(deftemplate car

(slot car\_name)

(slot body\_type)

(slot brand)

(slot engineCC)

(slot price))

(deffacts carmodel

(car (car\_name “Rusa EX”) (body\_type mpv) (brand perodua) (engineCC 1.3) (price 12000))

(car (car\_name "Sage FLX") (body\_type sedan) (brand proton) (engineCC 1.3) (price 33000))

(car (car\_name "Axia SE") (body\_type hatchback) (brand perodua) (engineCC 1.0) (price 40000))

(car (car\_name Iriz) (body\_type hatchback) (brand proton) (engineCC 1.3) (price 43000))

(car (car\_name "Myvi SE") (body\_type hatchback) (brand perodua) (engineCC 1.5) (price 53000))

(car (car\_name Avanza) (body\_type mpv) (brand toyota) (engineCC 1.5) (price 63000))

(car (car\_name Alza) (body\_type mpv) (brand perodua) (engineCC 1.5) (price 64000))

(car (car\_name Suprima) (body\_type hatchback) (brand proton) (engineCC 1.5) (price 77000))

(car (car\_name Inspira) (body\_type sedan) (brand proton) (engineCC 1.7) (price 80000))

(car (car\_name "Carrola Altis") (body\_type sedan) (brand toyota) (engineCC 2.0) (price 110000))

(car (car\_name Camry) (body\_type sedan) (brand toyota) (engineCC 2.5) (price 145000))

(car (car\_name Alphard) (body\_type mpv) (brand toyota) (engineCC 3.5) (price 324000)))

(assert (start))

(defrule question1

(start)

=>

(printout t crlf "---------------------------------------------------------------" crlf)

(printout t " Welcome to Trace Your Favourite Car System " crlf)

(printout t "---------------------------------------------------------------" crlf)

(printout t "Thank you for using this system, what can I call you?" crlf)

(assert (name (read))))

(defrule question2

(name ?a)

=>

(printout t crlf "Hi " ?a ", please answering some question so I will show you the car" crlf)

(assert (go)))

(defrule choosing

(go)

=>

(printout t crlf crlf “Please select which one is the most important thing that you to prefer to buy a car” crlf)

(printout t “1. Body Type” crlf)

(printout t “2. Brand” crlf)

(printout t “3. Price” crlf)

(printout t “4. Don’t care” crlf)

(printout t “What is your choice (entering number) : “ )

(assert (choice (read))))

(defrule ChooseError

?fact <- (go)

?fact1 <- (choice ?)

(and ( and (and (not (choice 1)) (not (choice 2))) (not (choice 3))) (not (choice 4)))

=>

(printout t crlf “Wrong input!!!” )

(retract ?fact)

(retract ?fact1)

(assert (go)))

(defrule checkBodyType

(choice 1) (name ?a)

=>

(printout t crlf “Which body type of car you prefer? “crlf)

(printout t “1. HatchBack” crlf)

(printout t “2. Sedan” crlf)

(printout t “3. MPV” crlf)

(printout t “What is your choice (entering number) : “)

(assert (body (read)))

(printout t crlf ?a “, the car you can buy is: “ crlf))

(defrule CeckBodyTypeError

?fact <- (choice 1)

?fact1 <- (body ?)

( and (and (not (body 1)) (not (body 2))) (not (body 3)))

=>

(printout t crlf “Wrong input!!!” )

(retract ?fact)

(retract ?fact1)

(assert (choice 1)))

(defrule bodyHatch

(body 1)

(car (car\_name ?a) (body\_type hatchback) (brand ?c) (engineCC ?d) (price ?e))

=>

(printout t crlf “Car Name: “ ?a crlf)

(printout t “Body Type: HatchBack “ crlf)

(printout t “Brand: “ ?c crlf)

(printout t “Engine CC: “ ?d crlf)

(printout t “Price: RM “ ?e crlf)

(assert (tot1 ?e)))

(defrule cl1

?fact <- (tot1 ?a)

(not (test (= ?a 40000)))

=>

(retract ?fact))

(defrule BodyCont1

?fact <- (tot1 ?a)

?fact1 <- (body 1)

(test (= ?a 40000))

=>

(printout t “Want to continue? [yes / no] ”)

(retract ?fact)

(retract ?fact1)

(assert (con (read))))

(defrule bodySedan

(body 2)

(car (car\_name ?a) (body\_type sedan) (brand ?c) (engineCC ?d) (price ?e))

=>

(printout t crlf “Car Name: “ ?a crlf)

(printout t “Body Type: Sedan “ crlf)

(printout t “Brand: “ ?c crlf)

(printout t “Engine CC: “ ?d crlf)

(printout t “Price: RM “ ?e crlf)

(assert (tot2 ?e)))

(defrule cl2

?fact <- (tot2 ?a)

(not (test (= ?a 33000)))

=>

(retract ?fact))

(defrule BodyCont2

?fact <- (tot2 ?a)

?fact1 <- (body 2)

(test (= ?a 33000))

=>

(printout t “want to continue? [yes / no] ”)

(retract ?fact)

(retract ?fact1)

(assert (con (read))))

(defrule bodyMPV

(body 3)

(car (car\_name ?a) (body\_type mpv) (brand ?c) (engineCC ?d) (price ?e))

=>

(printout t crlf “Car Name: “ ?a crlf)

(printout t “Body Type: MPV “ crlf)

(printout t “Brand: “ ?c crlf)

(printout t “Engine CC: “ ?d crlf)

(printout t “Price: RM “ ?e crlf)

(assert (tot3 ?e)))

(defrule cl3

?fact <- (tot3 ?a)

(not (test (= ?a 12000)))

=>

(retract ?fact))

(defrule BodyCont3

?fact <- (tot3 ?a)

?fact1 <- (body 3)

(test (= ?a 12000))

=>

(printout t “want to continue? [yes / no] ”)

(retract ?fact)

(retract ?fact1)

(assert (con (read))))

(defrule checkbrand

(choice 2) (name ?a)

=>

(printout t crlf “Which brand of car you prefer? “crlf)

(printout t “1. Proton” crlf)

(printout t “2. Toyota” crlf)

(printout t “3. Perodua” crlf)

(printout t “What is your choice (entering number) : “)

(assert (bd (read)))

(printout t crlf ?a “, the car you can buy is: “ crlf))

(defrule CeckBrandError

?fact <- (choice 2)

?fact1 <- (bd ?)

( and (and (not (bd 1)) (not (bd 2))) (not (bd 3)))

=>

(printout t crlf “Wrong input!!!” )

(retract ?fact)

(retract ?fact1)

(assert (choice 2)))

(defrule brandProton

(bd 1)

(car (car\_name ?a) (body\_type ?b) (brand proton) (engineCC ?d) (price ?e))

=>

(printout t crlf “Car Name: “ ?a crlf)

(printout t “Body Type: “ ?b crlf)

(printout t “Brand: Proton “ crlf)

(printout t “Engine CC: “ ?d crlf)

(printout t “Price: RM “ ?e crlf)

(assert (tot4 ?e)))

(defrule cancel1

?fact <- (tot4 ?a)

(not (test (= ?a 33000)))

=>

(retract ?fact))

(defrule goCont2

?fact <- (tot4 ?a)

?fact1 <- (bd 1)

(test (= ?a 33000))

=>

(printout t “Want to continue? [yes / no] ”)

(retract ?fact)

(retract ?fact1)

(assert (con (read))))

(defrule brandToyota

(bd 2)

(car (car\_name ?a) (body\_type ?b) (brand toyota) (engineCC ?d) (price ?e))

=>

(printout t crlf “Car Name: “ ?a crlf)

(printout t “Body Type: “ ?b crlf)

(printout t “Brand: Toyota “ crlf)

(printout t “Engine CC: “ ?d crlf)

(printout t “Price: RM “ ?e crlf)

(assert (tot5 ?e)))

(defrule cancel2

?fact <- (tot5 ?a)

(not (test (= ?a 63000)))

=>

(retract ?fact))

(defrule goCont3

?fact <- (tot5 ?a)

?fact1 <- (bd 2)

(test (= ?a 63000))

=>

(printout t “Want to continue? [yes / no] ”)

(retract ?fact)

(retract ?fact1)

(assert (con (read))))

(defrule brandPerodua

(bd 3)

(car (car\_name ?a) (body\_type ?b) (brand perodua) (engineCC ?d) (price ?e))

=>

(printout t crlf “Car Name: “ ?a crlf)

(printout t “Body Type: “ ?b crlf)

(printout t “Brand: Perodua “ crlf)

(printout t “Engine CC: “ ?d crlf)

(printout t “Price: RM “ ?e crlf)

(assert (tot6 ?e)))

(defrule cancel3

?fact <- (tot6 ?a)

(not (test (= ?a 12000)))

=>

(retract ?fact))

(defrule goCont4

?fact <- (tot6 ?a)

?fact1 <- (bd 3)

(test (= ?a 12000))

=>

(printout t “Want to continue? [yes / no] ”)

(retract ?fact)

(retract ?fact1)

(assert (con (read))))

(defrule prices

(choice 3) (name ?a)

=>

(printout t ?a “, what is your badget? : RM ” )

(assert (badget (read)))

(printout t crlf “The car you can buy is:” crlf))

(defrule sugcar

(badget ?a)

(car (car\_name ?b) (body\_type ?c) (brand ?d) (engineCC ?e) (price ?f))

(test (> ?a ?f))

=>

(printout t crlf “Car Name: “ ?b crlf)

(printout t “Body Type: “ ?c crlf)

(printout t “Brand: “ ?d crlf)

(printout t “Engine CC: “ ?e crlf)

(printout t “Price: RM “ ?f crlf)

(assert (total ?f)))

(defrule cancel

?fact <- (total ?a)

(not (test (= ?a 12000)))

=>

(retract ?fact))

(defrule goCont5

?fact <- (total ?a)

?fact1 <- (badget ?)

(test (= ?a 12000))

=>

(printout t “Want to continue? [yes / no] : ”)

(retract ?fact)

(retract ?fact1)

(assert (con (read))))

(defrule ConError

?fact <- (con ?)

(and (not (con yes))

(not (con no)))

=>

(printout t crlf “Wrong input!!!” crlf)

(printout t “Want to continue? [yes / no] : ”)

(retract ?fact)

(assert (con (read))))

(defrule ContYes

?fact <- (go)

?fact1 <- (con yes)

?fact2 <- (choice ?)

=>

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(assert (go)))

(defrule ContNo

?fact <- (go)

?fact1 <- (con no)

?fact2 <- (start)

?fact3 <- (choice ?)

?fact4 <- (name ?)

=>

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(retract ?fact3)

(retract ?fact4)

(assert (start)))

(defrule dontCare

(choice 4)

=>

(printout t crlf “Please answering the following question.” crlf)

(assert (start1)))

(defrule bodytype

(start1)

=>

(printout t crlf "What body type of car you like?" crlf)

(printout t "1. Hatchback" crlf)

(printout t "2. Sedan" crlf)

(printout t "3. MPV" crlf)

(printout t "What is your choice (entering number): ")

(assert (bt (read))))

(defrule errorbodytype1

?fact <- (start1)

?fact1 <- (bt ?)

(and (and (not (bt 1))

(not (bt 2)))

(not (bt 3)))

=>

(printout t crlf “Wrong input!!” crlf)

(retract ?fact)

(retract ?fact1)

(assert (start1)))

(defrule hatchback

(bt 1)

=>

(printout t crlf "What brand of car you like?" crlf)

(printout t "1. Proton" crlf)

(printout t "2. Perodua" crlf)

(printout t “3. Back to choosing other body type” crlf)

(printout t "What is your choice (entering number): ")

(assert (b (read))))

(defrule errorhatchback

?fact <- (bt 1)

?fact1 <- (b ?)

(and (and (not (b 1))

(not (b 2)))

(not (b 3)))

=>

(printout t crlf “Wrong input!!” crlf)

(retract ?fact)

(retract ?fact1)

(assert (bt 1)))

(defrule sedan

(bt 2)

=>

(printout t crlf "What brand of car you like?" crlf)

(printout t "1. Toyota" crlf)

(printout t "2. Proton" crlf)

(printout t “3. Back to choosing other body type” crlf)

(printout t "What is your choice (entering number): ")

(assert (b (read))))

(defrule errorsedan

?fact <- (bt 2)

?fact1 <- (b ?)

(and (and (not (b 1))

(not (b 2)))

(not (b 3)))

=>

(printout t crlf “Wrong input!!” crlf)

(retract ?fact)

(retract ?fact1)

(assert (bt 2)))

(defrule mpv

(bt 3)

=>

(printout t crlf "What brand of car you like?" crlf)

(printout t "1. Toyota" crlf)

(printout t "2. Perodua" crlf)

(printout t “3. Back to choosing other body type” crlf)

(printout t "What is your choice (entering number): ")

(assert (b (read))))

(defrule errormpv

?fact <- (bt 3)

?fact1 <- (b ?)

(and (and (not (b 1))

(not (b 2)))

(not (b 3)))

=>

(printout t crlf “Wrong input!!” crlf)

(retract ?fact)

(retract ?fact1)

(assert (bt 3)))

(defrule brand1

(bt 1) (b 1)

=>

(printout t crlf "Please select the engine CC of the car?" crlf)

(printout t "1. 1.3" crlf)

(printout t "2. 1.5" crlf)

(printout t “3. Back to choosing other brand” crlf)

(printout t "What is your choice (entering number): ")

(assert (cc (read))))

(defrule errorbrand1

?fact <- (bt 1)

?fact1 <- (b 1)

?fact2 <- (cc ?)

(and (and (not (cc 1))

(not (cc 2)))

(not (cc 3)))

=>

(printout t crlf “Wrong input!!” crlf)

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(assert (bt 1))

(assert (b 1)))

(defrule brand2

(bt 1) (b 2)

=>

(printout t crlf "Please select the engine CC of the car?" crlf)

(printout t "1. 1.0" crlf)

(printout t "2. 1.5" crlf)

(printout t “3. Back to choosing other brand” crlf)

(printout t "What is your choice (entering number): ")

(assert (cc (read))))

(defrule errorbrand2

?fact <- (bt 1)

?fact1 <- (b 2)

?fact2 <- (cc ?)

(and (and (not (cc 1))

(not (cc 2)))

(not (cc 3)))

=>

(printout t crlf “Wrong input!!” crlf)

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(assert (bt 1))

(assert (b 2)))

(defrule brand3

(bt 2) (b 1)

=>

(printout t crlf "Please select the engine CC of the car?" crlf)

(printout t "1. 2.0" crlf)

(printout t "2. 2.5" crlf)

(printout t “3. Back to choosing other brand” crlf)

(printout t "What is your choice (entering number): ")

(assert (cc (read))))

(defrule errorbrand3

?fact <- (bt 2)

?fact1 <- (b 1)

?fact2 <- (cc ?)

(and (and (not (cc 1))

(not (cc 2)))

(not (cc 3)))

=>

(printout t crlf “Wrong input!!” crlf)

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(assert (bt 2))

(assert (b 1)))

(defrule brand4

(bt 2) (b 2)

=>

(printout t crlf "Please select the engine CC of the car?" crlf)

(printout t "1. 1.3" crlf)

(printout t "2. 1.7" crlf)

(printout t “3. Back to choosing other brand” crlf)

(printout t "What is your choice (entering number): ")

(assert (cc (read))))

(defrule errorbrand4

?fact <- (bt 2)

?fact1 <- (b 2)

?fact2 <- (cc ?)

(and (and (not (cc 1))

(not (cc 2)))

(not (cc 3)))

=>

(printout t crlf “Wrong input!!” crlf)

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(assert (bt 2))

(assert (b 2)))

(defrule brand5

(bt 3) (b 1)

=>

(printout t crlf "Please select the engine CC of the car?" crlf)

(printout t "1. 1.5" crlf)

(printout t "2. 3.5" crlf)

(printout t “3. Back to choosing other brand” crlf)

(printout t "What is your choice (entering number): ")

(assert (cc (read))))

(defrule errorbrand5

?fact <- (bt 3)

?fact1 <- (b 1)

?fact2 <- (cc ?)

(and (and (not (cc 1))

(not (cc 2)))

(not (cc 3)))

=>

(printout t crlf “Wrong input!!” crlf)

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(assert (bt 3))

(assert (b 1)))

(defrule brand6

(bt 3) (b 2)

=>

(printout t crlf "Please select the engine CC of the car?" crlf)

(printout t "1. 1.3" crlf)

(printout t "2. 1.5" crlf)

(printout t “3. Back to choosing other brand” crlf)

(printout t "What is your choice (entering number): ")

(assert (cc (read))))

(defrule errorbrand6

?fact <- (bt 3)

?fact1 <- (b 2)

?fact2 <- (cc ?)

(and (and (not (cc 1))

(not (cc 2)))

(not (cc 3)))

=>

(printout t crlf “Wrong input!!” crlf)

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(assert (bt 3))

(assert (b 2)))

(defrule sug1

(bt 1) (b 1) (cc 1) (name ?a)

=>

(printout t crlf ?a ", we suggest you to buy Iriz which achieve your requirement" crlf)

(printout t “The price is about RM 43000” crlf)

(printout t “Thank you” crlf)

(assert (continue)))

(defrule sug2

(bt 1) (b 1) (cc 2) (name ?a)

=>

(printout t crlf ?a ", we suggest you to buy Suprima S which achieve your requirement" crlf)

(printout t “The price is about RM 77000” crlf)

(printout t “Thank you” crlf)

(assert (continue)))

(defrule sug3

(bt 1) (b 2) (cc 1) (name ?a)

=>

(printout t crlf ?a ", we suggest you to buy Axia SE which achieve your requirement" crlf)

(printout t “The price is about RM 40000” crlf)

(printout t “Thank you” crlf)

(assert (continue)))

(defrule sug4

(bt 1) (b 2) (cc 2) (name ?a)

=>

(printout t crlf ?a ", we suggest you to buy Myvi SE which achieve your requirement" crlf)

(printout t “The price is about RM 53000” crlf)

(printout t “Thank you” crlf)

(assert (continue)))

(defrule sug5

(bt 2) (b 1) (cc 1) (name ?a)

=>

(printout t crlf ?a ", we suggest you to buy Carrola Altis which achieve your requirement" crlf)

(printout t “The price is about RM 110000” crlf)

(printout t “Thank you” crlf)

(assert (continue)))

(defrule sug6

(bt 2) (b 1) (cc 2) (name ?a)

=>

(printout t crlf ?a ", we suggest you to buy Camry which achieve your requirement" crlf)

(printout t “The price is about RM 145000” crlf)

(printout t “Thank you” crlf)

(assert (continue)))

(defrule sug7

(bt 2) (b 2) (cc 1) (name ?a)

=>

(printout t crlf ?a ", we suggest you to buy Saga FLX which achieve your requirement" crlf)

(printout t “The price is about RM 33000” crlf)

(printout t “Thank you” crlf)

(assert (continue)))

(defrule sug8

(bt 2) (b 2) (cc 2) (name ?a)

=>

(printout t crlf ?a ", we suggest you to buy Inspira which achieve your requirement" crlf)

(printout t “The price is about RM 80000” crlf)

(printout t “Thank you” crlf)

(assert (continue)))

(defrule sug9

(bt 3) (b 1) (cc 1) (name ?a)

=>

(printout t crlf ?a ", we suggest you to buy Avanza which achieve your requirement" crlf)

(printout t “The price is about RM 63000” crlf)

(printout t “Thank you” crlf)

(assert (continue)))

(defrule sug10

(bt 3) (b 1) (cc 2) (name ?a)

=>

(printout t crlf ?a ", we suggest you to buy Alphard which achieve your requirement" crlf)

(printout t “The price is about RM 324000” crlf)

(printout t “Thank you” crlf)

(assert (continue)))

(defrule sug11

(bt 3) (b 2) (cc 1) (name ?a)

=>

(printout t crlf ?a ", we suggest you to buy Rusa EX which achieve your requirement" crlf)

(printout t “The price is about RM 12000” crlf)

(printout t “Thank you” crlf)

(assert (continue)))

(defrule sug12

(bt 3) (b 2) (cc 2) (name ?a)

=>

(printout t crlf ?a ", we suggest you to buy Alza which achieve your requirement" crlf)

(printout t “The price is about RM 64000” crlf)

(printout t “Thank you” crlf)

(assert (continue)))

(defrule cont

?fact1 <- (start1)

?fact2 <- (bt ?)

?fact3 <- (b ?)

?fact4 <- (cc ?)

?fact5 <- (continue)

=>

(printout t crlf “The car is not suitable for you or you want to check the other car” crlf)

(printout t “yes for continue, no for finish : “ )

(assert (cont (read)))

(retract ?fact1)

(retract ?fact2)

(retract ?fact3)

(retract ?fact4)

(retract ?fact5))

(defrule errorcont

?fact <- (cont ?)

(and (not (cont yes))

(not (cont no)))

=>

(printout t crlf “Wrong input, please type yes or no only!!” crlf)

(printout t crlf “The car is not suitable for you or you want to check the other car” crlf)

(printout t “yes for continue, no for finish : “ )

(retract ?fact)

(assert (cont (read))))

(defrule cont1

?fact <- (cont yes)

?fact1 <- (go)

?fact2 <- (choice ?)

=>

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(assert (go)))

(defrule cont2

?fact <- (start)

?fact1 <- (cont no)

?fact2 <- (name ?)

?fact3 <- (go)

?fact4 <- (choice ?)

=>

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(retract ?fact3)

(retract ?fact4)

(assert (start)))

(defrule back1

?fact <- (start1)

?fact1 <- (bt ?)

?fact2 <- (b 3)

=>

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(assert (start1)))

(defrule back2

?fact <- (bt 1)

?fact1 <- (b ?)

?fact2 <- (cc 3)

=>

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(assert (bt 1)))

(defrule back3

?fact <- (bt 2)

?fact1 <- (b ?)

?fact2 <- (cc 3)

=>

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(assert (bt 2)))

(defrule back4

?fact <- (bt 3)

?fact1 <- (b ?)

?fact2 <- (cc 3)

=>

(retract ?fact)

(retract ?fact1)

(retract ?fact2)

(assert (bt 3)))