LAB 7 | Artificial Intelligence

Aim : Study Compound objects and Functors in PROLOG.

1. Modify the sample program II so that it will also print the birth dates of the people listed. Next, add telephone numbers to the report.

Code :

```
domains
        name = person(symbol,symbol)
        birthday = b_date(symbol,integer,integer)
        ph_num = symbol

predicates
        phone_list(name,symbol,birthday)
        get_months_birthdays
        convert_month(symbol,integer)
        check_birthday_month(integer,birthday)
        write_person_birthdate_mobileno(name,birthday,ph_num)

clauses
        get_months_birthdays:-
                        write("******************** This Month's Birthday List ********************"),nl,
                        write("First name\tLast Name\tBirth Date\tMobile No\n"),
                        write("************************************************************************"),nl,
                        date(_, This_month, _),
                        phone_list(Person, Mobile, Date),
                        check_birthday_month(This_month, Date),
                        write_person_birthdate_mobileno(Person,Date,Mobile),
                        fail.

        get_months_birthdays:-
                                write("\n\n Press any key to continue: "),nl,
                                readchar(_).


write_person_birthdate_mobileno(person(First_name,Last_name),b_date(M,D,Y),Mobile):-
                        write(" ",First_name,"\t\t ",Last_name,"\t\t",M,"-",D,"-",Y,"\t",Mobile),nl.

check_birthday_month(Mon,b_date(Month,_,_)):-
                                        convert_month(Month,Month1),
                                        Mon = Month1.
```

```
phone_list(person(apurva, mehta), "767-8463", b_date(jan, 13, 1955)).
phone_list(person(apurva, shah), "438-8400", b_date(feb, 04, 1985)).
phone_list(person(apurva, parikh), "555-5653", b_date(mar, 22, 1935)).
phone_list(person(apurva, doshi), "767-2223", b_date(apr, 04, 1951)).
phone_list(person(apurva, joshi), "555-1212", b_date(may, 31, 1962)).
phone_list(person(apurva, baxi), "438-8400", b_date(jun, 13, 1980)).
phone_list(person(apurva, dave), "767-8463", b_date(jun, 22, 1986)).
phone_list(person(apurva, bhatt), "555-5653", b_date(jul, 22, 1981)).
phone_list(person(apurva, patel), "767-2223", b_date(aug, 13, 1981)).
phone_list(person(apurva, dangar), "438-8400", b_date(sep, 22, 1981)).
phone_list(person(apurva, pandya), "438-8400", b_date(sep, 31, 1952)).
phone_list(person(apurva, vaishnav), "555-1212", b_date(nov, 22, 1984)).
phone_list(person(apurva, gor), "767-2223", b_date(sep, 04, 1987)).
phone_list(person(apurva, kanani), "438-8400", b_date(dec, 31, 1981)).

convert_month(jan, 1).
convert_month(feb, 2).
convert_month(mar, 3).
convert_month(apr, 4).
convert_month(may, 5).
convert_month(jun, 6).
convert_month(jul, 7).
convert_month(aug, 8).
convert_month(sep, 9).
convert_month(oct, 10).
convert_month(nov, 11).
convert_month(dec, 12).
```

Output:

```
Goal : get_months_birthdays
**********************************This Month's Birthday List**********************************
First name          Last Name              Birth Date          Mobile No
************************************************************************************************
apurva              danagar              sep-22-1981          438-8400
apurva              pandya               sep-31-1952          438-8400
apurva              gor                  sep-4-1987           767-2223
```

2. Write a prolog program for an IT company that stores employee details like Name, Address, Department,Position, Salary. Use compound objects to properly formulate the representation of each employee's details. Find out,

   I. employee(s) with salary higher than a threshold.
   II. employee(s) available in a particular department.
   III. employee(s) holding a particular position.

Code :

```
domains
        name = person(first,last)
        location = address(street, city, state, zip)
        first, last, street, city, state, zip, department, position = symbol
        salary =integer

predicates
        employee(name,location,department,position,salary).
        employee_with_salary_higher_than_5000.
        employee_available_particular_department(department).
        employee_with_particular_position(position).
        write_name_salary(name,salary).
        write_name(name).

clauses

employee(person("Raj","Panchal"),address("HONEY
PARK","Surat","Gujarat","395009"),"Development","Senior Head",15000).

employee(person("Parth","Patel"),address("Dungari","Valsad","Gujarat","324856"),"Developmen
t","Senior Head",12000).

employee(person("Siddhi","Shah"),address("Ring
Road","Vadodra","Gujarat","375002"),"Marketing","Junior",700).

employee(person("Rutu","Joshi"),address("M G Road","Bhavnagar","Gujarat","362009"),"Human
Resource","Fresher",20000).

employee(person("Pranav","Patel"),address("Jalapor","Navsari","Gujarat","322009"),"Quality
Assurance","Senior Head",4000).

employee(person("Prachi","Shah"),address("Kabilpor","Navsari","Gujarat","322059"),"Quality
Assurance","Junior",2000).
```

```
employee_with_salary_higher_than_5000:-
                                        write("Employee with 50000 salary"),nl,
                                        write("-------------------------"),nl,
                                        employee(Name,_,_,_,Salary),
                                        Salary>5000,
                                        write_name_salary(Name,Salary),
                                        fail.

 employee_available_particular_department(Department):-
                                    write("Employee with ",Department," Department"),nl,
                                    write("-------------------------"),nl,
                                    employee(Name,_,Department,_,_),
                                    write_name(Name),
                                    fail.

 employee_with_particular_position(Position):-
                                        write("Employee with ",Position," Position"),nl,
                                        write("-------------------------"),nl,
                                        employee(Name,_,_,Position,_),
                                        write_name(Name),
                                        fail.

 write_name_salary(person(First, Last), Salary):-
                                                write(First," ",Last,"\t",Salary),nl.

 write_name(person(First,Last)):-
                                    write(First," ",Last),nl.
```

Output :

```
Goal : employee_with_salary_higher_than_5000
Employee with 5000 salary
---------------------------------------
Raj Panchal       15000
Parth Patel       12000
Rutu Joshi        20000
No

Goal : employee_available_particular_department(“Development”)
Employee with Development Department
------------------------------------------------------------
Raj Panchal
Parth Patel
No
```

```
Goal : employee_with_particular_position("Junior")
Employee with Junior Position
-----------------------------------------
Siddhi Shah
Prachi Shah
No
```

3. Try the following link and verify whether the system is intelligent or not and justify your answer.
   www.manifestation.com/neurotoys/eliza.php3

- ELIZA emulates a Rogerian psychotherapist.
- ELIZA has almost no intelligence whatsoever, only tricks like string substitution and canned responses based on keywords.
- The illusion of intelligence works best, however if you limit your conversation to talking about yourself and your life.
- Hence , ELIZA is a dump.