

CSE505 – Spring 2021
Assignment 4 Problem 1
(may be done by a team of two students)

Problem 1 is given below. A problem on lambda calculus to be posted later.

Problem 1: Constraints (due: May 3, 11:59 pm)

Consider the following two relations:

db_student(S, L, M, F) – a student **S** has a list of assignment scores **L**, a midterm score **M**, and final exam score **F**. As the final exam has not yet taken place, the value for **F** is left unbound.

map(Lo, Hi, G) – meaning, the letter grade is **G** if the weighted average score $W \geq Lo$ and $W < Hi$, where the weighted average score **W** is obtained assuming that the assignments are collectively weighted 40% and the mid-term and final are weighted 30% each.

Sample facts for the two relations are shown below; they also in file `grade.pl` posted on Piazza.

db_student(tom, [85,95,80,75],75, _).	map(90, 100.01, 'A').
db_student(ding, [80,90,70,80],85, _).	map(80, 90, 'A-').
db_student(hari, [90,70,60,55],65, _).	map(70, 80, 'B+').
db_student(ann, [95,80,70,85],55, _).	map(60, 70, 'B').
db_student(aisha, [100,90,95,100],95, _).	map(50, 60, 'B-').
db_student(aidong, [70,65,70,55],65, _).	map(40, 50, 'C+').
db_student(zhazha, [5,5,5,5],10, _).	map(30, 40, 'C').
	map(20, 30, 'C-').
	map(10, 20, 'D').
	map(0, 10, 'F').

Using the above relations, define a Prolog predicate **grade(S,F,G)**, where **S** is a student, **F** is his/her final exam score, and **G** is his/her overall letter grade. The predicate should work for any combination of given input parameters and should produce the correct output for the remaining parameters. Examples:

```
?- grade(tom, 90, 'A').           % Can tom get an 'A' grade with 90 on the final?  
false
```

```
?- grade(tom, 90, Answer).       % What grade can tom get with 90 on final?  
Answer = 'A-'
```

A key aspect in this problem is the reporting of constraints as answers.

```
?- grade(Student, Final, 'A').   % Who can earn an 'A' on the course?
```

```
Student = aisha,  
{Final>=76.66666666666667, Final=<100.0, _13366=67.0+0.3*F} ;  
false.
```

For the above query, the only student who can earn an **A** grade is **aisha** provided her **Final** $\geq 76.666\dots$ and **Final** ≤ 100.0 .

```
?- grade(Student, F, 'A-').
```

*% Who can earn an 'A-' on the course?
% Report answers one at a time*

```
Student = tom,  
{F>=80.0, F<100.0, _18190=56.0+0.3*F} ;  
Student = ding,  
{F>=75.0, F<100.0, _22368=57.5+0.3*F} ;  
Student = aisha,  
{F>=43.33333333333336, F<76.66666666666667, _26510=67.0+0.3*F} ;  
false.
```

```
?- grade(zhazha, F, G).
```

*% What grade(s) can zhazha earn on the course?
% Report answers one at a time.*

```
G = 'C',  
{F>=83.33333333333334, F<100.0, _492=5.0+0.3*F} ;  
G = 'C-',  
{F>=50.0, F<83.33333333333334, _4460=5.0+0.3*F} ;  
G = 'D',  
{F>=16.666666666666668, F<50.0, _8542=5.0+0.3*F} ;  
G = 'F',  
{F>= -0.0, F<16.666666666666668, _12632=5.0+0.3*F}.
```

IMPORTANT:

The power of constraint programming is that you do **not** need to write any code to print out answer constraints. Answer constraints are automatically generated by Prolog provided the program has laid out the constraints correctly. You can define the **grade(S, F, G)** predicate with just *one clause* containing about 10 lines of code – assuming each goal and each constraint is written on a separate line.

Note: Prolog has two relevant builtin predicates: **sum_list(L, S)**, which returns in **S** the sum of the numbers in list **L**; and **length(L, N)**, which returns in **N** the number of elements in list **L**.

WHAT TO SUBMIT:

Prepare a top-level directory named **A4_Problem1_UBITId1_UBITId2** if the assignment is done by two students; otherwise, name it as **A4_Problem1_UBITId** if the assignment is done solo. (Order the **UBITId**'s in alphabetic order, in the former case.) In this directory, place the file **grade.pl** containing the definitions of the predicates **db_student**, **map**, and **grade**.

Compress the directory and submit the compressed file using the online submission procedure – instructions posted at Resources → Assignments → Online_Submission.pdf. Only one submission per team is required.