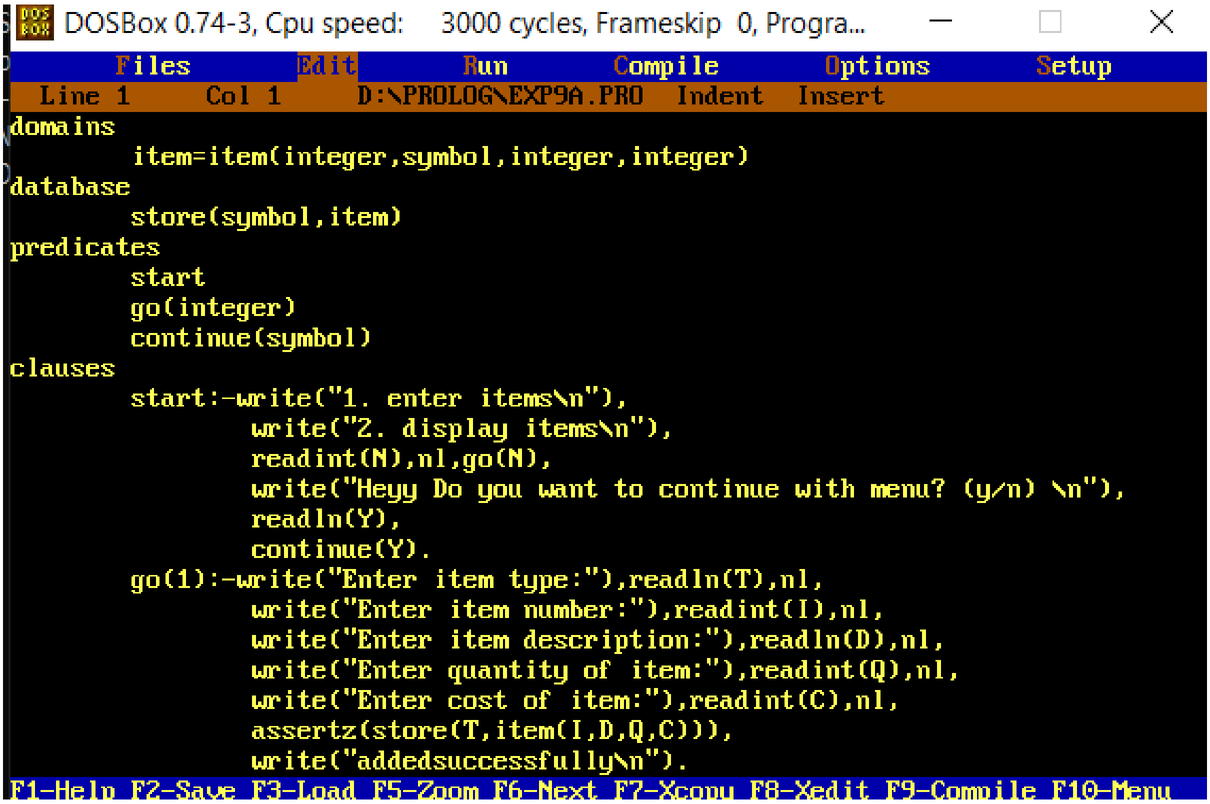
Lab-9 WAP to study Using compound object and lists in prolog.

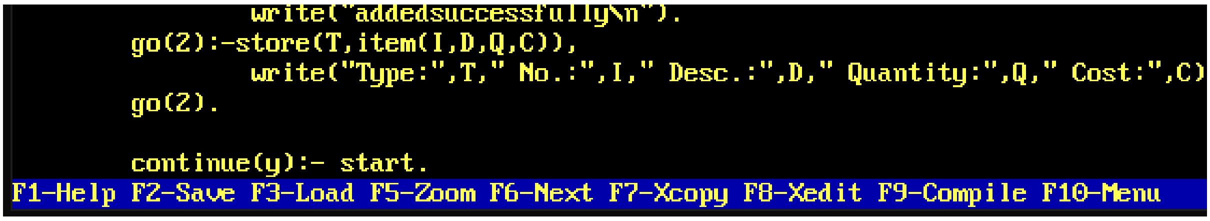
Procedure:-

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
   i) Accept from user the details of atleast 10 such objects
   ii) Display the details of objects entered by user.

Code:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Files
Edit
Run
Compile
Options
Setup
Line 1 Col 1 D:\PROLOG\EXP9A.PRO Indent Insert
F1-Help F2-Save F3-Load F5-Zoom F6-Next F7-Xcopy F8-Xedit F9-Compile F10-Menu

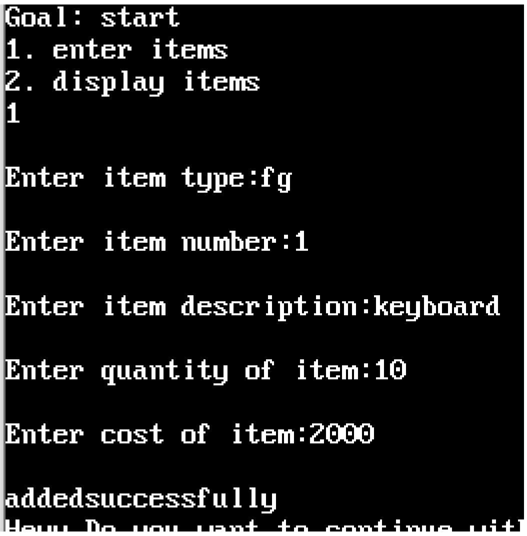
```
domains
        item=item(integer,symbol,integer,integer)
database
        store(symbol,item)
predicates
        start
        go(integer)
        continue(symbol)
clauses
        start:-write("1. enter items\n"),
                write("2. display items\n"),
                readint(N),nl,go(N),
                write("Heyy Do you want to continue with menu? (y/n) \n"),
                readln(Y),
                continue(Y).
        go(1):-write("Enter item type:"),readln(T),nl,
                write("Enter item number:"),readint(I),nl,
                write("Enter item description:"),readln(D),nl,
                write("Enter quantity of item:"),readint(Q),nl,
                write("Enter cost of item:"),readint(C),nl,
                assertz(store(T,item(I,D,Q,C))),
                write("addedsuccessfully\n").
```


F1-Help F2-Save F3-Load F5-Zoom F6-Next F7-Xcopy F8-Xedit F9-Compile F10-Menu

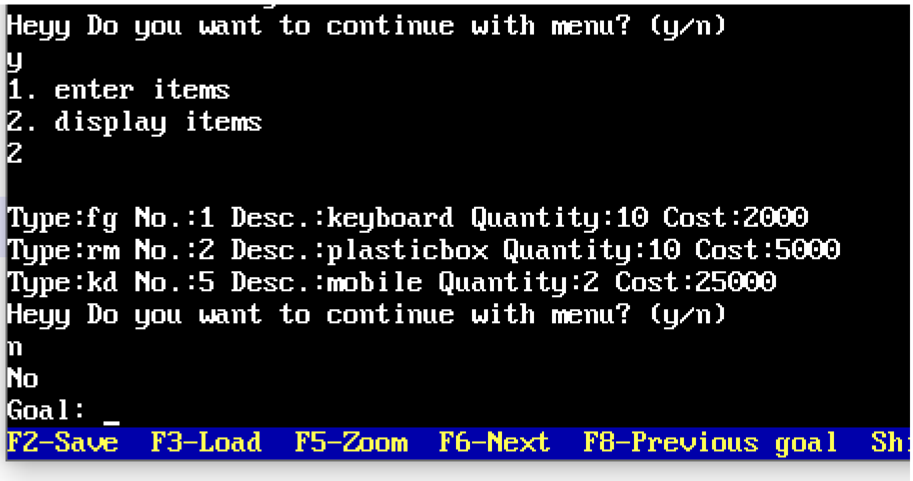
```
                write("addedsuccessfully\n").
        go(2):-store(T,item(I,D,Q,C)),
                write("Type:",T," No.:",I," Desc.:",D," Quantity:",Q," Cost:",C)
        go(2).

        continue(y):- start.
```

```
domains
        item=item(integer,symbol,integer,integer)
database
        store(symbol,item)
predicates
        start
        go(integer)
        continue(symbol)
clauses
        start:-write("1. enter items\n"),
                write("2. display items\n"),
                readint(N),nl,go(N),
                write("Heyy Do you want to continue with menu? (y/n) \n"),
                readln(Y),
                continue(Y).
        go(1):-write("Enter item type:"),readln(T),nl,
                write("Enter item number:"),readint(I),nl,
                write("Enter item description:"),readln(D),nl,
                write("Enter quantity of item:"),readint(Q),nl,
                write("Enter cost of item:"),readint(C),nl,
                assertz(store(T,item(I,D,Q,C))),
                write("addedsuccessfully\n").
        go(2):-store(T,item(I,D,Q,C)),
                write("Type:",T," No.:",I," Desc.:",D," Quantity:",Q," Cost:",C),nl,fail.
        go(2).

        continue(y):- start.
```

Output:



```
Goal: start
1. enter items
2. display items
1

Enter item type:fg

Enter item number:1

Enter item description:keyboard

Enter quantity of item:10

Enter cost of item:2000

addedsuccessfully
Heyy Do you want to continue with
```



```
Heyy Do you want to continue with menu? (y/n)
y
1. enter items
2. display items
2

Type:fg No.:1 Desc.:keyboard Quantity:10 Cost:2000
Type:rm No.:2 Desc.:plasticbox Quantity:10 Cost:5000
Type:kd No.:5 Desc.:mobile Quantity:2 Cost:25000
Heyy Do you want to continue with menu? (y/n)
n
No
Goal: _
F2-Save  F3-Load  F5-Zoom  F6-Next  F8-Previous goal  Sh
```

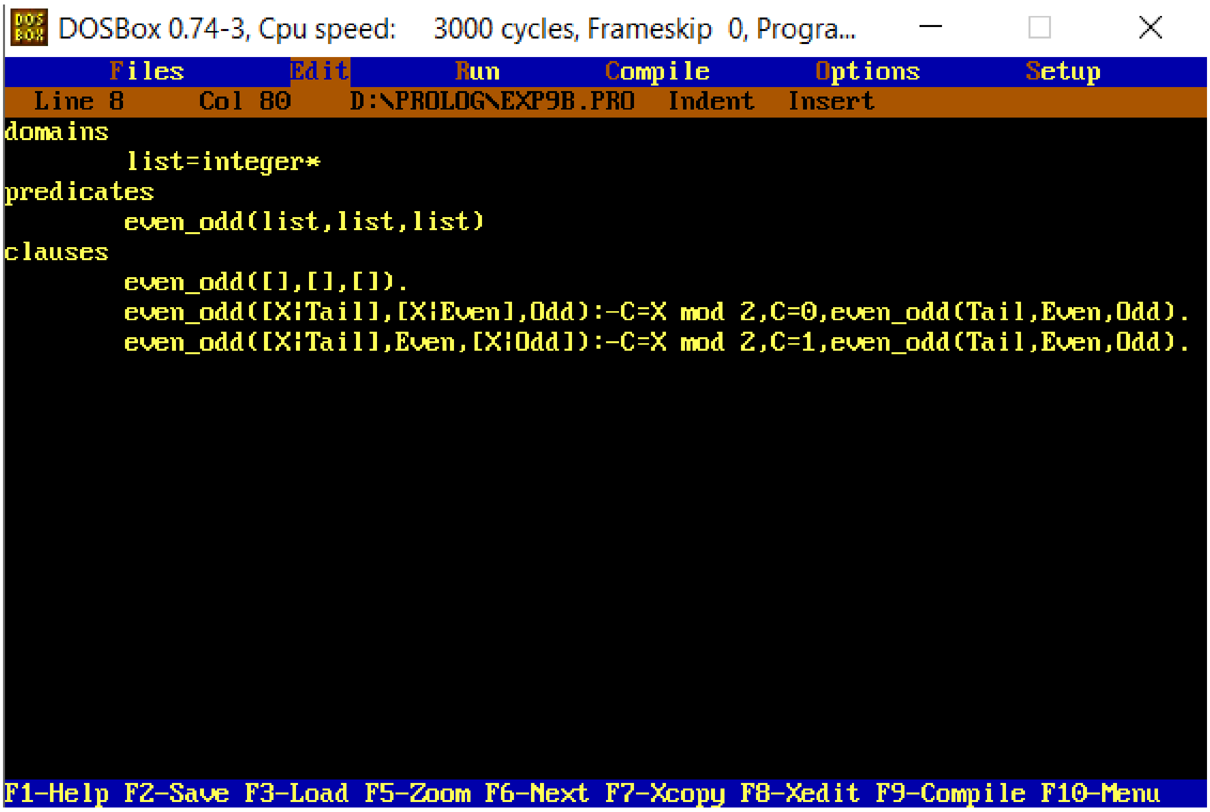
Procedure:-

2) Find and display odd and even numbers from a given input list of integers Example:- Output:- Enter list of 10 integer numbers 1 2 3 4 5 6 7 8 9 10

   Even numbers -> 2,4,6,8,10

   Odd numbers-> 1,3,5,7,9

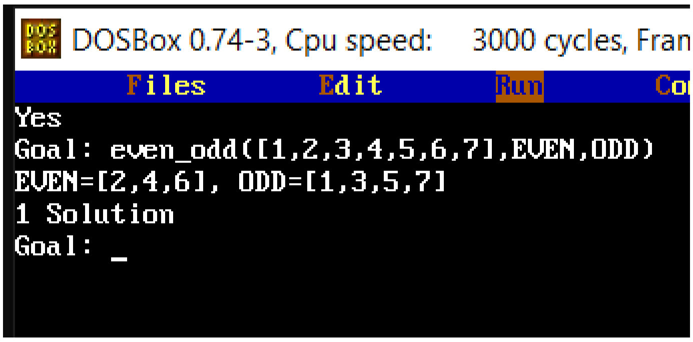
Code:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Files
Edit
Run
Compile
Options
Setup
Line 8
Col 80
D:\PROLOG\EXP9B.PRO
Indent
Insert
F1-Help F2-Save F3-Load F5-Zoom F6-Next F7-Xcopy F8-Xedit F9-Compile F10-Menu

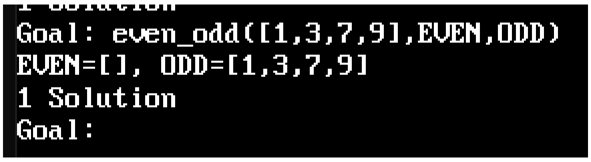
```
domains
        list=integer*
predicates
        even_odd(list,list,list)
clauses
        even_odd([],[],[]).
        even_odd([X|Tail],[X|Even],Odd):-C=X mod 2,C=0,even_odd(Tail,Even,Odd).
        even_odd([X|Tail],Even,[X|Odd]):-C=X mod 2,C=1,even_odd(Tail,Even,Odd).
```

```
domains
        list=integer*
predicates
        even_odd(list,list,list)
clauses
        even_odd([],[],[]).
        even_odd([X|Tail],[X|Even],Odd):-C=Xmod2,C=0,even_odd(Tail,Even,Odd).
        even_odd([X|Tail],Even,[X|Odd]):-C=Xmod2,C=1,even_odd(Tail,Even,Odd).
```

Output:



```
Yes
Goal: even_odd([1,2,3,4,5,6,7],EVEN,ODD)
EVEN=[2,4,6], ODD=[1,3,5,7]
1 Solution
Goal: _
```



```
1 Solution
Goal: even_odd([1,3,7,9],EVEN,ODD)
EVEN=[], ODD=[1,3,7,9]
1 Solution
Goal:
```