**LAB#9**

**Question#1**

Code is provided to you. Run and understand the code

**Solution:**

;;; This expert system is a sample which test if someone

;;; could be ill, based on his temperature, and on

;;; is exposition or no to the sun (recently).

;;; (ie. difference fever vs hyperthermia : (https://en.wikipedia.org/wiki/Hyperthermia))

;; Function to concatenate "string"

(defun concatString (list)

"A non-recursive function that concatenates a list of strings."

(if (listp list)

(let ((result ""))

(dolist (item list)

(if (stringp item)

(setq result (concatenate 'string result item))))

result)))

;; Return 'T' if argument is not nil

(defun isNotNil (list)

(cond ((null list) nil)

(T)))

;; Get right pronoun

(defun getRightPronoun (genderOneLetterList)

(cond ((eq genderOneLetterList 'F) "she")

((eq genderOneLetterList 'M) "he")

(nil)))

;; We define a structure to store each 'person'

(defstruct person firstname (age 0) gender (temperature 0) wasSunExposed)

;;; Facts database

;; We make multiple instances of person.

; Reminder : 'LISP uses the self-evaluating symbol nil to mean false. Anything other than nil means true.'

(setq lea (make-person :firstname "Lea" :age 22 :gender 'F :temperature 39.4 :wasSunExposed 1))

(setq luca (make-person :firstname "Luca" :age 20 :gender 'M :temperature 37.5 :wasSunExposed 1))

(setq manon (make-person :firstname "Manon" :age 24 :gender 'F :temperature 36.8 :wasSunExposed 1))

(setq mathis (make-person :firstname "Mathis" :age 21 :gender 'M :temperature 37.7 :wasSunExposed nil))

(setq jade (make-person :firstname "Jade" :age 23 :gender 'F :temperature 37.7 :wasSunExposed nil))

(setq mael (make-person :firstname "Mael" :age 25 :gender 'M :temperature 36.2 :wasSunExposed 1))

(setq lina (make-person :firstname "Lina" :age 22 :gender 'F :temperature 38.7 :wasSunExposed nil))

(setq paul (make-person :firstname "Paul" :age 22 :gender 'M :temperature 39.2 :wasSunExposed 1))

(setq luna (make-person :firstname "Luna" :age 23 :gender 'F :temperature 36.1 :wasSunExposed 1))

;; Put them in a list (with (list instance1 instance2 instanceN) )

(setq personsToListenList (list lea luca manon mathis jade mael lina paul luna))

; We can acces to instance 'lina' only by writing 'lina' in listener

;; We created the conditions that will serve varables of stops, Stop condition

(setq initWhatToCheckOnEachPersonList '(

(setq hypothermia nil) ; less than 36.5

(setq highTemperature nil) ; hyperthermia or fever

(setq hyperthermia nil) ; sun exposed + more than 37.5

(setq fever nil) ; not sun exposed + equal or more than 38

(setq deathLikely nil) ; more than 41.5 (irreversible brain complications)

))

;; Temporary variables

(setq currentPersonListen nil)

(setq currentPersonTemperature nil)

(setq wasCurrentPersonSunExposed nil)

;; Diagnostics

(setq hypothermiaDiagnostic '(concatString (list (person-firstname currentPersonListen)

": Beware the person does not seem to be able to maintain its internal temperature, "

(write-to-string (person-temperature currentPersonListen))

", " (getRightPronoun (person-gender currentPersonListen)) " is hypothermia."

)))

(setq hyperthermiaDiagnostic '(concatString (list (person-firstname currentPersonListen)

": Beware " (person-firstname currentPersonListen)

" seems to be unable to regulate its internal temperature, "

(write-to-string (person-temperature currentPersonListen))

", perhaps due to too long exposure to sunlight."

)))

(setq feverDiagnostic '(concatString (list (person-firstname currentPersonListen)

": Beware " (person-firstname currentPersonListen)

" seems to be sick. Indeed its internal temperature is high, "

(write-to-string (person-temperature currentPersonListen))

", maybe because " (getRightPronoun (person-gender currentPersonListen)) " has been exposed to the sun."

)))

(setq deathLikelyDiagnostic '(concatString (list (person-firstname currentPersonListen)

": DANGER risk of death of " (person-firstname currentPersonListen)

", temperature to high, "

(write-to-string (person-temperature currentPersonListen))

", more than 41.5 degrees C. From this temperature, irreversible brain complications are likely."

)))

;;; Rules database

;; Define 'newrule', which will active all 'cond'

(defun newrule (nom predicats actionsToDo)

(cond ((eval predicats)

(eval actionsToDo))))

(setq rulesList '(

;; hypothermia

(newrule 'ruleTemperatureTooLowHypothermia '(< currentPersonTemperature 36.5) '(setq hypothermia 1))

;; highTemperature

(newrule 'ruleTemperatureTooHigh '(> currentPersonTemperature 37.5) '(setq highTemperature 1))

;; hyperthermia

(newrule 'ruleTemperatureTooHighHyperthermia 'highTemperature '(cond ((isNotNil wasCurrentPersonSunExposed) (setq hyperthermia 1))))

;; fever

(newrule 'ruleTemperatureTooHighFever 'highTemperature '(cond ((null wasCurrentPersonSunExposed) (setq fever 1))))

;; deathLikely

(newrule 'ruleTemperatureTooHighDeathLikely '(> currentPersonTemperature 41.5) '(setq deathLikely 1))

)) ; End (setq list) rules

;; Apply 'eval' on each element of a list

(defun evalList (liste)

(mapc 'eval liste))

;;; Inference engine

; Reminders :

; (car list) -> list's first element

; (cdr list) -> all list's elements except first

;; Run engine

; while ( list not empty ) -> evaluate all persons

(defun runEngine ()

(cond (personsToListenList

(evalList initWhatToCheckOnEachPersonList)

; Set temporary variables with current person

(setq currentPersonListen (car personsToListenList))

(setq currentPersonTemperature (person-temperature currentPersonListen))

(setq wasCurrentPersonSunExposed (person-wasSunExposed currentPersonListen))

; We evaluate the rules (which use temporary variables assigned above)

(evalList rulesList)

; It displays any problems : diagnostics

(cond (hypothermia (print (eval hypothermiaDiagnostic))))

(cond (hyperthermia (print (eval hyperthermiaDiagnostic))))

(cond (fever (print (eval feverDiagnostic))))

(cond (deathLikely (print (eval deathLikelyDiagnostic))))

; The first element is removed (<=> person listen)

(setq personsToListenList (cdr personsToListenList))

(runEngine))

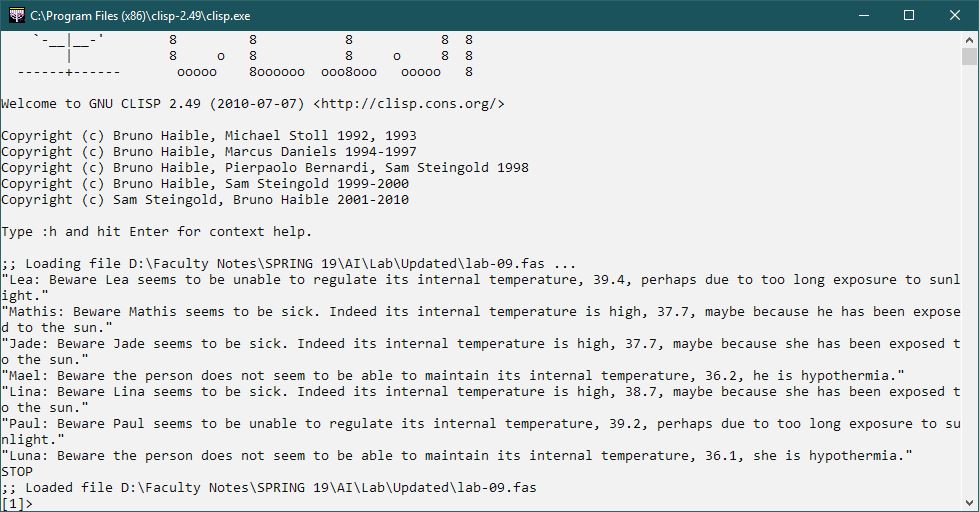
(T (print 'stop)))

)

;; Run engine

(runEngine)

**OUTPUT:**

****

**QUESTION#2**

Test the Expert System to diagnose whether following patients has hypothermia or simply high fever. Also check whether patient’s high temperature is due to his long exposure to sun. hence, check whether patient has high risk of death or not.

Patient name : Sarah , Age : 38 , Gender : Female, Temperature: 39.2 , sun exposure: No,

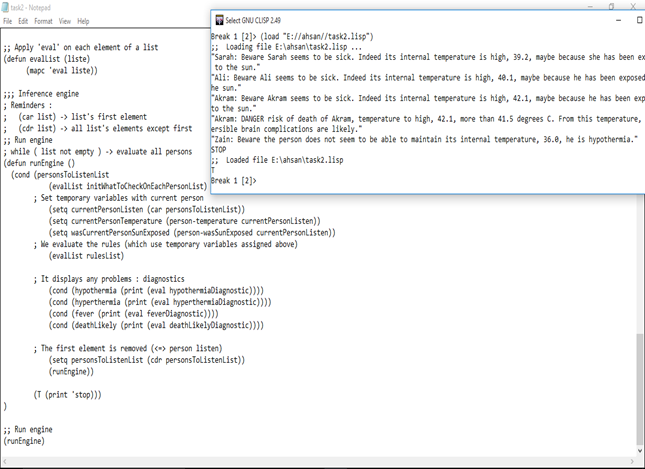
Patient name : Amna, Age : 27 , Gender : Female, Temperature : 37.1 , sun exposure: Yes

Patient name : Ali, Age : 30 , Gender : Male, Temperature: 40.1 , sun exposure: No

Patient name : Akram, Age : 30 , Gender : Male, Temperature: 42.1 , sun exposure: No

Patient name : Zain, Age : 20 , Gender : Male, Temperature : 36.0 , sun exposure: Yes

**SOLUTION:**

****