

Programming Languages

Homework 7

Due Wednesday, February 26th at 2 AM

Download this homework assignment, and fill in your answers, then convert this document into pdf as a report to GradeScope. Be sure you know how to trace programs like this on paper; you will do it on the quiz and on the final.

1. **[30]** Consider the following procedure in Ada-like syntax (with 1-indexed arrays):

```
PROCEDURE Main IS
  i: integer := 3;
  type vector is array (1..3) of integer;
  a: vector := (3, 1, 2);
  PROCEDURE Put (v: vector) IS
    BEGIN
      Put_Line ("A = (");
      FOR i IN v'range LOOP
        Put_Line (v(i));
        Put_Line (" ");
      END LOOP;
      Put_Line (")");
    END Put;
  PROCEDURE Do_Something (t: mode integer) IS
    BEGIN
      i := i - 1;
      t := i + 1;
      a(t) := a(i) + 1;
      a(i) := a(t) + 1;
      i := i - 1;
      a(t) := t - 1;
      a(i) := a(t) - 1;
    END Do_Something;
  BEGIN
    Do_Something(a(i));
    Put (a);
  END Main;
```

What would be the output, assuming the parameter passing mode is pass by:

e.g. copy: value	<code>A = (1 3 2)</code>
a. reference	<code>A = (? ? ?)</code>
b. copy: value-result	<code>A = (? ? ?)</code>
c. name	<code>A = (? ? ?)</code>

2. [20] Write a Prolog program to answer questions about moving throughout a house. You should describe the access via doors with the 'door_between' relation. Describe your house and the relation 'path_from' as follows.

```
% facts about your house, real or imaginary
door_between(bed_room, hidden_chamber).
door_between(hidden_chamber, basement).
% how do I get from the bed_room to the basement?
?- path_from(bed_room, basement, Y).
% go from bed_room to hidden_chamber to the basement
Y = [bed_room, hidden_chamber, basement]
% where can I go from the basement and how do I get there?
?- path_from(Y, basement, Z).
```

Requirements:

- All doors are bidirectional, but the fact will only provide one direction for each pair.
- There could be loops of room design, but each room would be accessed at most once in a path.
- You may only use "=", "\==", "\+", "member/2", "append/3".
- [Test Cases](#)

YOUR CODE:

3. [20] Klefstad is hosting a party of an international organization. Not everybody speaks the same language, and some of them speak more than 1 language. He invited 10

guests and – including himself – he plans to have a ROUND TABLE CHATTING SESSION. To make everybody comfortable, he sets the following rules:

- Each person must be able to have a conversation with both people they are seated next to (in some language).
- No two females sit next to each other.
- You can use the following information about the invitees and the host for testing your solution to the problem but **DO NOT** include these in the final code you submit. I will provide you with guests and the languages they speak (in the form of male(name). (or female) and speaks(name, language). If you include the below information in your final submission it is likely you will not get the correct answer on gradescope:
 - **Klefstad, Bill, Emily, Heidi, and Isaac speak English.**
 - **Beth, Mark, Susan, and Isaac can speak French.**
 - **Klefstad, Bill, Susan, Fred, and Jane speak Spanish.**
 - **Klefstad, Bill, Mark, Isaac, and Fred identify as Male.**
 - **Emily, Heidi, Beth, Susan, and Jane identify as Female.**

Your program should be able to come up with a seating in a ring (around a round table) to satisfy the constraints described above. e.g.

```
?- party_seating(L).
```

```
L = [jane, klefstad, susan, bill, emily, isaac, heidi, fred,  
beth, mark].
```

This is a made-up sample answer, not a correct answer, but shows how your program should show correct seating.

YOUR CODE:

4. [30] Write a Prolog program to do symbolic differentiation of polynomials with respect to x. Fun fact, this was the first program I wrote in Prolog. My second program was symbolic integration which is harder.
- What you are allowed to use:
 - you only need “=”, “+”, “-”, “*”, “/”, “^”, “is”, “atomic/1”, “number/1”
 - you may use the cut symbol, “!”, only after it finds an answer, so prevents it from returning the same answer again.
 - For any symbol times a constant, put the constant in the front: 2*x instead of x*2
 - Simplify as much as you can

Use the following test cases:

```

?- deriv(x^2, Y).
Y = 2*x.
?- deriv((x^2*x)/x, Y).
Y = 2.
?- deriv(x^4+2*x^3-x^2+5*x-1/x, Y).
Y = 4*x^3+6*x^2-2*x+5+1/x^2.
?- deriv(4*x^3+6*x^2-2*x+5+1/x^2, Y).
Y = 12*x^2+12*x-2-2/x^3.
?- deriv(12*x^2+12*x-2-2/x^3, Y).
Y = 24*x+12+6/x^4.

```

YOUR CODE:

Submit two things:

- a report with this exact template and your answers to question 1 on Gradescope
- submit the file “main.pl” that contains all your function definitions inside on Gradescope so we may run them through [MOSS](#)