**Experiment - 9**

**Aim:**

**WAP for studying usage of 1) compound object and 2) list in prolog.**

**Procedure:**

1. **Write a program to maintain inventory items using a compound object. The format of compound object should be (item type, item (no, description, qty, cost)) Item-type can be fg-finish good, sf-semi finish good, rm-raw material,**

**Write appropriate predicates to**

1. **Accept from user the details of atleast 10 such objects**
2. **Display the details of objects entered by user.**

**Solution:-**

Prolog Code:

domains

ino,qty,cost=integer

desc,itype=string

items=item(ino,desc,qty,cost)

predicates

go

inven(itype,items)

search

enter(integer)

clauses

inven("fg",item(1,"keybord",10,2000)).

inven("rm",item(2,"plasticbox",10,300)).

inven("sf",item(3,"keypad",10,500)).

inven("fgi",item(4,"mouse",10,900)).

inven("rms",item(5,"plasticbag",10,500)).

inven("ssf",item(6,"remote",10,200)).

inven("fp",item(7,"speaker",10,5000)).

inven("lg",item(8,"refrigrator",10,9000)).

inven("lg",item(9,"oven",10,8000)).

inven("rel",item(10,"mobile",10,9000)).

go:-

enter(10),nl,

search.

enter(0):-write("All data is entered.").

enter(N):-

write("enter the ten objects data:"),nl,

write("enter item type: "),readln(T),

write("enter item number: "),readint(NO),

write("enter item description: "),readln(D),

write("enter item quantity: "),readint(Q),

write("enter item cost: "),readint(C),

inven(T,item(NO,D,Q,C)),

N=N-1,enter(N).

search:-

write("The details of objects entered by user is:"),nl,

write(" item type"),

write(" item number"),

write(" item description"),

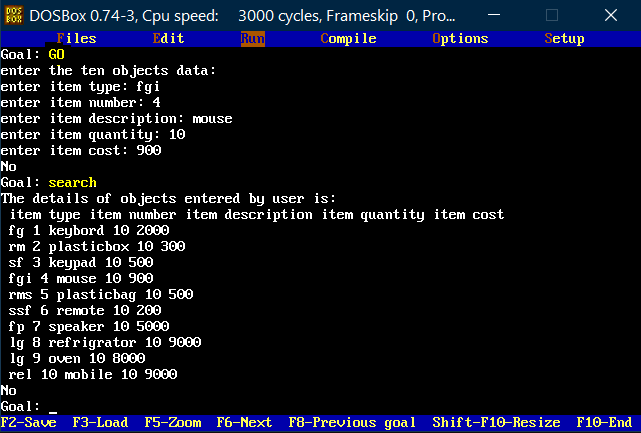
write(" item quantity"),

write(" item cost"),nl,

!,inven(T,item(NO,D,Q,C)),

write(" ",T," ",NO," ",D," ",Q," ",C),nl,fail.

Output Screenshots:



1. **Write a predicate which accepts integer number as an input and displays its square .It should also find its positive square root value ,if its sqrt is integer, otherwise display ‘NA’ .Use arithmetic operators /in-built conversion predicates to achieve this.**

**Solution:-**

Prolog Code:

domains

list=integer\*

database

odd(integer)

even(integer)

predicates

finde(list)

check(integer,integer)

outpute

outputo

clauses

finde([]).

finde([X|Tail]):-

Y=X mod 2,

check(X,Y),

finde(Tail).

check(X,0):-write("Even no:",X),asserta(even(X)),save("xyz.txt"),nl.

check(X,1):-write("Odd no:",X),asserta(odd(X)),save("xyz.txt"),nl.

outpute:-write("Evennumbers: "),consult("xyz.txt"),!,even(X),write(X,","),fail.

outputo:-write("Oddnumbers: "),consult("xyz.txt"),!,odd(X),write(X,","),fail.

Output Screenshots:





