IS MSc Artificial Intelligence Programming II

Exercise 5
(Issued: week 5)

January 29, 2002

For the following exercises you will need store telephone entries in a list. Each entry should be of the following general form

[[<name of person>][<gender><telephone no.><school>][<interests>]]

e.g.
[[mary jones][female 477238 SMS][welding yoga dance archery]] or
[[fred alfred smith][male 345262 COGS][yoga cooking programming]] or
[[jack higgins][male no_phone EURO][dance knitting cars]] or
[[susan augustina louise bloggs][female 12345 COGS][football]]

1. Create a POP-11 list called **information_list** with at least 10 such telephone entries.

2. Write a procedure **show_entries** that makes use of procedure foreach to print out the complete list of entries in **information_list**, one entry per line.

3. Write a procedure **show_details** that takes two arguments namely gender and school. The procedure should print out all the entries in **information_list**, one per line, which have both the given gender and the given school.

   e.g.

   show_details("male","COGS");
   ** [[fred alfred smith][male 345262 COGS][yoga cooking programming]]

4. Write a procedure **add_clients** that prompts the user for details of a new client, checks to see if the client's name is not already in **information_list**, prompts for the gender, telephone number, school and

interests and then adds the entry to **information_list**. The procedure should loop until the user types nomore in response to the request for a new client name. e.g.

add_clients();
** [what is the name of the next client]
? mary jones
** [we already have details of this person]
** [what is the name of the next client]
? mark brown
** [what is the gender of mark brown]
? male
** [what is the telephone number of mark brown]
? 34567
** [what is the school of mark brown]
? AFRAS
** [what are the interests of mark brown]
? hang_gliding raffia reading
** [what is the name of the next client]
? nomore

Check that your procedure adds new entries to **information_list** in the correct form.

5. Write a procedure called **overlap** that takes two lists as arguments and returns <true>if the two lists have any element in common and otherwise returns <false>e.g.

   overlap([a b c d e],[f g h i j])=>
   ** <false>
   overlap([yoga cricket marbles],[rugby electronics yoga dance]) =>
   ** <true>

PTO

6. Write a procedure called **find_friend** that takes a list of interests as its only argument and prints out, one per line, every entry in **information_list** that has at least one interest in common with the list supplied e.g.

   find_friend([food yoga knitting]);
   ** [mary jones 477238]
   ** [fred alfred smith 345262]
   ** [jack higgins no_phone]

7. Write a procedure called **pairs** that prints out the names of every compatible pair of persons in **information_list**. To be compatible each person of a pair must have at least one interest in common with the other person. A person cannot form a pair with him/herself. e.g.

   pairs();
   ** [mary jones AND fred alfred smith]
   ** [mary jones AND jack higgins]
   ** [fred alfred smith AND mary jones]
   ** [jack higgins AND mary jones]

8. Fix **pairs** so that each pair is printed once only e.g.

   new_pairs();
   ** [mary jones AND fred alfred smith]
   ** [mary jones AND jack higgins]