

Porofoliu FOX

1. Să se realizeze o aplicație pentru gestionarea unor filme. Pentru fiecare film se vor memora: numele filmului, categoria, actor principal, încasări. Aplicația va permite introducerea de noi filme, afișarea filmelor dintr-o anumită categorie, căutarea unui actor și afișarea filmelor în care joacă. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```
CREATE TABLE 1.dbf (nume C(15),  
categorie C(15) , actor C(15) ,  
incasari N(10) )
```

```
input "Numarul de filme:" to n
```

```
FOR i=1 TO n
```

```
APPEND BLANK
```

```
accept "Numele filmului:" to a
```

```
accept "Categoria:" to b
```

```
accept "Actor principal" to c
```

```
input "Incasari:" to d
```

```
replace nume WITH a, categorie  
WITH b, actor WITH c, incasari  
WITH d
```

```
ENDFOR
```

```
BROWSE
```

```
USE 1.dbf
```

```
*0/Introd campuri noi
```

```
accept "Daca doriti sa adaugati  
campuri noi scrieti 'DA' " to x
```

```
IF UPPER(ALLTRIM(x))="DA"
```

```
MODIFY STRUCTURE
```

```
APPEND
```

```
ENDIF
```

```
*a)afi?area filmelor dintr-o  
anumit? categorie
```

```
accept"dati categoria" to x
```

```
LIST FIELDS FOR categorie=x
```

```
*b)cautarea unui actor ?i afi?  
area filmelor în care joac?
```

```
accept"dati numele actorului" to  
y
```

```
LIST FIELDS nume FOR actor=y
```

2. Să se realizeze o aplicație pentru gestionarea unui magazin de telefoane mobile. Se vor memora: marca telefonului, marca, an apariție, preț. Aplicația va permite introducerea de noi telefoane, căutarea și afișarea telefoanelor dintr-o anumită marcă, afișarea prețului maxim al unui telefon și afișarea acelor telefoane cu preț maxim. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```
CREATE TABLE 2.dbf (marcatel  
C(10),marca C(10),an N(4),pret  
N(10))
```

```

input "Numarul de telefoane:" to n
FOR i=1 TO n

APPEND BLANK

accept "Marca telefonului:" to a
accept "Marca:" to b
input "An:" to c
input "Pret:" to d

replace marcatel WITH a,marca
WITH b,an WITH c, pret WITH d

ENDFOR

BROWSE
USE 2.dbf

*0/Introd campuri noi
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x
IF UPPER(ALLTRIM(x))="DA"
MODIFY STRUCTURE
APPEND
ENDIF

*a)c?utarea ?i afi?area
telefoanelor dintr-o anumit?
marc?

accept"dati marca" to x
LIST FIELDS FOR marca=x

*b)afi?area pre?ului maxim al
unui telefon ?i afi?area acelor
telefoane cu pre? maxim

```

```

GO top
pretm=pret
GO 2
SCAN
if (pretm<pret
pretm=prêt
endif
endscan
?"pretul maxim este",pretm
LIST FIELDS FOR pret=pretm

```

3 Să se realizeze o aplicație pentru evidența elevilor. Se vor memora: numele, cnp, clasa, profil. Aplicația va permite introducerea de noi elevi, având grijă să nu se introducă un cnp care deja apare. De asemenea, aplicația va permite căutarea unui elev și afișarea datelor sale și afișarea elevilor de la o anumită clasă. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```

CREATE TABLE 3.dbf (numele C(20),
cnp N(14), clasa N(2),profil
C(10))

input "Numarul de elevi:" to n
FOR i=1 TO n

APPEND BLANK

accept "Numele:" to a
input "CNP:" to b
input "Clasa:" to c

```

```

accept "Profilul:" to d

replace numele WITH a,cnp WITH
b,clasa WITH c,profil WITH d

ENDFOR

BROWSE

USE 3.dbf

*0/Introd campuri noi

accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x

IF UPPER(ALLTRIM(x))="DA"

MODIFY STRUCTURE

APPEND

ENDIF

*a)aplica?ia va permite c?utarea
unui elev ?i afi?area datelor
sale

accept "dati numele" to x

LIST FIELDS FOR numele=x

*b)afi?area elevilor de la o
anumit? clas?.

accept "dati clasa" to y

LIST FIELDS FOR clasa=y

```

4. Să se realizeze o aplicație pentru a memora date despre țările lumii. Se vor memora: numele țării, suprafața, număr locuitori, continentul. Aplicația va permite introducerea de noi țări, având grijă să nu se introducă o țară care deja apare, căutarea și afișarea alfabetică a țărilor de pe un anumit continent, afișarea țării cu suprafața maximă, afișarea țării cu populație maximă. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```

CREATE TABLE 4.dbf(tara
C(10),suprafata N(10),locuitori
N(13),continent C(15))

```

```

input "Nr. de tari" to n

FOR i=1 TO n

```

```

APPEND blank

```

```

accept "Tara" to a

input "Suprafata" to b

input "Nr. locuitori" to c

accept "Continent" to d

replace tara WITH a,suprafata
WITH b,locuitori WITH
c,continent WITH d

```

```

ENDFOR

```

```

BROWSE

```

```

USE 4.dbf

```

```

*0/Introd campuri noi

```

```

accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x

IF UPPER(ALLTRIM(x))="DA"

```

```

MODIFY STRUCTURE

APPEND

ENDIF

*a)c?utarea ?i afi?area
alfabetic? a ??rilor de pe un
anumit continent

SORT ON tara TO 4_2.dbf

USE 4_2.dbf

LIST

accept "dati continentul" to
conti

LIST FIELDS tara FOR
continent=conti

*b)afi?area ??rii cu suprafata?
maxim?

USE 4.dbf

maxs=0

GO top

SCAN

if(suprafata>max)

maxs=suprafata

ENDIF

ENDSCAN

LIST tara for suprafata=maxs

*c)afi?area ??rii cu popula?ie
maxim?

maxs=0

GO top

SCAN

if(locuitori>maxp)

maxp=locuitori

ENDIF

```

```

ENDSCAN

LIST tara for locuitori=maxp

5. Să se realizeze o aplicație pentru evidența
unor angajați. Se vor memora: numele
angajatului, secția, salariul, codul
angajatului. Aplicația va permite
introducerea de noi angajați, având grijă să
nu se introducă un angajat având codul deja
existent, căutarea și afișarea alfabetică a
angajaților de la o anumită secție, afișarea
sumei totale a salariilor, afișarea angajaților
cu salariul mai mare decât o anumită
valoare. Pot fi adăugate și alte câmpuri în
afara celor specificate. Este obligatoriu să
apară minimum 10 înregistrări pentru a se
putea testa aplicația.

*CREATE TABLE 5.dbf ( nume C(25)
, sectia C(25) , salariu N(10) ,
cod N(13))

*input "Dati nr de angajati: "
to n

*FOR i=1 TO n

*APPEND BLANK

*accept "Dati numele: " to a
*accept "Dati sectia: " to b
*input "Dati salariul: " to c
*input "Dati codul: " to d

*replace nume WITH a , sectia
WITH b , salariu WITH c , cod
WITH d

```

```

*ENDFOR

USE 5.dbf

accept"Dati o sectie" to sec

SORT on nume/A TO 5_2.dbf FOR
sectia=sec

USE 5_2.dbf

LIST FIELDS
nume,sectia,salariu,cod

USE 5.dbf

sm=0

suma=0

GO TOP

SCAN

suma=suma+salariu

IF sm<salariu

sm=salariu

ENDIF

ENDSCAN

GO top

SCAN

IF sm=salariu

?nume,sectia,cod,salariu

?

ENDIF

ENDSCAN

?"Suma salariilor este",suma

input"Afiseaza angajatii cu
salariul mai mare de " to maxim

LIST FIELDS nume FOR
salariu>maxim

```

6. Să se realizeze o aplicație pentru evidența unor abonați la o companie de telefonie. Se vor memora: numele abonatului, numar de telefon, localitate, pret abonament. Aplicația va permite introducerea de noi abonați, având grijă să nu se introducă un abonat având numărul de telefon deja existent, căutarea și afișarea alfabetică a abonaților dintr-o anumită localitate, afișarea numărului total de abonați, afișarea abonaților care încep cu o anumită literă. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```

*CREATE TABLE 6.dbf ( nume C(25)
, nrtel N(14) , localitate C(25)
, pret N(10))

```

```

*input "Dati numarul de abonati:
" to n

```

```

*FOR i=1 TO n

```

```

*APPEND BLANK

```

```

*accept "Dati numele: " to a

```

```

*input "Dati nr telefon: " to b

```

```

*accept "Dati localitatea: " to
c

```

```

*input "Dati pretul: " to d

```

```

*replace nume WITH a , nrtel
WITH b , localitate WITH c ,
pret WITH d

```

```

*ENDFOR

```

```

USE 6.dbf

accept"Daca doriti introducerea
de noi valori, scrieti DA" to x

IF UPPER(ALLTRIM(x))="DA"

MODIFY STRUCTURE

APPEND

ENDIF

accept"Afisarea abonatilor din
localitatea" to loc

SORT ON nume TO 6_2.dbf FOR
UPPER(ALLTRIM(localitate))=UPPER
(ALLTRIM(loc))

USE 6_2.dbf

LIST FIELDS nume

USE 6.dbf

n=reccount()

?"Numarul total de abonati
este",n

accept"Afiseaza abonatii care
incep cu litera" to ch

LIST FIELDS nume FOR
UPPER(ALLTRIM(LEFT(nume,1)))=UPP
ER(ALLTRIM(ch))

```

7. Să se realizeze o aplicație pentru un concurs de informatică. Se vor memora: numele elevului, codul elevului, liceul, punctajul. Aplicația va permite introducerea de noi elevi, având grijă să nu se introducă un elev având codul deja existent, afișarea alfabetică a elevilor și afișarea descrescătoare după punctaj, media punctajelor elevilor de la un anumit liceu. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```

*CREATE TABLE 7.dbf ( nume C(25)
, cod C(10) , liceu C(25) ,
punctaj F(3,2))

```

```

*input "Dati nr elevi: " to n

```

```

*FOR i=1 TO n

```

```

*APPEND BLANK

```

```

*accept "Dati numele: " to a

```

```

*accept "Dati codul: " to b

```

```

*accept "Dati liceul: " to c

```

```

*input "Dati punctajul: " to d

```

```

*replace nume WITH a , cod WITH
b , liceu WITH c , punctaj WITH
d

```

```

*ENDFOR

```

```

USE 7.dbf

```

```

*Introducere de noi val

accept"Daca doriti introducerea
de noi campuri,scrieti DA" to x

IF UPPER(ALLTRIM(x))="DA"
MODIFY STRUCTURE
APPEND
ENDIF

*a)Afisarea elevilor ordonati
alfabetic

MESSAGEBOX("Afisarea elevilor
ordonati alfabetic")

SORT ON nume/A TO 7_a.dbf
USE 7_a.dbf
LIST FIELDS nume
USE 7.dbf

*b)Afisarea elevilor
descrescator dupa punctaj

MESSAGEBOX("Afisarea elevilor
ordonati descrescator dupa
punctaj")

SORT ON punctaj/D TO 7_b.dbf
USE 7_b.dbf
LIST fields nume,punctaj
USE 7.dbf

*c)Media punctajelor de la un
anumit liceu

accept"Afiseaza media
punctajelor de la liceul"to l
n=0

```

```

s=0
GO top
SCAN
IF
UPPER(ALLTRIM(liceu))=UPPER(ALLT
RIM(l))
n=n+1
s=s+punctaj
ENDIF
ENDSCAN
m=s/n
?m

```

8. Să se realizeze o aplicație pentru a ține evidența unor albume de muzică. Se vor memora: titlul albumului, autorul, anul apariției, preț. Aplicația va permite introducerea de noi albume, afișarea albumelor unui anumit autor, descrescător după anul apariției. De asemenea să se afișeze anul în care au apărut cele mai multe albume, precum și acele albume care au apărut în acel an. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```

*CREATE TABLE 8.dbf ( titlu
C(25) , autor C(25) , anap
N(4) , pret F(10,2))

```

```

*input "Dati nr de albume: " to
n

```

```

*FOR i=1 TO n

```

```

*APPEND BLANK

```

```

*accept "Dati titlul: " to a
*accept "Dati autorul: " to b
*input "Dati anul aparitiei: "
to c
*input "Dati pretul: " to d

*replace titlu WITH a , autor
WITH b , anap WITH c , pret WITH
d

*ENDFOR

USE 8.dbf

*0/Introd val noi
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x

IF UPPER(ALLTRIM(x))="DA"
MODIFY STRUCTURE
APPEND
ENDIF

*a/Afisare albume unui autor
descrescator in functie de anul
de aparitie\
accept "Dati autorul: " to
autor_2
SORT ON anap TO 8_2.dbf
USE 8_2.dbf
GO TOP

LIST FIELDS titlu,autor,anap FOR
autor=autor_2

```

```

*b/Anul in care au aparut cele
mai multe albume

```

```

USE 8.dbf
n=RECCOUNT()
maxim=0
maxan=0

FOR i=1 TO n
GO i
anul=anap
nr=0
GO TOP
SCAN
IF anap=anul
nr=nr+1
ENDIF
ENDSCAN
IF nr>maxim
maxim=nr
maxan=anul
ENDIF
ENDFOR

?"Anul in care au aparut cele
mai multe albume este: ",maxan

?"Albumele care au aparut in
acest an sunt: "

LIST FIELDS titlu,autor FOR
anap=maxan

```


9. Să se realizeze o aplicație pentru a ține evidența unor angajați. Se vor memora: numele angajatului, codul numeric personal, departamentul, salariul. Aplicația va permite introducerea de noi angajați, afișarea alfabetică a angajaților unui anumit departament. De asemenea să se afișeze salariul mediu, precum și angajații care sunt femei (au codul numeric personal care începe cu 2 sau cu 6). Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```
*CREATE TABLE 9.dbf ( nume C(15)
, cnp C(13) , departam C(15) ,
salariu N(10))
```

```
*input "Dati nr de angajati: "
to n
```

```
*FOR i=1 TO n
```

```
*APPEND BLANK
```

```
*accept "Dati numele: " to a
```

```
*accept "Dati CNP-ul: " to b
```

```
*accept "Dati departamentul: "
to c
```

```
*input "Dati salariul: " to d
```

```
*replace nume WITH a , cnp WITH
b , departam WITH c , salariu
WITH d
```

```
*ENDFOR
```

```
USE 9.dbf
```

```
accept"Daca doriti sa
introduceti noi valori, scrieti
DA" to x
```

```
IF UPPER(ALLTRIM(x))="DA"
```

```
MODIFY STRUCTURE
```

```
APPEND
```

```
ENDIF
```

```
*a)Afisarea in ordine alfabetica
a angajatilor unui anumit
departament
```

```
accept"Sa se afiseze in ordine
alfabetica angajatii
departamentului " to dep
```

```
SORT ON nume/A TO 9_a.dbf FOR
UPPER(ALLTRIM(departam))=UPPER(A
LLTRIM(dep))
```

```
USE 9_a.dbf
```

```
LIST FIELDS nume
```

```
USE 9.dbf
```

```
*b)Salariul mediu
```

```
n=RECCOUNT()
```

```
s=0
```

```
GO top
```

```
SCAN
```

```
s=s+salariu
```

```
ENDSCAN
```

```
m=s/n
```

```
? "Salariul mediu este",m
```

```
*c)Nr de angajati care sunt
femei
```

```

nrf=0

GO top

SCAN

IF LEFT(cnp,1)="2" OR
LEFT(cnp,1)="6"

nrf=nrf+1

ENDIF

ENDSCAN

?"Numarul de angajati care sunt
femei este",nrf

```

10. Să se realizeze o aplicație pentru a ține evidența unor cărți. Se vor memora: titlul cărții, autorul, editura, prețul. Aplicația va permite introducerea de noi cărți. De asemenea să se afișeze titlurile cărților scrise de un anumit autor, numărul cărților de la o anumită editură, precum și prețul maxim al unei cărți. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```

*CREATE TABLE 10.dbf ( titlu
C(15) , autor C(15) , editura
C(15) , pretul F(10,2))

*input "Dati numarul unor carti:
" to n

*FOR i=1 TO n

*APPEND BLANK

*accept "Dati titlul: " to a
*accept "Dati autorul: " to b
*accept "Dati editura: " to c

```

```

*input "Dati pretul: " to d

*replace titlu WITH a , autor
WITH b , editura WITH c , pretul
WITH d

*ENDFOR

USE 10.dbf

accept"Daca doriti introducerea
de noi carti,scrieti DA" to x

IF UPPER(ALLTRIM(x))="DA"

MODIFY STRUCTURE

APPEND

ENDIF

*a)Afisarea cartilor scrise de
un anumit autor

accept"Afiseaza cartile scrise
de " to aut

LIST FIELDS titlu FOR
UPPER(ALLTRIM(autor))=UPPER(ALLT
RIM(aut))

*b)Afiseaza numarul cartilor de
la o anumita editura si pretul
maxim al unei carti

accept"Afiseaza nr. cartilor de
la editura" to edi

n=0

pm=0

scan

IF pm<pretul

pm=pretul

```

```

ENDIF

IF
UPPER(ALLTRIM(editura))=UPPER(ALLTRIM(edi))

n=n+1

ENDIF

ENDSCAN

?n

?"Pretul maxim al unei carti
este",pm

```

11. Să se realizeze o aplicație pentru gestionarea camerelor unui hotel. Se vor memora: numărul camerei, numărul de paturi ai camerei, etajul, prețul, stare (liberă/ocupată). Aplicația va permite introducerea de noi camere (având grijă să nu se introducă un număr de cameră deja existent), afișarea camerelor libere, afișarea numărului de camere libere cu un număr specificat de paturi, precum și prețul minim al unei camere. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```

*CREATE TABLE 11.dbf ( nrcam
N(3),nrpat N(2),etajul
N(2),pretul N(4),stare C(10))

*input"Dati nr de camere" to n

*FOR i=1 TO n

*APPEND BLANK

*input "Dati numarul camerei" to
a

*input "Dati nr de paturi ai
camerei" to b

```

```

*input "Dati etajul camerei" to
c

*input "Dati pretul camerei" to
d

*accept "Dati starea camerei" to
e

```

```

*replace nrcam WITH a , nrpat
WITH b , etajul WITH c ,pretul
WITH d , stare WITH e

```

```

*ENDFOR

```

```

*BROWSE

```

```

USE 11.dbf

```

```

*Introducerea de noi campuri

```

```

accept "Daca doriti introducerea
de noi campuri, scrieti DA" to x

```

```

IF UPPER(ALLTRIM(x))="DA"

```

```

MODIFY STRUCTURE

```

```

APPEND

```

```

ENDIF

```

```

*a)Afisarea camerele libere

```

```

?"Afisarea camerelor libere"

```

```

LIST FIELDS

```

```

nrcam,nrpat,etajul,pretul FOR
UPPER(ALLTRIM(stare))="LIBERA"

```

```

*b)Afisarea numarului camerelor
libere cu un numar specificat de
paturi

```

```

input"Afisaza numarului
camerelor libere cu numar de
paturi " to y

n=0

GO TOP

SCAN

IF
UPPER(ALLTRIM(stare))="LIBERA"
AND nrpat=y

n=n+1

ENDIF

ENDSCAN

?n

*c)Pretul minim al unei camere

GO top

min=pretul

SCAN

IF pretul<min

min=pretul

ENDIF

ENDSCAN

?"Pretul minim al unei camere
este",min,"lei"

```

12. Să se realizeze o aplicație pentru evidența încasărilor și plăților. Se vor memora: operațiune (încasare sau plată), suma, număr document, denumirea firmei care plătește sau de la care se încasează. Aplicația va permite introducerea de noi operațiuni (având grijă să nu se introducă un număr de document deja existent), afișarea sumelor totale încasate și a celor plătite, afișarea numărului de încasări, afișarea

numărului de plăți. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```

CREATE TABLE 12.DBF (oper
C(10),suma N(10) ,nrdoc N(10)
,firma C(20))

input"Dati numarul de
operatiuni" to n

FOR i=0 TO n

APPEND BLANK

accept"Dati operatiunea
(incasare/plata)" to a

input"Dati suma" to b

input"Dati numarul documentului"
to c

accept"Dati numele firmei" to d

REPLACE oper WITH a,suma WITH
b,nrdoc WITH c,firma WITH d

ENDFOR

BROWSE

USE 12.dbf

*Introducerea de noi campuri

accept"Daca doriti introducerea
de noi campuri, scrieti DA"to x

IF UPPER(ALLTRIM(x))="DA"

MODIFY STRUCTURE

APPEND

ENDIF

*a)Afisarea sumelor totale
incasate si a celor platite

```

```

i=0
p=0
GO top
SCAN
IF
UPPER(ALLTRIM(oper))="INCASARE"
i=i+suma
ELSE
p=p+suma
ENDIF
ENDSCAN

?"Suma totala incasata",i,"lei"
?"Suma totala platita",p,"lei"

*b)Numarul de incasa si numarul
de plati

ni=0
np=0
GO top
SCAN
IF
UPPER(ALLTRIM(oper))="INCASARE"
ni=ni+1
ELSE
np=np+1
ENDIF
ENDSCAN

?"Numarul total de
incasa",i,"lei"
?"Numarul total de
platiti",p,"lei"

```

13. Să se realizeze o aplicație pentru gestionarea rezultatelor la proba scrisă a bacalaureatului. Se vor memora: numele elevului, cnp, nota la proba 1, nota la proba 2, nota la proba 3, media. Aplicația va permite introducerea de noi elevi (având grijă să nu se introducă un cnp deja existent), afișarea alfabetică a elevilor, calculul mediei tuturor elevilor în funcție de cele trei note, precum și afișarea descrescător după medie a primilor cinci elevi cu mediile cele mai mari. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```

*CREATE TABLE 13.dbf (nume
C(20), cnp N(20), nota1 N(4),
nota2 N(4), nota3 N(4), media
F(4,2))

*input "Dati nr de elemente" to
n
*FOR i=1 TO n
*APPEND BLANK
*accept "Dati numele" to a
*input "Dati cnp" to b
*input "Nota 1" to c
*input "Nota 2" to d
*input "Nota 3" to e

*replace nume WITH a, cnp WITH
b, nota1 WITH c, nota2 WITH d,
nota3 WITH e, media WITH
(nota1+nota2+nota3)/3
*ENDFOR

USE 13.dbf

```

*Introducerea de noi campuri

accept "Daca doriti introducerea
de noi campuri, scrieti DA" to x

IF UPPER(ALLTRIM(x))="DA"

MODIFY STRUCTURE

APPEND

ENDIF

?"Afisare elevi ordine
alfabetica"

*a/Afisare alfabetica

SORT ON nume TO 13_2.dbf

USE 13_2.dbf

GO TOP

LIST FIELDS nume

?"Afisare medii"

*b/Calcul medii

USE 13.dbf

GO TOP

SCAN

?nume," ",media

ENDSCAN

?"Afisare descr dupa medie 5
elevi"

*c/Afisare descr dupa medie 5
elevi

SORT ON media/d TO 13_3.dbf

USE 13_3.dbf

GO TOP

FOR i=1 TO 5

GO i

?nume," ", media

ENDFOR

14. Să se realizeze o aplicație pentru gestionarea cărților unei biblioteci. Se vor memora ISBN-ul, autorul, titlul, anul apariției, numărul de exemplare. Aplicația va permite adăugarea de noi cărți (având grijă să nu se introducă un ISBN deja existent), afișarea pe ecran a cărților care sunt într-un anumit număr volume, creșterea cu 1 a numărului de exemplare ale cărților unui anumit autor, precum și numărul total de cărți din bibliotecă. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

*CREATE TABLE 14.dbf (isbn
N(15), autor C(20), titlu C(20),
an N(4), nr N(4))

*Crearea tabelului

*input "Dati nr de carti" to n

*FOR i=1 TO n

*APPEND BLANK

*input "Dati ISBN-ul" to a

*accept "Dati autorul" to b

*accept "Dati titlul" to c

```

*input "Dati anul aparitiei" to
d

*input "Dati nr. de exemplare"
to e

*replace isbn WITH a, autor WITH
b, titlu WITH c, an WITH d, nr
WITH e

*ENDFOR

```

```

*BROWSE

```

```

BROWSE

```

```

*Introducere de noi valori

USE 14.dbf

accept "Daca doriti introducerea
de noi campuri, scrieti DA" to x

IF UPPER(ALLTRIM(x))="DA"

MODIFY STRUCTURE

APPEND

ENDIF

```

```

*a)Afisarea cartilor care sunt
intr un anumit numar de volume

```

```

input "Afiseaza cartile intr un
anumit nr. de volume" to ex

LIST FIELDS titlu,autor FOR
nr=ex

```

```

*b)Cresterea cu 1 a num?rului de
exemplare ale c?r?ilor unui
anumit autor

```

```

accept "Dati autorul pentru care
nr. de exemplare creste cu 1" to
aut

```

```

GO top

```

```

SCAN

```

```

IF autor=aut

```

```

replace nr WITH nr+1

```

```

ENDIF

```

```

ENDSCAN

```

```

LIST FIELDS titlu,nr FOR
autor=aut

```

```

*c)Numarul total de carti din
biblioteca

```

```

t=0

```

```

GO top

```

```

SCAN

```

```

t=t+nr

```

```

ENDSCAN

```

```

?"Numarul total de carti din
biliboteca",t

```

15. Să se realizeze o aplicație care să permită gestiunea unor produse. Pentru fiecare produs se vor memora: codul, denumirea, prețul și cantitatea. Aplicația va permite adăugarea de noi produse (având grijă să nu se introducă un cod deja existent), afișarea numărului de produse, afișarea produselor care au prețul mai mic decât o valoare dată, afișarea produsului având prețul maxim. Pot fi adăugate și alte câmpuri în afara celor specificate. Este obligatoriu să apară minimum 10 înregistrări pentru a se putea testa aplicația.

```
CREATE TABLE 15.dbf (codul N(8),
denumirea C(25), pretul N(4),
cantitate N(5))
```

```
input "Dati nr de valori" to n
```

```
FOR i=1 TO n
```

```
APPEND BLANK
```

```
input "Dati codul: " to a
accept "Dati denumirea: " to b
input "Dati pretul: " to c
input "Dati cantitatea: " to d
replace codul WITH a , denumirea
WITH b , pretul WITH c ,
cantitate WITH d
```

```
ENDFOR
```

```
USE 15.dbf
```

```
*0/Introd campuri noi
```

```
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x
```

```
IF UPPER(ALLTRIM(x))="DA"
```

```
MODIFY STRUCTURE
```

```
APPEND
```

```
ENDIF
```

```
*a/Afisare numarului de produse
```

```
?n
```

```
?
```

```
*b/Afisare produse care au pret
mai mic decat o valoarea data
```

```
input "Dati valoarea dorita " to
x
```

```
GO top
```

```
LIST FIELDS denumirea FOR
pretul<x
```

```
*c/Afisare produs cu pret maxim
```

```
pmax=0
```

```
GO top
```

```
SCAN
```

```
if (pretul>pmax)
```

```
pmax=pretul
```

```
ENDIF
```

```
ENDSCAN
```

```
LIST FIELDS denumirea FOR
pretul=pmax
```

16. Sa se realizeze o aplicatie care sa prelucreze date despre melodii. Pentru fiecare melodie se vor memora: numele, artistul, durata, genul melodiei. Aplicatia va permite adaugarea de noi melodii, afisarea numarului de melodii ale unui anumit artist, afisarea melodiilor care au durata mai mica decât o valoare data, afisarea artistului având cele mai multe melodii, afisarea numarului melodiilor dintr-un anumit gen. Pot fi adaugate si alte câmpuri în afara celor specificate. Este obligatoriu sa apara minimum 10 înregistrari pentru a se putea testa aplicatia.

```
CREATE TABLE 16.dbf (nume C(25),
artistul C(25), durata N(10) ,
genul C(25))
```



```

input "Dati nr de inregistrari: " to n

FOR i=1 TO n

APPEND BLANK

accept "Dati numele: " to a
accept "Dati artistul: " to b
input "Dati durata: " to c
accept "Dati genul muzical: " to d

replace nume WITH a , artistul
WITH b , durata WITH c , genul
WITH d

ENDFOR

USE 16.dbf

*0/Introd campuri noi
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x
IF UPPER(ALLTRIM(x))="DA"
MODIFY STRUCTURE
APPEND
ENDIF

*a/Numarului de melodii ale unui
anumit artist
accept "Dati numele artistului: " to x
GO TOP
LIST FIELDS nume,artistul FOR
artistul=x

*b/Afisarea melodiilor care au
durata mai mica decât o valoare
input "Dati valoarea: " to y
GO TOP
LIST FIELDS nume,artistul FOR
durata<y

*c/Afisarea artistului având
cele mai multe melodii
USE 16.dbf
MODIFY STRUCTURE
REPLACE ALL idd WITH 1
INDEX on artistul TAG comp

```

```

TOTAL ON artistul TO 16_2.dbf
FIELDS idd
USE 16_2.dbf
SORT ON idd/d TO 16_3.dbf
USE 16_3.dbf
GO top
?artistul
x=artistul
USE 16.dbf
LIST FIELDS nume FOR artistul=x

*d/Afisare nr melodii dintr-un
anumit gen
accept "Dati genul dorit: " to x
s=0
GO TOP
SCAN
IF genul=x
s=s+1
ENDIF
ENDSCAN
?s
?

```

17. Sa se realizeze o aplicatie care sa gestioneze materialele dintr-o magazie. Pentru fiecare material se vor memora: denumire, pret unitar, cantitate. Aplicatia va permite adaugarea de noi materiale, având grija sa nu se adauge un material deja existent, afisarea numarului de de materiale, afisarea materialelor care au cantitatea cuprinsa între 10 si 100, afisarea denumirii materialului având pretul minim, stergerea materialelor care au cantitatea egala cu 0. Pot fi adaugate si alte câmpuri în afara celor specificate. Este obligatoriu sa apara minimum 10 înregistrari pentru a se putea testa aplicatia.

```

CREATE TABLE 17.dbf (denumire
C(25), pretul F(5,2), cantitate
N(5))

```

```

input "Dati nr de valori" to n

FOR i=1 TO n

```

```

APPEND BLANK

accept "Dati denumirea: " to a
input "Dati pretul: " to b
input "Dati cantitatea: " to c

replace denumire WITH a , pretul
WITH b , cantitate WITH c

ENDFOR

USE 17.dbf

*0/Introd campuri noi
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x
IF UPPER(ALLTRIM(x))="DA"
MODIFY STRUCTURE
APPEND
ENDIF

*a/Afisare nr materiale
?n
?

*b/Afisare nr materiale care au
cantitate cuprinsa intre 10 si
100
s=0
GO top
SCAN
IF cantitate>10 AND
cantitate<100
s=s+1
ENDIF
ENDSCAN
?s
?

*c/Afisare denumire produs pret
minim
GO top
minimp=pretul
GO 2
SCAN
IF minimp<pretul
minimp=pretul
ENDIF
ENDSCAN
LIST FIELDS denumire FOR
pretul=minimp

```

```

*d/Stergerea materialelor care
au cantitatea egala cu 0
GO TOP
DELETE denumire for cantitate=0

```

18. Sa se realizeze o aplicatie care sa gestioneze excursiile organizate de o agen? ie. Pentru fiecare excursie organizata se vor memora: locatia excursiei, costul, numar persoane, numarul de zile. Aplicatia va permite adaugarea de noi excursii, afisarea excursiei având costul minim, afisarea excursiilor catre o anumita locatie, ordonate crescator dupa numarul de zile, afisarea numarului total de persoane care merg în toate excursiile, scaderea cu 10 lei a pretului excursiilor catre o anumita locatie. Pot fi adaugate si alte

câmpuri în afara celor specificate. Este obligatoriu sa apara minimum 10 înregistrari pentru a se putea testa aplicatia.

```

CREATE TABLE 18.dbf(locatia
C(25), costul N(4), nrpers N(4),
nrzile N(4))

```

```

input "Dati nr de valori: " to n

```

```

FOR i=1 TO n

```

```

APPEND BLANK

```

```

accept "Dati locatia: " to a
input "Dati costul: " to b
input "Dati nr de persoane: " to
c
input "Dati nr de zile: " to d

```

```

replace locatia WITH a , costul
WITH b , nrpers WITH c , nrzile
WITH d

```

```

ENDFOR

```

```

USE 18.dbf

*0/Introd campuri noi
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x
IF UPPER(ALLTRIM(x))="DA"
MODIFY STRUCTURE
APPEND
ENDIF

*a/Excursie cu costul minim
GO top
exmin=costul
GO 2
SCAN
IF costul<exmin
exmin=costul
ENDIF
ENDSCAN
LIST FIELDS
locatia,costul,nrpers,nrzile FOR
costul=exmin

*b/Afisare excursii catre o
anumita locatie
accept "Dati locatie: " to x
GO TOP
LIST FIELDS
locatia,costul,nrpers,nrzile FOR
locatia=x

*c/Afisare cresc dupa nr de zile
GO top
SORT ON nrzile TO 18_2.dbf
USE 18_2.dbf
LIST FIELDS
locatia,costul,nrpers,nrzile

*d/Scaderea pretului cu 10
pentru o anumita locatie
USE 18.dbf
accept "Dati locatie: " to y
GO top
SCAN
IF locatie=y
costul=costul-10
ENDIF
ENDSCAN

```

19. Sa se realizeze o aplicatie care sa permita gestionarea zborurilor într-un aeroport. Pentru fiecare zbor se memoreaza compania, destinatia, durata în ore, numarul de locuri libere. Aplicatia sa permita introducerea de noi zboruri. Pentru o anumita companie sa se afiseze o lista cu zborurile care mai au locuri libere. De asemenea se va afisa durata maxima a unui zbor precum si destinatiile acelor zboruri având durata maxima. Pot fi adaugate si alte câmpuri în afara celor specificate. Este obligatoriu sa apara minimum 10 înregistrari pentru a se putea testa aplicatia.

```

*CREATE TABLE 19.dbf(compania
C(25), destinatia C(25), durata
N(4), loclib N(4))

input "Dati nr de valori: " to n

FOR i=1 TO n

APPEND BLANK

accept "Dati compania: " to a
accept "Dati destinatia: " to b
input "Dati durata: " to c
input "Dati nr de locuri libere:
" to d

replace compania WITH a ,
destinatia WITH b , durata WITH
c , loclib WITH d

ENDFOR

USE 19.dbf

*0/Introd campuri noi
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x
IF UPPER(ALLTRIM(x))="DA"
MODIFY STRUCTURE
APPEND
ENDIF

```

```

*a/Pentru o anumita companie sa
se afiseze o lista cu zborurile
care mai au locuri libere
accept "Dati compania: " to x
GO top
SCAN
IF compania=x AND loclib>0
?compania,destinatia,durata
ENDIF
ENDSCAN

```

```

*b/Durata maxima a unui zbor +
zborurile cu durata respectiva
maxd=0
GO top
SCAN
IF durata>maxd
maxd=durata
ENDIF
ENDSCAN
LIST FIELDS
compania,destinatia,durata FOR
durata=maxd

```

20. Sa se realizeze o aplicatie care sa permita gestiunea medicamentelor dintr-o farmacie. Pentru fiecare medicament se memoreaza: numele medicamentului, compania furnizoare, pretul întreg, suma compensata (daca medicamentul nu e compensat, aici se va memora valoarea 0). Aplicatia sa permita introducerea de noi medicamente, afisarea medicamentelor de la o anumita companie, numarul medicamentelor compensate, numarul companiilor care au cel putin un medicament. Pot fi adaugate si alte câmpuri în afara celor specificate. Este obligatoriu sa apara minimum 10 înregistrari pentru a se putea testa aplicatia.

```

CREATE TABLE 20.dbf(nume C(25),
compania C(25), pretul N(4),
sumacomp N(4))

```

```

input "Dati nr de valori: " to n

FOR i=1 TO n

APPEND BLANK

```

```

accept "Dati numele
medicamentului: " to a
accept "Dati compania: " to b
input "Dati pretul: " to c
input "Dati suma compensata: "
to d

replace nume WITH a , compania
WITH b , pretul WITH c ,
sumacomp WITH d

```

```

ENDFOR

```

```

USE 20.dbf

```

```

*0/Introd campuri noi
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x
IF UPPER(ALLTRIM(x))="DA"
MODIFY STRUCTURE
APPEND
ENDIF

```

```

*a/Afisare medicamente anumita
companie
accept "Dati compania dorita: "
to x
GO top
LIST FIELDS nume FOR compania=x

```

```

*b/Nr de medicamente compensate
s=0
GO top
SCAN
IF sumacomp=0
s=s+1
ENDIF
ENDSCAN
?s
?

```

```

*c/Nr companii care au cel putin
un medicam
s=0
USE 20.dbf
MODIFY STRUCTURE
replace ALL idd WITH 1
INDEX on compania TAG comp
TOTAL ON compania TO 20_2.dbf
FIELDS idd
USE 20_2.dbf

```

```

GO top
SCAN
IF idd>=1
s=s+1
ENDIF
ENDSCAN
?s
?
```

21. Sa se realizeze o aplicatie care permite gestionarea datelor despre trenurile care vin si pleaca dintr-o gara. Se vor retine date despre numarul trenului, directia din care vine, directia în care pleaca, ora sosirii, ora plecarii, tipul trenului (personal, etc.). Aplicatia sa permita introducerea de noi trenuri, sa afiseze trenurile care pleaca spre o directie data, lista trenurilor care vor sosi într-o statie între anumite ore. Pot fi adaugate si alte câmpuri în afara celor specificate. Este obligatoriu sa apara minimum 10 înregistrari pentru a se putea testa aplicatia.

```

CREATE TABLE 21.dbf (nrtren
N(5), directia C(25), destinatia
C(25), orasos N(4), oraplec
N(4), tiptr C(25))
```

```

input "Dati nr de elemente: " to
n
```

```

FOR i=1 TO n
```

```

APPEND BLANK
```

```

input "Dati nr trenului: " to a
accept "Dati directia din care
vine trenul: " to b
accept "Dati destinatia
trenului: " to c
input "Dati ora sosirii: " to d
input "Dati ora plecarii: " to e
accept "Dati tipul trenului: "
to f
```

```

replace nrtren WITH a , directia
WITH b , destinatia WITH c ,
```

```

orasos WITH d , oraplec WITH e ,
tiptr WITH f
```

```

ENDFOR
```

```

USE 21.dbf
```

```

*0/Introd campuri noi
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x
IF UPPER(ALLTRIM(x))="DA"
MODIFY STRUCTURE
APPEND
ENDIF
```

```

*a/Trenurile care pleaca spre o
directie data
accept "Dati destinatia dorita:
" to x
GO TOP
LIST FIELDS nrtren,tiptr FOR
destinatia=x
```

```

*b/Lista trenuri care ajung la o
destinatie intre anumite ore
input "Dati ora plecarii: " to a
input "Dati ora sorii: " to b
GO top
LIST FIELDS nrtren,tiptr FOR
(oraplec>=a AND orasos<=b)
```

22. Sa se realizeze o aplicatie care sa permita gestionarea datelor referitoare la membrii parlamentului. Pentru fiecare membru se va memora codul, numele, partidul din care face parte, vechimea în partid. Aplicatia va permite introducerea de noi parlamentari (nu se poate introduce un cod care deja exista), afisarea numarului de parlamentari ai fiecarui partid, afisarea alfabetica a parlamentarilor de la un anumit partid. Pot fi adaugate si alte câmpuri în afara celor specificate. Este obligatoriu sa apara minimum 10 înregistrari pentru a se putea testa aplicatia.'

```
CREATE TABLE 22.dbf (codul N(8),
numele C(25), partidul C(25),
vechimea N(4))
```

```
input "Dati nr de introduceri: "
to n
```

```
FOR i=1 TO n
```

```
APPEND BLANK
```

```
input "Dati codul: " to a
accept "Dati numele: " to b
accept "Dati partidul: " to c
input "Dati vechimea: " to d
```

```
replace codul WITH a , numele
WITH b , partidul WITH c ,
vechimea WITH d
```

```
ENDFOR
```

```
USE 22.dbf
```

```
*0/Introd campuri noi
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x
IF UPPER(ALLTRIM(x))="DA"
MODIFY STRUCTURE
APPEND
ENDIF
```

```
*a/Afisare nr parlamentari din
fiecare partid
MODIFY STRUCTURE
replace ALL idd WITH 1
INDEX on
UPPER(ALLTRIM(partidul)) TAG
comp
TOTAL ON
UPPER(ALLTRIM(partidul)) TO
22_4.dbf FIELDS idd
USE 22_4.dbf
LIST FIELDS partidul,idd
```

```
*b/Afisare alfabetica de la un
anumit partid
USE 22.dbf
SORT ON numele TO 22_3.dbf
accept "Dati partidul dorit: "
to x
```

```
LIST FIELDS numele FOR
partidul=x
```

23. Sa se conceapa o aplicatie care gestioneaza date referitoare la campionatul intern de fotbal. Pentru fiecare meci se vor memora: numarul etapei, echipa gazda, echipa oaspete, numarul de goluri înscrise de gazde si numarul de goluri înscrise de oaspeti. Aplicatia va permite introducerea de noi meciuri, afisarea echipelor în ordinea descrescatoare a numarului de goluri înscrise în total, afisarea rezultatelor obsinute de o anumita echipa. Pot fi adaugate si alte câmpuri în afara celor specificate. Este obligatoriu sa apara minimum 10 înregistrari pentru a se putea testa aplicatia.

```
CREATE TABLE 23.dbf (etapa N(4),
gazda C(10), oaspete C(10), gol1
N(3), gol2 N(2))
```

```
input "Dati nr de valori: " to n
```

```
FOR i=1 TO n
```

```
input "Dati nr etapei: " to a
accept "Dati numele echipei
gazda: " to b
accept "Dati numele echipei
oaspete: " to c
input "Dati nr de goluri
inscrise de gazde: " to d
input "Dati nr de goluri
inscrise de oaspeti: " to e
replace etapa WITH a , gazda
WITH b , oaspete WITH c , gol1
WITH d, gol2 WITH e
```

```
ENDFOR
```

```
USE 23.dbf IN 1
```

```
*0/Introd campuri noi
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x
```

```

IF UPPER(ALLTRIM(x))="DA"
MODIFY STRUCTURE
APPEND
ENDIF
x=RECCOUNT()
CREATE TABLE echipe.dbf (nume
C(10),goluri N(10))
USE echipe.dbf IN 2
SELECT 1
FOR i=1 TO x
GO i
y=gazda
z=gol1
SELECT 2
APPEND BLANK
replace nume WITH y, goluri WITH
z
SELECT 1
y=oaspete
z=gol2
SELECT 2
APPEND BLANK
replace nume WITH y, goluri WITH
z
SELECT 1
ENDFOR
SELECT 2
INDEX ON nume TAG comp
TOTAL ON nume TO tabel.dbf
FIELDS goluri
USE tabel.dbf
SORT ON goluri/d TO tab2.dbf
USE tab2.dbf
LIST
SELECT 1
accept "dati numele echipei
cautate" to x
LIST FIELDS
gazda,gol1,gol2,oaspete FOR
gazda=x OR oaspete=x

```

24. Sa se realizeze o aplicatie pentru gestionarea datelor despre animalele detinute de o gradina zoologica. Se vor memora pentru fiecare animal: numarul unic de identificare, specia, sexul, vârsta în ani. Aplicatia va permite adaugarea de noi animale, afisarea datelor despre animalele

dintr-o specie data, numarul animalelor femele, vârsta maxima si animalele care au aceasta vârsta maxima. Pot fi adaugate si alte câmpuri în afara celor specificate. Este obligatoriu sa apara minimum 10 înregistrari pentru a se putea testa aplicatia.

```

CREATE TABLE 24.dbf (nriden
N(4), specia C(25), sexul C(25),
varsta N(3))

```

```
input "Dati nr de valori: " to n
```

```
FOR i=1 TO n
```

```
input "Dati nr de indentificare:
" to a
```

```
accept "Dati specia: " to b
```

```
accept "Dati sexul: " to c
```

```
input "Dati varsta: " to d
```

```
replace nriden WITH a , specia
WITH b , sexul WITH c , varsta
WITH d
```

```
ENDFOR
```

```
USE 24.dbf
```

```
*0/Introd campuri noi
```

```
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x
```

```
IF UPPER(ALLTRIM(x))="DA"
```

```
MODIFY STRUCTURE
```

```
APPEND
```

```
ENDIF
```

```
*a/Afisare date despre animale
dintr-o specie data
```

```
accept "Dati specia: " to x
```

```
GO top
```

```
LIST FIELDS
```

```
nriden,specia,sexul,varsta FOR
specia=x
```

```
*b/Nr animale femele
```

```
s=0
```

```
GO top
```

```
SCAN
```

```
IF UPPER(ALLTRIM(sexul))="F"
```

```
s=s+1
```

```

ENDIF
ENDSCAN

*c/Varsta maxima si animale cu
varsta maxima
vmax=0
GO top
SCAN
IF varsta>vmax
vmax=varsta
ENDIF
ENDSCAN
GO top
?"Varsta maxima e: ",vmax," la
animalele:"
LIST FIELDS nriden,specia,sexul
FOR varsta=vmax

```

25. Sa se realizeze o aplicatie pentru gestionarea unor date despre filme la un cinematograf. Se vor memora pentru fiecare film: codul, regizorul, anul aparitiei, numele filmului, durata (în minute). Aplicatia va permite introducerea de noi filme, afisarea numarului de filme regizate de fiecare regizor, afisarea duratei maxime a unui film, afisarea filmelor aparute într-un anumit an. Pot fi adaugate si alte câmpuri în afara celor specificate. Este obligatoriu sa apara minimum 10 înregistrari pentru a se putea testa aplicatia.

```

CREATE TABLE 25.dbf (cod N(10),
regizor C(25), anap N(4),
numefilm C(25), durata N(5))

input "Dati nr de valori: " to n

FOR i=1 TO n

input "Dati codul: " to a
accept "Dati regizorul: " to b
input "Dati anul aparitiei: " to
c
accept "Dati numele filmului: "
to d

```

```

input "Dati durata filmului in
minute: " to e

```

```

replace cod WITH a , regizor
WITH b , anap WITH c , numefilm
WITH d , durata WITH e

```

```

ENDFOR

```

```

USE 25.dbf

```

```

*0/Introd campuri noi
accept "Daca doriti sa adaugati
campuri noi scrieti 'DA' " to x
IF UPPER(ALLTRIM(x))="DA"
MODIFY STRUCTURE
APPEND
ENDIF

```

```

*a/Afisare filme regizate de
fiecare regizor

```

```

USE 25.dbf
MODIFY STRUCTURE
REPLACE ALL idd WITH 1
INDEX on regizor TAG comp
TOTAL ON regizor to 25_2.dbf
FIELDS idd
USE 25_2.dbf
LIST FIELDS numefilm

```

```

*b/Durata maxima

```

```

USE 25.dbf
durmax=0
GO top
SCAN
IF durata>durmax
durmax=durata
ENDIF
ENDSCAN
LIST FIELDS numefilm FOR
durata=durmax

```

```

*c/Afisare filme aparute intr-un
anumit an

```

```

GO top
input "Dati anul: " to x
LIST FIELDS numefilm FOR anap=x

```