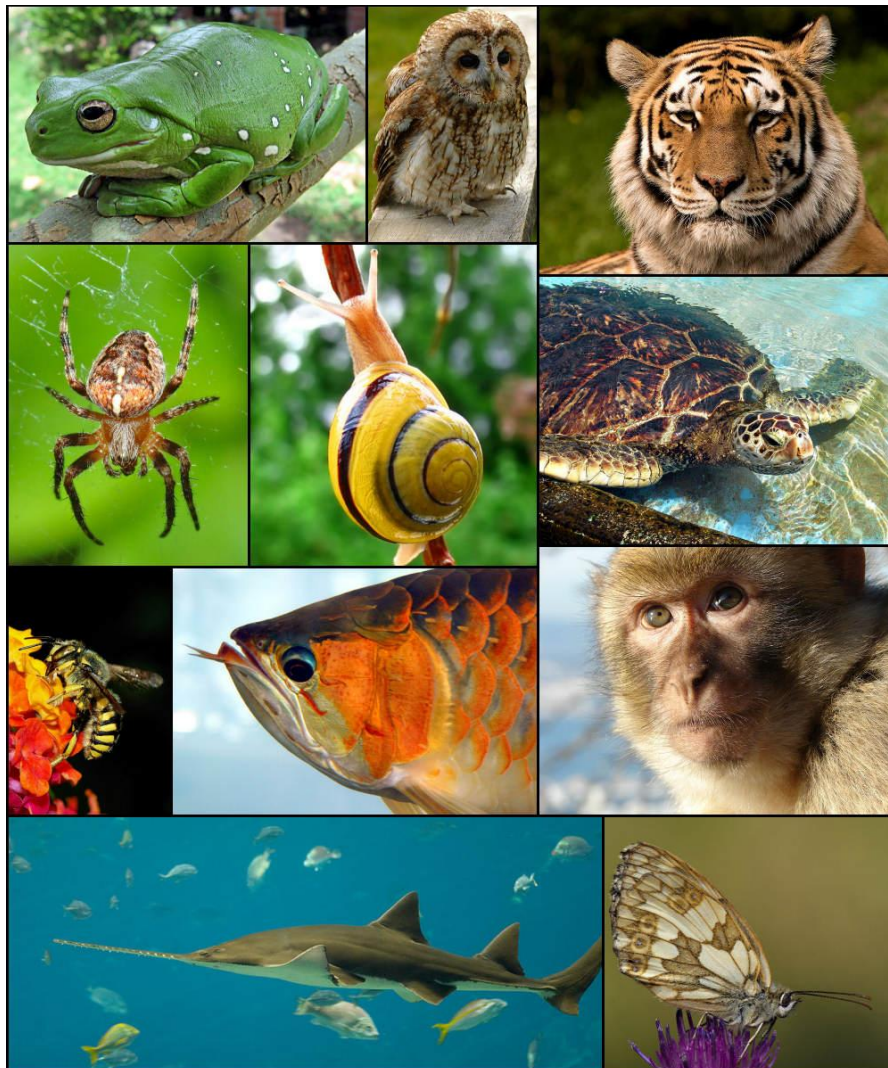


Name of project : Animal Identification Expert System

A simple expert system which attempts to identify an animal based on its characteristics .

Programming by : Mohammed Al Thobiti

License : GPL V.3



Notes:

- I devolved this project depending on other similar project.
- I wrote the code from zero.
- This project is released under GPL license, so you can modify / redistribute it's without any restrictions.

Name of Expert system : Animal Identification ES .

Domain of Expert system : Education / Classification .

Interface : Yes or No questions .

Knowledge representation method : Production rules (if -> then)

Objectives :

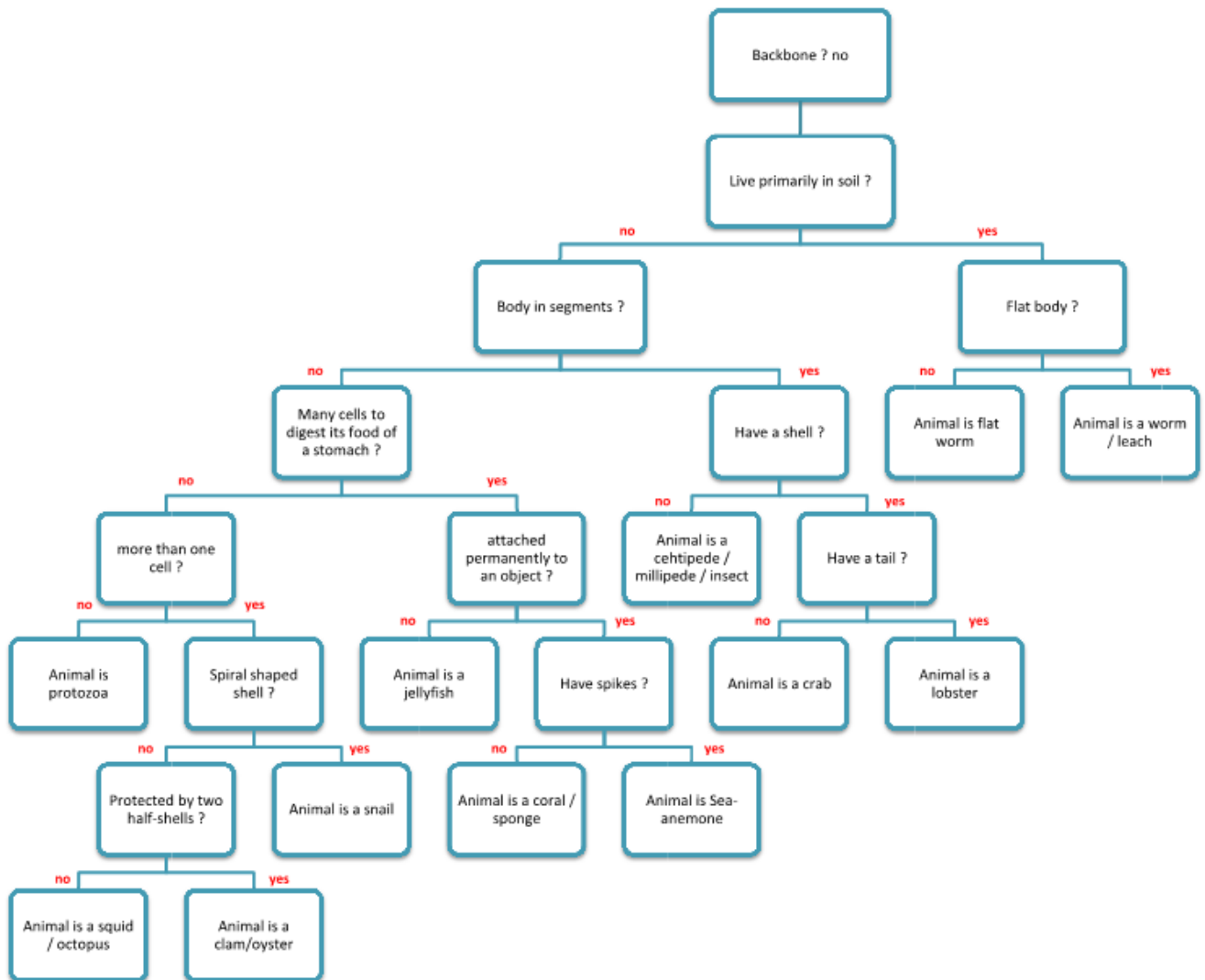
A simple expert system which attempts to identify a non backbone/backbone animal based on its characteristics.

Tools used in this project : CLIPS Shell / VC++ .

Data structure technique : Graph (And / OR) .

Inference : Forward chaining . (Data driven) .

Graph of the system shown in next page :



Sample Of Rules :

Rule 1 :

if (Backbone=no And Live primarily in soil=yes And Flat body = yes)

Then

Animal is a worm/leach .

Rule 2 :

if (Backbone=no And Live primarily in soil=yes And Flat body = no)

Then

Animal is a flat worm.

Rule 3 :

if (Backbone=no And Live primarily in soil=no And Body in segments = yes And Have a shell=no)

Then

Animal is a centipede/millipede/insect .

Rule 4 :

if (Backbone=no And Live primarily in soil=no And Body in segments = yes And Have a shell=yes And Have a tail=yes)

Then

Animal is a lobster .

Rule 5 :

if (Backbone=no And Live primarily in soil=no And Body in segments = yes And Have a shell=yes And Have a tail=no)

Then

Animal is a crab .

Rule 6 :

if (Backbone=no And Live primarily in soil=no And Body in segments = no And Many cells to digest its food of a stomach=yes And attached permanently to an object=yes And Have Spikes=yes)

Then

Animal is Sea-anemone.

Rule 7 :

if (Backbone=no And Live primarily in soil=no And Body in segments = no And Many cells to digest its food of a stomach=yes And attached permanently to an object=yes And Have Spikes=no)

Then

Animal is coral / sponge.

Rule 8 :

if (Backbone=no And Live primarily in soil=no And Body in segments = no And Many cells to digest its food of a stomach=yes And attached permanently to an object=no)

Then

Animal is jellyfish.

Rule 9 :

if (Backbone=no And Live primarily in soil=no And Body in segments = no And Many cells to digest its food of a stomach=no And Animal have more than on cell=no)

Then

Animal is protozoa.

Rule 10 :

if (Backbone=no And Live primarily in soil=no And Body in segments = no And Many cells to digest its food of a stomach=no And Animal have more than on cell=yes And spiral shaped shell=yes)

Then

Animal is a snail.

Rule 11 :

if (Backbone=no And Live primarily in soil=no And Body in segments = no And Many cells to digest its food of a stomach=no And Animal have more than on cell=yes And spiral shaped shell=yes)

Then

Animal is a snail.

Rule 12 :

if (Backbone=no And Live primarily in soil=no And Body in segments = no And Many cells to digest its food of a stomach=no And Animal have more than on cell=yes And spiral shaped shell=no And Protected by two half-shells=yes)

Then

Animal is a clam/oyster.

Rule 13 :

if (Backbone=no And Live primarily in soil=no And Body in segments = no And Many cells to digest its food of a stomach=no And Animal have more than on cell=yes And spiral shaped shell=no And Protected by two half-shells=no)

Then

Animal is a Animal is a squid / octopus.

The code :

```
9 (defrule main-backbone
10 (initial-fact)
11 =>
12 (printout t "===== "crlf)
13 (printout t " Animal Identification Expert System "crlf)
14 (printout t " Programming by: Mohammed D. Al Thobiti "crlf)
15 (printout t " E-mail:mmml11m@gmail.com "crlf)
16 (printout t " A simple expert system which attempts to identify "crlf)
17 (printout t " an animal based on its characteristics. "crlf)
18 (printout t "===== "crlf)
19 (printout t"Does your animal have a backbone ?(yes/no) ")
20 (assert(why))
21 (assert(backbone(read)))
22
23 ;;;=====
24 (defrule non-backbone
25 (backbone no)
26 =>
27 (printout t"Does your animal live primarily in soil ?(yes/no) ")
28 (assert(live-in-soil(read)))
29
30 ;;;=====
31 (defrule soil-yes
32 (live-in-soil yes)
33 =>
34 (printout t "Does your animal have a flat body? (yes/no) ")
35 (assert(flat-body(read)))
36 ;;;=====
37 (defrule flat-body-yes
38 (flat-body yes)
39 =>
40 (printout t"Your animal is a flat-worm" crlf)
41 (bind $?animal flat-worm)
42 (assert(animal-is $?animal)))
43 ;;;=====
44 (defrule flat-body-no
45 (flat-body no)
46 =>
47 (printout t"Your animal is a worm/leach" crlf)
48 (bind $?animal worm leach)
49 (assert(animal-is $?animal)))
50 ;;;=====
51 (defrule soil-no
52 (live-in-soil no)
53 =>
54 (printout t "Is the animals body in segments? (yes/no)")
55 (assert(body-in-segments(read)))
56 ;;;=====
57 (defrule body-segments-yes
58 (body-in-segments yes)
59 =>
60 (printout t"Does your animal have a shell? (yes/no)")
61 (assert(animal-have-shell(read)))
```



```

62 ;;=====
63 (defrule shell=yes
64 (animal-have-shell yes)
65 =>
66 (printout t"Does you animal have a tail? (yes/no)")
67 (assert(animal-have-tail(read))))
68 ;;=====
69 (defrule tail=yes
70 (animal-have-tail yes)
71 =>
72 (printout t"Your animal is a lobster"crLf)
73 (bind $?animal lobster)
74 (assert(animal-is $?animal)))
75 ;;=====
76 (defrule tail=no
77 (animal-have-tail no)
78 =>
79 (printout t"Your animal is a crab" crLf)
80 (bind $?animal crab)
81 (assert(animal-is $?animal)))
82 ;;=====
83 (defrule shell=no
84 (animal-have-shell no)
85 =>
86 (printout t"Your animal is a cehtiped/millipede/insect " crLf)
87 (bind $?animal cehtiped millipede insect)
88 (assert(animal-is $?animal)))
89 ;;=====
90 (defrule body-segments=no
91 (body-in-segments no)
92 =>
93 (printout t"Does your animal use many cells to digest its food insted of a
stomach?(yes/no)")
94 (assert(many-cells(read))))
95 ;;=====
96 (defrule many-cells-to-digest=yes
97 (many-cells yes)
98 =>
99 (printout t"Is your animal attached permanently to an object? (yes/no)")
100 (assert(animal-attached-to-object(read))))
101 ;;=====
102 (defrule no-attached-to-object
103 (animal-attached-to-object no)
104 =>
105 (printout t"Your animal is a jellyfish " crLf)
106 (bind $?animal jellyfish)
107 (assert(animal-is $?animal)))
108 ;;=====
109 (defrule yes-attachment-to-object
110 (animal-attached-to-object yes)
111 =>
112 (printout t"Does your animal normally have spikes radiating fromits body?
(yes/no) ")
113 (assert(have-spike(read))))
114 ;;=====
115 (defrule spike=yes
116 (have-spike yes)
117 =>
118 (printout t "Your animal is Sea-anemone " crLf)
119 (bind $?animal Sea-anemone)
120 (assert(animal-is $?animal)))

```

```

121 ;;;=====
122 (defrule spike-no
123 (have-spike no)
124 =>
125 (printout t "Your animal is coral/sponge " crlf)
126 (bind $?animal coral sponge)
127 (assert(animal-is $?animal)))
128 ;;;=====
129 (defrule many-cells-to-digest-no
130 (many-cells no)
131 =>
132 (printout t "Is your animal made up of arethan one cell? (yes/no) ")
133 (assert(one-cell(read))))
134 ;;;=====
135 (defrule one-cell-no
136 (one-cell no)
137 =>
138 (printout t "Your animal is Protozoa " crlf)
139 (bind $?animal Protozoa)
140 (assert(animal-is $?animal)))
141 ;;;=====
142 (defrule one-cell-yes
143 (one-cell yes)
144 =>
145 (printout t "Does your animal have a shaped-shell? (yes/no) " crlf)
146 (assert(spiral-shaped-shell(read))))
147 ;;;=====
148 (defrule spiral-shaped-shell-yes
149 (spiral-shaped-shell yes)
150 =>
151 (printout t "Your animal is a snail " crlf)
152 (bind $?animal snail)
153 (assert(animal-is $?animal)))
154 ;;;=====
155 (defrule spiral-shaped-shell-no
156 (spiral-shaped-shell no)
157 =>
158 (printout t "Is your animal protected by two half-shells? (yes/no) " crlf)
159 (assert(two-half-shells(read))))
160 ;;;=====
161 (defrule two-shell-yes
162 (two-half-shells yes)
163 =>
164 (printout t "Your animal is a clam/oyster " crlf)
165 (bind $?animal clam oyster)
166 (assert(animal-is $?animal)))
167
168 (defrule two-shell-no
169 (two-half-shells no)
170 =>
171 (printout t "Your animal is a squid/octopus " crlf)
172 (bind $?animal squid octopus)
173 (assert(animal-is $?animal)))
174
175 ;;;=====
176 ;;;***
177 ;;;=====
178 ;;;***
179 ;;;=====
180 (defrule yes-backbone
181 (backbone yes)

```

```

182 =>
183 (printout t"Is the animal warm blooded ?(yes/no) ")
184 (assert(warm-blooded(read)))
185 ;;=====
186 (defrule warm-blooded-no
187 (warm-blooded no)
188 =>
189 (printout t"IS your animal always in water? (yes/no) ")
190 (assert(animal-in-water(read)))
191
192 (defrule animal-always-in-water-yes
193 (animal-in-water yes)
194 =>
195 (printout t "Does your animal have a boney skeketon? (yes/no) ")
196 (assert(boney-skeketon(read)))
197
198 (defrule boney-skeketon-yes
199 (boney-skeketon yes)
200 =>
201 (printout t "Your animal is a fish " crlf)
202 (bind $?animal fish)
203 (assert(animal-is $?animal)))
204
205 (defrule boney-skeketon-no
206 (boney-skeketon no)
207 =>
208 (printout t "Your animal is a shark/ray " crlf)
209 (bind $?animal shark ray)
210 (assert(animal-is $?animal)))
211
212 (defrule animal-always-in-water-no
213 (animal-in-water no)
214 =>
215 (printout t "Is your animal covered with scaled skin? (yes/no) ")
216 (assert(scaled-skin(read)))
217
218 (defrule scaled-skin-no
219 (scaled-skin no)
220 =>
221 (printout t"Does your animal jump? (yes/no) ")
222 (assert(animal-jump(read)))
223
224 (defrule jump-yes
225 (animal-jump yes)
226 =>
227 (printout t "Your animal is a frog " crlf)
228 (bind $?animal frog)
229 (assert(animal-is $?animal)))
230
231 (defrule jump-no
232 (animal-jump no)
233 =>
234 (printout t "Your animal is a salamander " crlf)
235 (bind $?animal salamander)
236 (assert(animal-is $?animal)))
237
238 (defrule scaled-skin-yes
239 (scaled-skin yes)
240 =>
241 (printout t"Does the animal have a rounded shell? (yes/no) ")
242 (assert(rounded-shell(read)))

```

```
243
244 (defrule rounded-shell-yes
245 (rounded-shell yes)
246 =>
247 (printout t "Your animal is a trurtle " crlf)
248 (bind $?animal trurtle)
249 (assert(animal-is $?animal)))
250
251 (defrule rounded-shell-no
252 (rounded-shell no)
253 =>
254 (printout t "Does your animal have limbs? (yes/no) " crlf)
255 (assert(have-limbs(read))))
256
257 (defrule limbs-yes
258 (have-limbs yes)
259 =>
260 (printout t "Your animal is a crocodile/alligator " crlf)
261 (bind $?animal crocodile alligator)
262 (assert(animal-is $?animal)))
263
264 (defrule limbs-no
265 (have-limbs no)
266 =>
267 (printout t "Your animal is a snake " crlf)
268 (bind $?animal snake)
269 (assert(animal-is $?animal)))
270
271 (defrule warm-blooded-yes
272 (warm-blooded yes)
273 =>
274 (printout t "Does the female of your animal nurse its young with milk? (yes/no)
275 ")
276 (assert(drink-milk(read))))
277
278 (defrule bird
279 (drink-milk no)
280 =>
281 (printout t "Your animal is a bird/penguin " crlf)
282 (bind $?animal bird penguin)
283 (assert(animal-is $?animal)))
284
285 (defrule drink-milk-yes
286 (drink-milk yes)
287 =>
288 (printout t "Does your animal eat red meat? (yes/no) ")
289 (assert(eat-red-meat(read))))
290
291 (defrule eat-red-meat-yes
292 (eat-red-meat yes)
293 =>
294 (printout t "Can your animal fly? (yes/no)")
295 (assert(animal-fly(read))))
296
297 (defrule fly-yes
298 (animal-fly yes)
299 =>
300 (printout t "Your animal is a bat " crlf)
301 (bind $?animal bat)
302 (assert(animal-is $?animal)))
```

```

303 (defrule fly-no
304 (animal-fly no)
305 =>
306 (printout t "Does Your animal have an opposing thumb? (yes/no) " )
307 (assert(oppesing-thumb(read))))
308
309 (defrule oppesing-thumb=yes
310 (oppesing-thumb yes)
311 =>
312 (printout t "Does your animal have a prehensile tail? (yes/no)" )
313 (assert(prehensile-tail(read))))
314
315 (defrule prehensile-tail=yes
316 (prehensile-tail yes)
317 =>
318 (printout t "Your animal is monkey" crlf)
319 (bind $?animal monkey)
320 (assert(animal-is $?animal)))
321
322 (defrule prehensile-tail=no
323 (prehensile-tail no)
324 =>
325 (printout t "Is Your animal nearly hairless? (yes/no) " )
326 (assert(hairless(read))))
327
328 (defrule hairless=yes
329 (hairless yes)
330 =>
331 (printout t "Your animal is man" crlf)
332 (bind $?animal man)
333 (assert(animal-is $?animal)))
334
335 (defrule hairless=no
336 (hairless no)
337 =>
338 (printout t "Does your animal have long powerfull arms? (yes/no) " )
339 (assert(have-long-arms(read))))
340
341 (defrule arms=yes
342 (have-long-arms yes)
343 =>
344 (printout t "Your animal is orangutan/gorilla/chimpanzie " crlf)
345 (bind $?animal orangutan gorilla chimpanzie)
346 (assert(animal-is $?animal)))
347
348 (defrule arms=no
349 (have-long-arms no)
350 =>
351 (printout t "Your animal is babon " crlf)
352 (bind $?animal babon)
353 (assert(animal-is $?animal)))
354
355 (defrule oppesing-thumb=no
356 (oppesing-thumb no)
357 =>
358 (printout t "Does an adult normally weight over 400 pounds? (yes/no)" )
359 (assert(weight-over400(read))))
360
361 (defrule weight-over400=yes
362 (weight-over400 yes)
363 =>

```

```
364 (printout t "Is your animal land based? (yes/no) ")
365 (assert(land-based(read)))
366
367 (defrule land-based-yes
368 (land-based yes)
369 =>
370 (printout t "Your animal is a bear/tiger/lion" crlf)
371 (bind $?animal bear tiger lion)
372 (assert(animal-is $?animal)))
373
374 (defrule land-based-no
375 (land-based no)
376 =>
377 (printout t "Your animal is a walrus" crlf)
378 (bind $?animal walrus)
379 (assert(animal-is $?animal)))
380
381 (defrule weight-over400-no
382 (weight-over400 no)
383 =>
384 (printout t "Does your animal have a thin tail? (yes/no) ")
385 (assert(thin-tail(read)))
386
387 (defrule thin-tail-yes
388 (thin-tail yes)
389 =>
390 (printout t "Your animal is a Cat " crlf)
391 (bind $?animal Cat)
392 (assert(animal-is $?animal)))
393
394 (defrule thin-tail-no
395 (thin-tail no)
396 =>
397 (printout t "Your animal is a cayote/wolf/fox/dog " crlf)
398 (bind $?animal cayote wolf fox dog)
399 (assert(animal-is $?animal)))
400
401 (defrule eat-red-meat-no
402 (eat-red-meat no)
403 =>
404 (printout t "Does your animal have hooves? (yes/no)")
405 (assert(have-hooves(read)))
406
407 (defrule have-hooves-yes
408 (have-hooves yes)
409 =>
410 (printout t "Does your animal stand on two toes/hoves per foot? (yes/no) ")
411 (assert(two-toes(read)))
412
413 (defrule two-toes-no
414 (two-toes no)
415 =>
416 (printout t "Is your animal covered with a protective plating? (yes/no) ")
417 (assert(protective-plating(read)))
418
419 (defrule protective-plating-yes
420 (protective-plating yes)
421 =>
422 (printout t "Your animal is a rhinoceros " crlf)
423 (bind $?animal rhinoceros)
424 (assert(animal-is $?animal)))
```

```
425
426 (defrule protective-plating-no
427 (protective-plating no)
428 =>
429 (printout t "Your animal is a horse/zebra " crlf)
430 (bind $?animal horse zebra)
431 (assert (animal-is $?animal)))
432
433 (defrule two-toes-yes
434 (two-toes yes)
435 =>
436 (printout t "Does your animal have horns? (yes/no) ")
437 (assert (have-horns (read))))
438
439 (defrule horns-no
440 (have-horns no)
441 =>
442 (printout t "Does your animal normally live in the desert? (yes/no) ")
443 (assert (live-in-desert (read))))
444
445 (defrule desert-yes
446 (live-in-desert yes)
447 =>
448 (printout t "Your animal is a camel " crlf)
449 (bind $?animal camel)
450 (assert (animal-is $?animal)))
451
452 (defrule desert-no
453 (live-in-desert no)
454 =>
455 (printout t "Your animal is a giraffe " crlf)
456 (bind $?animal giraffe)
457 (assert (animal-is $?animal)))
458
459 (defrule horns-yes
460 (have-horns yes)
461 =>
462 (printout t "Does your animal have one horn? (yes/no) ")
463 (assert (one-in-horn (read))))
464
465 (defrule one-horn-yes
466 (one-in-horn yes)
467 =>
468 (printout t "Your animal is a hippopotamus " crlf)
469 (bind $?animal hippopotamus)
470 (assert (animal-is $?animal)))
471
472 (defrule one-horn-no
473 (one-in-horn no)
474 =>
475 (printout t "Does your animal have fleece? (yes/no) " )
476 (assert (have-fleece (read))))
477
478 (defrule fleece-yes
479 (have-fleece yes)
480 =>
481 (printout t "Your animal is a sheep/goat " crlf)
482 (bind $?animal sheep goat)
483 (assert (animal-is $?animal)))
484
485 (defrule fleece-no
```

```

486 (have-fleece no)
487 =>
488 (printout t "Is your animal domesticated ? (yes/no) ")
489 (assert(domesticated(read)))
490
491 (defrule domesticated-yes
492 (domesticated yes)
493 =>
494 (printout t "Your animal is a cow " crlf)
495 (bind $?animal cow)
496 (assert(animal-is $?animal)))
497
498 (defrule domesticated-no
499 (domesticated no)
500 =>
501 (printout t "Your animal is a deer/moose/antelope " crlf)
502 (bind $?animal deer moose antelope)
503 (assert(animal-is $?animal)))
504
505 (defrule have-hooves-no
506 (have-hooves no)
507 =>
508 (printout t "Does your animal live in water? (yes/no) ")
509 (assert(live-in-water(read)))
510
511 (defrule live-in-water-yes
512 (live-in-water yes)
513 =>
514 (printout t "Is your animal unfortunately, commercially hunted ? (yes/no) ")
515 (assert(commmercially-hunted(read)))
516
517 (defrule hunted-yes
518 (commmercially-hunted yes)
519 =>
520 (printout t "Your animal is a whale " crlf)
521 (bind $?animal whale)
522 (assert(animal-is $?animal)))
523
524 (defrule hunted-no
525 (commmercially-hunted no)
526 =>
527 (printout t "Your animal is a dolphin/porpoise " crlf)
528 (bind $?animal dolphin porpoise)
529 (assert(animal-is $?animal)))
530
531 (defrule live-in-water-no
532 (live-in-water no)
533 =>
534 (printout t "Does your animal large front teeth ? (yes/no) ")
535 (assert(large-front-teeth(read)))
536
537 (defrule front-teeth-yes
538 (large-front-teeth yes)
539 =>
540 (printout t "Does your animal have large ears? (yes/no) ")
541 (assert(large-ears(read)))
542
543 (defrule large-ears-yes
544 (large-ears yes)
545 =>
546 (printout t "Your animal is a rabbit " crlf)

```



```

547 (bind $?animal rabbit)
548 (assert (animal-is $?animal)))
549
550 (defrule large-ears-no
551 (large-ears no)
552 =>
553 (printout t "Your animal is unknown " crlf))
554
555 (defrule front-teeth-no
556 (large-front-teeth no)
557 =>
558 (printout t "Does your animal have a pouch? (yes/no) ")
559 (assert (have-pouch (read))))
560
561 (defrule pouch-yes
562 (have-pouch yes)
563 =>
564 (printout t "Your animal is a kangaroo/koala bear " crlf)
565 (bind $?animal kangaroo koala-bear)
566 (assert (animal-is $?animal)))
567
568 (defrule pouch-no
569 (have-pouch no)
570 =>
571 (printout t "Your animal is a mole/shrew/elephant " crlf)
572 (bind $?animal mole shrew elephant)
573 (assert (animal-is $?animal)))
574
575
576
577 ;;=====
578 ;;***
579 ;;=====
580 ;;***
581 ;;=====
582 ;;***
583 ;;=====
584 ;;***
585
586 (defrule explanation
587 (why)
588 =>
589 (printout t "===== " crlf)
590 (retract 0)
591 (facts)
592 (printout t "===== " crlf)
593 )

```

Testing of the system :

CLIPS> (run)

```
=====
Animal Identification Expert System
A simple expert system which attempts to identify
an animal based on its characteristics.
=====
```

```
Does your animal have a backbone ?(yes/no) no
Does your animal live primarily in soil ?(yes/no) yes
Does your animal have a flat body? (yes/no) yes
Your animal is a flat-worm
```

CLIP>>

```
Does your animal have a backbone ?(yes/no) no
Does your animal live primarily in soil ?(yes/no) no
Is the animals body in segments? (yes/no)no
Does your animal use many cells to digest its food insted of a
stomach?(yes/no)yes
Is your animal attached permanently to an object? (yes/no)yes
Does your animal normally have spikes radiating fromits body? (yes/no)
yes
Your animal is Sea-anemone
```

