

San José State University
Computer Science Department
CS152, Programming Paradigms, Springs 2022

Homework #2

Objective:

This homework's objective is to practice programming in Prolog programming language.

Details:

Solutions to all exercises for this homework assignment should be implemented in Prolog. Submit solution to each exercise in a separate .pl file. Make sure to include the following comments into each .pl file (replace with your information):

```
% CS152 Spring 2022
% Student name
% SJSU student ID
% Homework assignment 2
% Exercise X
```

Name each .pl file with the exercise number. For example, exercise1.pl, exercise2.pl, etc.

Exercise 1:

Implement a program that, given a list and an index k , outputs k 's element in that list. Name your rule ***get_i(List, K, E)***, where ***List*** is the list of elements, ***K*** is the index of the item to retrieve, and ***E*** is the variable that will hold the retrieved element. Indexing should start with 1 for this exercise.

An example of a query using this rule would be:

```
get_i([a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z], 3, E).
```

And the expected output should be: *c*

If the index is out of bounds the query should return *false*.

Exercise 2:

Implement a program that, given a list an element and an element value, outputs the index of that element in that list. If the element is not in the list, the query should output *false*. If there are multiple element values in the list, then the index of the first occurrence should be returned.

Homework # 2

Name your rule ***find(List, E, K)***, where ***List*** is the list of elements, ***E*** is the variable that will hold the retrieved element, and ***K*** is the index of the item to retrieve. Indexing should start with 1 for this exercise.

An example of a query using this rule would be:

`find([a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z], a, K).`

The expected output should be: *1*

An example of a query using this rule would be:

`find([a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z], b, K).`

The expected output should be: *2*

An example of a query using this rule would be:

`find([a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z], 9, K).`

The expected output should be: *false*

Exercise 3:

Implement a program that, given a list, outputs the value of the last element in that list. Name your rule ***get_last(List, E)***, where ***List*** is the list of elements, ***E*** is the variable that will hold the retrieved element. The query should return *false* for an empty list.

An example of a query using this rule would be:

`get_last([a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z], E).`

The expected output should be: *z*

Exercise 4:

Implement a program that, given two lists, outputs *true* if all the elements of the first list are in the second list, otherwise outputs *false*. Name your rule ***members(List1, List2)***, where ***List1*** is the first list, ***List2*** is the second list.

An example of a query using this rule would be:

`members([a,d,h],[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]).`

The expected output should be: *true*

An example of a query using this rule would be:

`members([a,d,8],[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]).`

The expected output should be: *false*

An example of a query using this rule would be:

`members([], [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z]).`

The expected output should be: *true*

Homework # 2

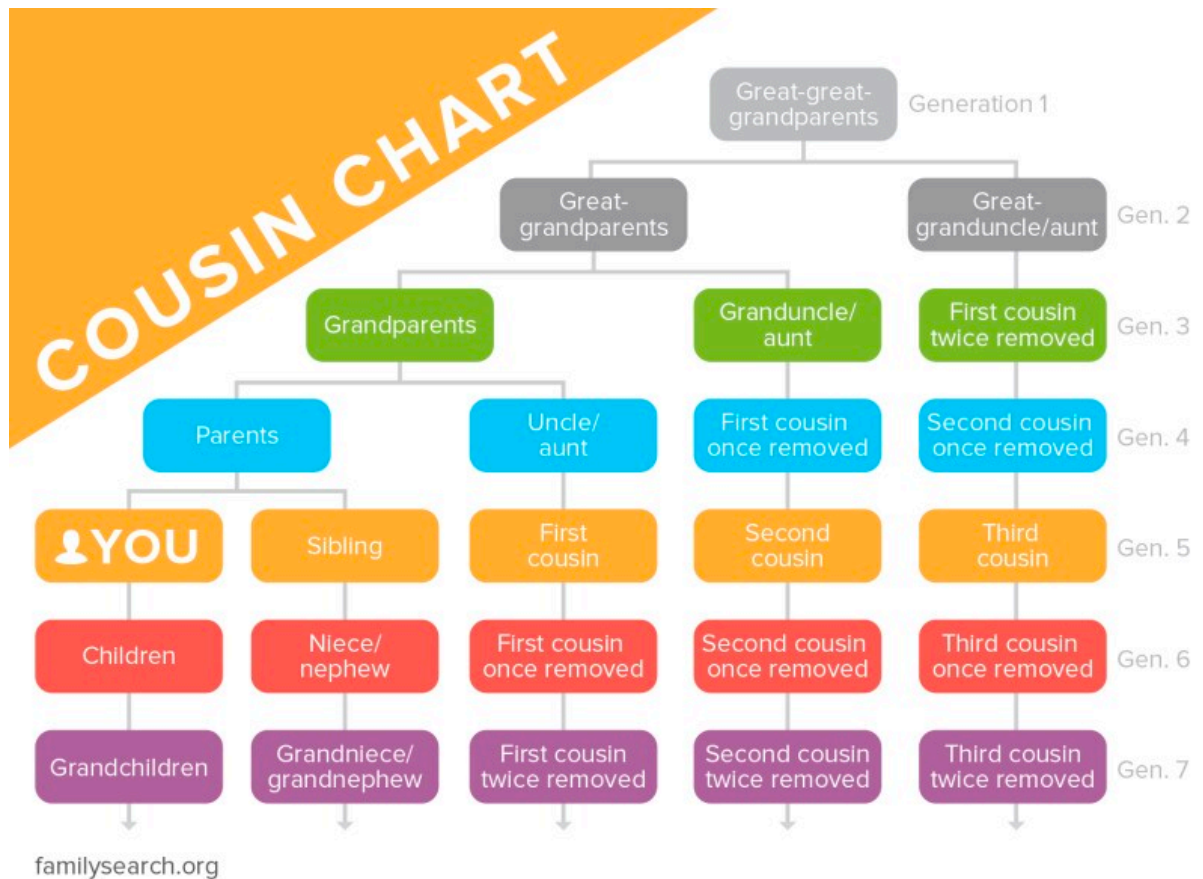
Exercise 5:

Remember our Prolog examples where we defined family relationships? Consider the following family (you can copy and paste these relationships into your program):

```
father_of(jose, david).  
father_of(nicolas, david).  
father_of(laura, david).  
father_of(tina, david).  
father_of(ricky, nicolas).  
father_of(mateo, nicolas).  
father_of(moira, greg).  
father_of(trevor, bruce).  
father_of(lisa, bruce).  
father_of(mary, jose).  
father_of(john, jose).
```

```
mother_of(ricky, lena).  
mother_of(jose, lupe).  
mother_of(nicolas, lupe).  
mother_of(laura, lupe).  
mother_of(tina, lupe).  
mother_of(mateo, lena).  
mother_of(moira, laura).  
mother_of(trevor, tina).  
mother_of(lisa, tina).  
mother_of(mary, cassandra).  
mother_of(john, cassandra).
```

For this exercise you are asked to define a rule that returns true if the two persons are first-cousins. Below is the chart for your information that gives definitions of different family relationships, including first-cousin relationship to a given individual:



Name your rule *first_cousin(A, B)*, where *A* is the first individual (e.g. john), *B* is the second individual (e.g. moira).

An example of a query using this rule would be:

`first_cousin(john, X).`

The expected output should be:

X	
ricky	1
mateo	2
moira	3
trevor	4
lisa	5

For this exercise you can implement additional “helper” rules that will make the *first_cousin* rule easier to code and more compact. Depending on how you implement your rule(s) you might notice duplicate values returned by the *first_cousin* query. That is because Prolog keeps track of all variable, including free variables. You can incorporate *distinct()* function into your query:

`distinct(first_cousin(john, X)).`

Submission:

Compress all the homework .pl files into a single compressed file named “*YourNameAssignment2*”, with the appropriate file extension. Make sure submit by 11:59pm on the due date listed in Canvas. Submit your solution via Canvas.

If you have any questions, message me or the grader or both:

Yulia.Newton@sjsu.edu

madhujitaranjit.ambaskar@sjsu.edu

Grading:

I will return the grades as fast as we can grade this homework. Normally it should not take more than a few weeks.

A total of 25 points are possible for this homework assignment.