Program to create match records for a horoscope

\*\* Comments

Public variables

\*\* end of comments

public c01,c02,c03,c04,c05,c06,c07,c08,c09,c10,c11,c12

public a3,a4,a5,a6,a7,s12

public m\_planet

public m\_key

\*\* Comments

New array() command defines a new array

\*\* end of comments

s12 = new array()

\*\* Comments

Select in dbase means to use that file. In dbase we can assign an alias which is 1 in this case. We can give the command later as select 1 which means use the file in 1 which is F2PLANETS

Go top means go to the beginning of the file (first record)

\*\* end of comments

select 1

use F2PLANETS

go top

select 2

use cusp

select 3

use match\_file

\*\* Comments

Beginning of program – Section 1

Do While not eof() and Enddo is a close loop which checks whether the end of file has been reached. If yes the loop will be exited

Ltrim \*trim on left side of string

Rtrim is trim on right side of string

Len is length of string

For and endfor are increment loops for a integer with start and end points. In case below the loop will be run from 1 to l (t2/2)

In this section every record from cusp is loaded in arrays (C01…C12) from S1 in CUSP file.

\*\* end of comments

select 2

go top

do while .not. eof()

store len(ltrim(rtrim(s1))) to tl

store tl/2 to l

if cuspid = '01'

c01 = new array(l)

start = 1

for k = 1 to l

c01[k] = substr(s1,start,2)

start = start + 2

endfor

endif

if cuspid = '02'

c02 = new array(l)

start = 1

for k = 1 to l

c02[k] = substr(s1,start,2)

start = start + 2

endfor

endif

if cuspid = '03'

c03 = new array(l)

start = 1

for k = 1 to l

c03[k] = substr(s1,start,2)

start = start + 2

endfor

endif

if cuspid = '04'

c04 = new array(l)

start = 1

for k = 1 to l

c04[k] = substr(s1,start,2)

start = start + 2

endfor

endif

if cuspid = '05'

c05 = new array(l)

start = 1

for k = 1 to l

c05[k] = substr(s1,start,2)

start = start + 2

endfor

endif

if cuspid = '06'

c06 = new array(l)

start = 1

for k = 1 to l

c06[k] = substr(s1,start,2)

start = start + 2

endfor

endif

if cuspid = '07'

c07 = new array(l)

start = 1

for k = 1 to l

c07[k] = substr(s1,start,2)

start = start + 2

endfor

endif

if cuspid = '08'

c08 = new array(l)

start = 1

for k = 1 to l

c08[k] = substr(s1,start,2)

start = start + 2

endfor

endif

if cuspid = '09'

c09 = new array(l)

start = 1

for k = 1 to l

c09[k] = substr(s1,start,2)

start = start + 2

endfor

endif

if cuspid = '10'

c10 = new array(l)

start = 1

for k = 1 to l

c10[k] = substr(s1,start,2)

start = start + 2

endfor

endif

if cuspid = '11'

c11 = new array(l)

start = 1

for k = 1 to l

c11[k] = substr(s1,start,2)

start = start + 2

endfor

endif

if cuspid = '12'

c12 = new array(l)

start = 1

for k = 1 to l

c12[k] = substr(s1,start,2)

start = start + 2

endfor

endif

skip

enddo

\*\* Comments

Beginning of program – Section 2

Store means assign value to the variable

P0..p8 are columns in F2PLANETS table

Match\_key is a function with parameters

Skip means go to the next record

So this record reads every record in F2PLANET, assign values to an array a8 and sends to match\_key function

\*\* end of comments

a8 = new array(8)

sele 1

go top

do while .not. eof()

store p0+p1+p2+p3+p4+p5+p6+p7 to m\_key

\* p0

a8[1] = p0

a8[2] = p1

a8[3] = p2

a8[4] = p3

a8[5] = p4

a8[6] = p5

a8[7] = p6

a8[8] = p7

match\_key(a8,c01,'01')

match\_key(a8,c02,'02')

match\_key(a8,c03,'03')

match\_key(a8,c04,'04')

match\_key(a8,c05,'05')

match\_key(a8,c06,'06')

match\_key(a8,c07,'07')

match\_key(a8,c08,'08')

match\_key(a8,c09,'09')

match\_key(a8,c10,'10')

match\_key(a8,c11,'11')

match\_key(a8,c12,'12')

\* p1

a8[1] = p1

a8[2] = p2

a8[3] = p3

a8[4] = p4

a8[5] = p5

a8[6] = p6

a8[7] = p7

a8[8] = p0

match\_key(a8,c01,'01')

match\_key(a8,c02,'02')

match\_key(a8,c03,'03')

match\_key(a8,c04,'04')

match\_key(a8,c05,'05')

match\_key(a8,c06,'06')

match\_key(a8,c07,'07')

match\_key(a8,c08,'08')

match\_key(a8,c09,'09')

match\_key(a8,c10,'10')

match\_key(a8,c11,'11')

match\_key(a8,c12,'12')

\* p2

a8[1] = p2

a8[2] = p3

a8[3] = p4

a8[4] = p5

a8[5] = p6

a8[6] = p7

a8[7] = p0

a8[8] = p1

match\_key(a8,c01,'01')

match\_key(a8,c02,'02')

match\_key(a8,c03,'03')

match\_key(a8,c04,'04')

match\_key(a8,c05,'05')

match\_key(a8,c06,'06')

match\_key(a8,c07,'07')

match\_key(a8,c08,'08')

match\_key(a8,c09,'09')

match\_key(a8,c10,'10')

match\_key(a8,c11,'11')

match\_key(a8,c12,'12')

\*p3

a8[1] = p3

a8[2] = p4

a8[3] = p5

a8[4] = p6

a8[5] = p7

a8[6] = p0

a8[7] = p1

a8[8] = p2

match\_key(a8,c01,'01')

match\_key(a8,c02,'02')

match\_key(a8,c03,'03')

match\_key(a8,c04,'04')

match\_key(a8,c05,'05')

match\_key(a8,c06,'06')

match\_key(a8,c07,'07')

match\_key(a8,c08,'08')

match\_key(a8,c09,'09')

match\_key(a8,c10,'10')

match\_key(a8,c11,'11')

match\_key(a8,c12,'12')

\* p4

a8[1] = p4

a8[2] = p5

a8[3] = p6

a8[4] = p7

a8[5] = p0

a8[6] = p1

a8[7] = p2

a8[8] = p3

match\_key(a8,c01,'01')

match\_key(a8,c02,'02')

match\_key(a8,c03,'03')

match\_key(a8,c04,'04')

match\_key(a8,c05,'05')

match\_key(a8,c06,'06')

match\_key(a8,c07,'07')

match\_key(a8,c08,'08')

match\_key(a8,c09,'09')

match\_key(a8,c10,'10')

match\_key(a8,c11,'11')

match\_key(a8,c12,'12')

\* p5

a8[1] = p5

a8[2] = p6

a8[3] = p7

a8[4] = p0

a8[5] = p1

a8[6] = p2

a8[7] = p3

a8[8] = p4

match\_key(a8,c01,'01')

match\_key(a8,c02,'02')

match\_key(a8,c03,'03')

match\_key(a8,c04,'04')

match\_key(a8,c05,'05')

match\_key(a8,c06,'06')

match\_key(a8,c07,'07')

match\_key(a8,c08,'08')

match\_key(a8,c09,'09')

match\_key(a8,c10,'10')

match\_key(a8,c11,'11')

match\_key(a8,c12,'12')

\* p6

a8[1] = p6

a8[2] = p7

a8[3] = p0

a8[4] = p1

a8[5] = p3

a8[6] = p4

a8[7] = p5

a8[8] = p6

match\_key(a8,c01,'01')

match\_key(a8,c02,'02')

match\_key(a8,c03,'03')

match\_key(a8,c04,'04')

match\_key(a8,c05,'05')

match\_key(a8,c06,'06')

match\_key(a8,c07,'07')

match\_key(a8,c08,'08')

match\_key(a8,c09,'09')

match\_key(a8,c10,'10')

match\_key(a8,c11,'11')

match\_key(a8,c12,'12')

\*p7

a8[1] = p7

a8[2] = p0

a8[3] = p1

a8[4] = p2

a8[5] = p3

a8[6] = p4

a8[7] = p5

a8[8] = p6

match\_key(a8,c01,'01')

match\_key(a8,c02,'02')

match\_key(a8,c03,'03')

match\_key(a8,c04,'04')

match\_key(a8,c05,'05')

match\_key(a8,c06,'06')

match\_key(a8,c07,'07')

match\_key(a8,c08,'08')

match\_key(a8,c09,'09')

match\_key(a8,c10,'10')

match\_key(a8,c11,'11')

match\_key(a8,c12,'12')

select 1

skip

enddo

close all

\*\* Comments

Beginning of Function

Alen() gives the length of the array

The parameters are matched Combo-a8, cusp – CXX (c01 ..C12) and cloc is the cuspid sent to the function

? means print on console

\*\* end of comments

function match\_key(combo,Cuspp,cloc)

fl1 = "N"

fl2 = "N"

fl3 = "N"

fl4 = "N"

fl5 = "N"

fl6 = "N"

fl7 = "N"

m\_str= ""

xcusp = new array(alen(cuspp))

xcombo = new arrAy(alen(combo))

\*\* Comments

This part below just prepares a string from the array

\*\* end of comments

pstr1 = ""

for i = 1 to alen(Combo)

pstr1 = pstr1 + Combo[i]

endfor

pstr2 = ""

for i = 1 to alen(Cuspp)

pstr2 = pstr2 + Cuspp[i]

endfor

\*\* Check first position

\*\* Comments

In the section below a working copy of the array is taken

Acopy() copies an array to a new array

It basically checks for every continuous match element from xcombo and xcuspp and if it matches it updated the xcuspp element with XX (to avoid duplicate matching)

\*\* end of comments

acopy(cuspp,xcusp)

acopy(combo,xcombo)

\*Checks first position of xcombo

for i = 1 to alen(xcusp)

if xcombo[1] = xcusp[i]

m\_str = m\_str + xcombo[1]

xcusp[i] = 'XX'

fl1 = 'Y'

\*\* IF THE CODE ENTERS THIS CONDITION IT WILL EXIT THE

FOR LOOP

exit

endif

if xcombo[1] = cuspp[i]

m\_str = m\_str + xcombo[1]

fl1 = 'Y'

endif

endfor

\*Checks second position of xcombo

if fl1 = 'Y' \*\* if the previous position of xcombo is found in Cuspp

for i = 1 to alen(xcusp)

if xcombo[2] = xcusp[i]

m\_str = m\_str + xcombo[2]

xcusp[i] = 'XX'

fl2 = 'Y'

exit

endif

if xcombo[2] = cuspp[i]

m\_str = m\_str + xcombo[2]

fl2 = 'Y'

endif

endfor

endif

\*Checks third position of xcombo

if fl1 = 'Y' and fl2 = 'Y'

for i = 1 to alen(xcusp)

if xcombo[3] = xcusp[i]

m\_str = m\_str + xcombo[3]

xcusp[i] = 'XX'

fl3 = 'Y'

exit

endif

if xcombo[3] = cuspp[i]

m\_str = m\_str + xcombo[3]

fl3 = 'Y'

endif

endfor

endif

* Checks fourth position of xcombo

if fl1 = 'Y' and fl2 = 'Y' and fl3 = 'Y'

for i = 1 to alen(xcusp)

if xcombo[4] = xcusp[i]

m\_str = m\_str + xcombo[4]

xcusp[i] = 'XX'

fl4 = 'Y'

exit

endif

if xcombo[4] = cuspp[i]

m\_str = m\_str + xcombo[4]

fl4 = 'Y'

endif

endfor

endif

\*Checks fifth position of combo

if fl1 = 'Y' and fl2 = 'Y' and fl3 = 'Y' and fl4 = 'Y'

for i = 1 to alen(xcusp)

if xcombo[5] = xcusp[i]

m\_str = m\_str + xcombo[5]

xcusp[i] = 'XX'

fl5 = 'Y'

exit

endif

if xcombo[5] = cuspp[i]

m\_str = m\_str + xcombo[5]

fl5 = 'Y'

endif

endfor

endif

\*Checks 6th

if fl1 = 'Y' and fl2 = 'Y' and fl3 = 'Y' and fl4 = 'Y' and fl5 = 'Y'

for i = 1 to alen(xcusp)

if xcombo[6] = xcusp[i]

m\_str = m\_str + xcombo[6]

xcusp[i] = 'XX'

fl6 = 'Y'

exit

endif

if xcombo[6] = cuspp[i]

m\_str = m\_str + xcombo[6]

fl6 = 'Y'

endif

endfor

endif

\*checks 7th

if fl1 = 'Y' and fl2 = 'Y' and fl3 = 'Y' and fl4 = 'Y' and fl5 = 'Y' and fl6 = 'Y'

for i = 1 to alen(xcusp)

if xcombo[7] = xcusp[i]

m\_str = m\_str + xcombo[7]

xcusp[i] = 'XX'

fl7 = 'Y'

exit

endif

if xcombo[7] = cuspp[i]

m\_str = m\_str + xcombo[7]

fl7 = 'Y'

endif

endfor

endif

\*checks 8th

if fl1 = 'Y' and fl2 = 'Y' and fl3 = 'Y' and fl4 = 'Y' and fl5 = 'Y' and fl6 = 'Y' and fl7=’Y’

for i = 1 to alen(xcusp)

if xcombo[8] = xcusp[i]

m\_str = m\_str + xcombo[8]

xcusp[i] = 'XX'

fl8 = 'Y'

exit

endif

if xcombo[8] = cuspp[i]

m\_str = m\_str + xcombo[8]

fl7 = 'Y'

endif

endfor

endif

\*\* Comments

\*\* if all cusp positions have been matched in a sequence in xcombo (if there is no sequential match the program will exit and the cussp array with contact non XX values

For example in:

SA,VE,JU (cusp) SA,VE,JU,ME,MO,MA,SU,VE (xcombo)

There is a continuous match SAVEJU in first three positions and f1 will be Y f2 will be Y and f3 will be Y and cuspp will be XXXXXX at the end of this routing

SA,VE,JU (cusp) SA,VE,ME,ME,MO,MA,SU,VE

In the above example there will be only two matches F1 will be Y and F2 will be Y and cusp will be XXXXJU

If CUSPP Is all XXXXXX… this means there is a match. Bewlo code checks for non XX in cuspp

\*\* end of comments

pattern = true

for i = 1 to alen(xcusp)

if xcusp[i] <> 'XX'

pattern = false

endif

endfor

\*\* if match the variable pattern will be true

if pattern = true

\*\* this section adds a record on match\_file

\*\* appe blank means add a blank record

select 3

appe blank

replace uid with "XXXXXXXXXXXXX"

replace hid with "100001"

replace planet with m\_planet

replace key with m\_key

replace combo with pstr1

replace cUSPSTR with pstr2

replace cuspid with cloc

replace matchstr with m\_str

endif

select 1

return