 1066 on given website(https://www.distancecalculator.net/).

# Part3:

**Logic:**

- **3.1)** Gets students' class and finds its time and place with when_ and where.
- **3.2)** Gets place as input and finds which class then checks time that class.
- **3.3)** Gets two class as input and gets times of those then compare or gets places of those then compare.
- **3.4)** Gets two student and then gets their schedule. After that compare the times and places of those schedules.

Example output:

```
% c:/Users/Berk/Desktop/PL HW4/part3.pl compiled 0.02 sec, 21 clauses
?- when_(102,10).
true.

?- schedule(a,P,T).
P = z23,
T = 10 ;
P = z11,
T = 12.

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

?- conflict(455,452).
true.

?- meet(a,b).
true
```

**Note**: Some outputs can be different than the outputs in pdf but those are incorrect and corrected by Mr. Dede later.

**Note 2**: Name of the fact "when" is replaced by "when_" because SWI Prolog says there is already a predicate named "when" in standard libraries.

# Part4:

**Logic:**

- **4.1)** Goes through all elements of the list and finds the given element.
- **4.2)** First creates union of S1 and S2 then compare with S3. If equal, then returns true.
- **3.3)** First creates intersection of S1 and S2 then compare with S3. If equal, then returns true.
- **3.4)** First compare the size of the lists. If they are equal, then checks for If all elements of S2 is in the S1.

Example output:

```
% c:/Users/Berk/Desktop/PL HW4/part4.pl compiled 0.00 sec, 16 clauses
?- element(5,[1,2,3,4,5,6,7]).
true .

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

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

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

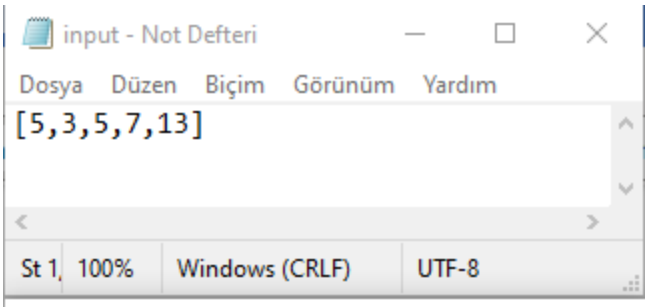
# Part5:

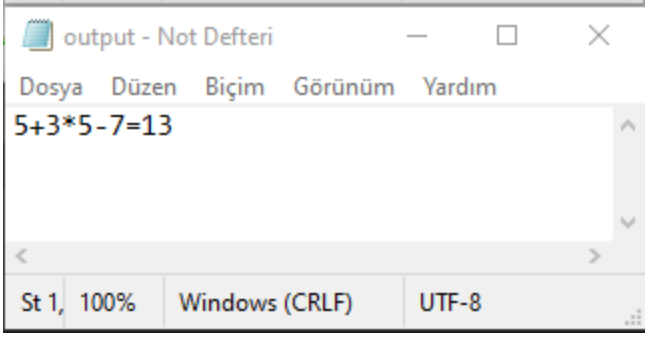
**Logic:** There are 4 four possible operation (add/sub/div/mult) and we have a predicate for each operation, but their predicate names are same. For add predicate, subtracts E1 from Res and calls same predicate with rest of the list and Newres. For sub predicate, subtracts E1 from Res, multiplies E2 with –1 and calls same predicate with E2 and rest of the list and Newres. For mult/div predicate multiply/divide E1 E2 and call the same predicate with Result of that multiply/divide + Rest. At each prediction, symbol of operation added to a list and at the end operator symbols merged with integers.

Example output:

```
% c:/Users/Berk/Desktop/PL HW4/part5.pl compiled 0.00 sec, 12 clauses
?- find_equation().
true .
```

File input/output:




**Note**: File names are given in Assignment pdf. And input.txt and output.txt must be in the "Prolog" folder. SWI Prolog forced them to be in that folder. In my computer that folder is in "User/Documents/".