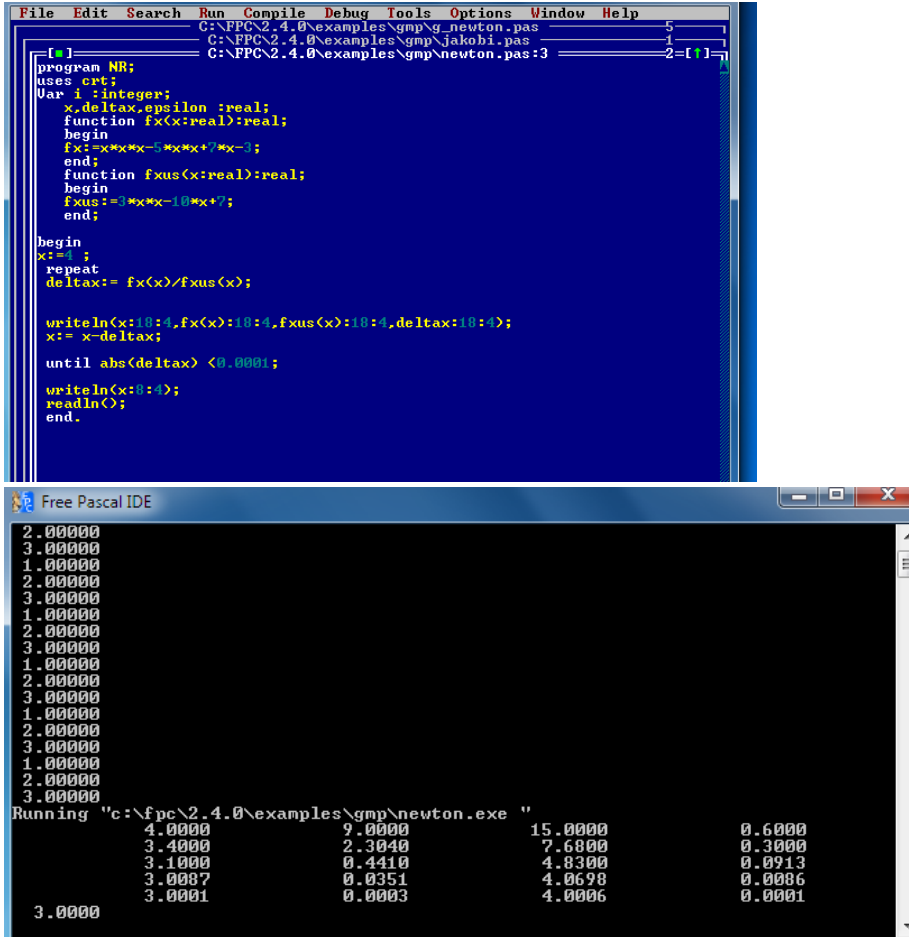


## Sayısal Çözümleme 1.Dönem Ödevi

- 1- Ekte verilen Newton Raphson, Matris Çarpma, Matris Tersi Alma ( Gauss-Jordan Yönt.), Jakobi İteratif Çözümleme PASCAL programlarını bilgisayarınıza kopyalayıp çalıştırın. Free Paccal veya Dev Pascal kullanılabilir.
- 2- Newton Raphson ile 3. Derece veya daha yüksek bir polinomun köklerinden ikisini uygun başlangıç değerleri ile bularak programın örneğe göre düzeltilmiş görüntüsünün ve çıktı ekranının çıktısını "PrtSc" ile bir doküman dosyasına kopyalayın. (Aşağıdaki örneğe bakın)



```
File Edit Search Run Compile Debug Tools Options Window Help
C:\FPC\2.4.0\examples\gmp\g_newton.pas 5
C:\FPC\2.4.0\examples\gmp\jakobi.pas 1
C:\FPC\2.4.0\examples\gmp\newton.pas:3 2-[F1]=

program NR;
uses crt;
var i : integer;
    x,deltax,epsilon : real;
function fx(x:real):real;
begin
  fx:=x*x*x-5*x*x+7*x-3;
end;
function fxus(x:real):real;
begin
  fxus:=3*x*x-10*x+7;
end;
begin
  x:=4;
  repeat
    deltax:= fx(x)/fxus(x);
    x:= x-deltax;
  until abs(deltax) < 0.0001;
  writeln(x:10:4,fx(x):10:4,fxus(x):10:4,deltax:10:4);
  x:= x-deltax;
  until abs(deltax) < 0.0001;
  writeln(x:10:4);
  readln();
end.
```

```
Free Pascal IDE
2.00000
3.00000
1.00000
2.00000
3.00000
1.00000
2.00000
3.00000
1.00000
2.00000
3.00000
1.00000
2.00000
3.00000
1.00000
2.00000
3.00000
1.00000
2.00000
3.00000
Running "c:\fpc\2.4.0\examples\gmp\newton.exe "
4.0000 9.0000 15.0000 0.6000
3.4000 2.3040 7.6800 0.3000
3.1000 0.4410 4.8300 0.0913
3.0007 0.0351 4.0698 0.0086
3.0001 0.0003 4.0006 0.0001
3.0000
```

- 3- Kendi seçeceğiniz 4x4'lük bir matrisin tersini, Matris tersi alma programı ile hesaplatarak matrisin tanımlandığı ve tersinin hesaplandığı ekran çıktısını yukarıdaki sekilde aynı dosyaya kopyalayın.

```

program macinv;
uses crt;

type MATRIS = array [1..20,1..20] of real;
var i,j,k,l,m,n :integer;
A,B,C :MATRIS;
p,t,det :real;
Label 1,2 ;

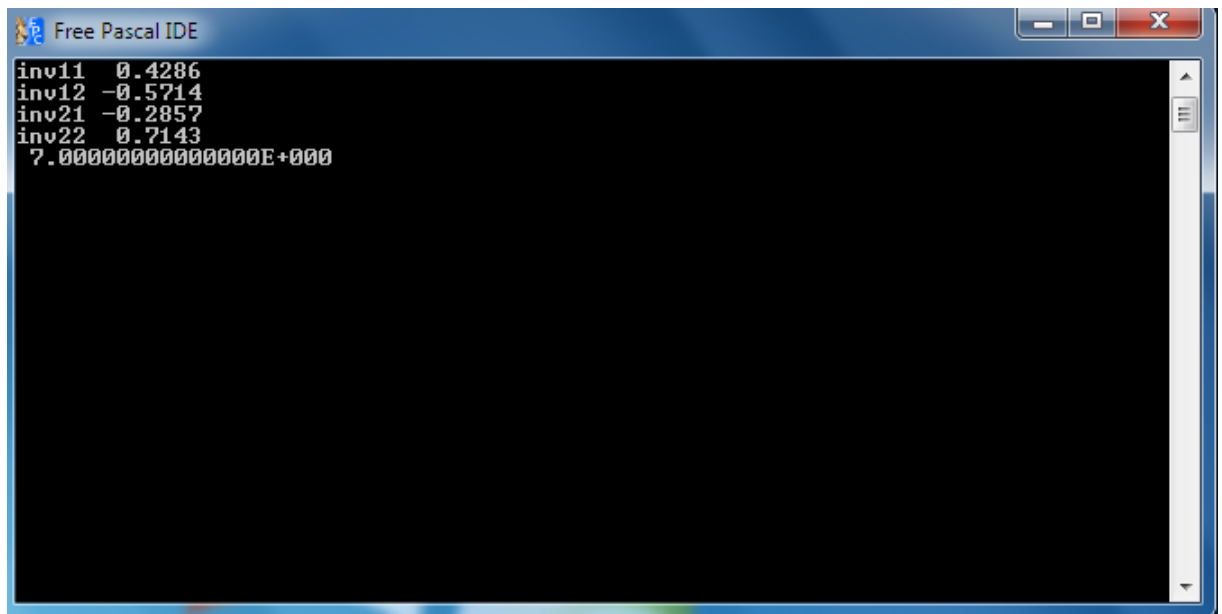
begin
clrscr;

A[1,1]:= 5 ; A[1,2]:= 4 ;
A[2,1]:= 2 ; A[2,2]:= 3 ;
N:=2;M:=2 ; det:=1;
B[1,1]:= 1 ; B[1,2]:=0 ;
B[2,1]:= 0 ; B[2,2]:=1 ;

for j:=1 to n do begin
T:=A[j,j];
det:=det*t;
for k:=1 to n do begin
a[j,k]:=a[j,k]/T;
b[j,k]:=b[j,k]/T;
end;
for i:= 1 to n do begin
p:=a[i,j];
if i=j then goto 1;
for k:=1 to n do begin
b[i,k]:=b[i,k]-p*b[j,k];
a[i,k]:=a[i,k]-p*a[j,k];
end;
1: end;
end;
for i:=1 to n do begin
for j:=1 to n do begin
writeln('inv',i,j,b[i,j]:0:4,'');
end;
end;
writeln(det);

end.

```



\*

4- Aynı programda  $A[1,1]=a_1 \dots A[n,n]=a_n$  ;  $B[1,1]=b_1 \dots B[n,1]=b_n$  tanımını yaparak en az 4x4'lük bir denklem sistemini çözerek ekran çıktılarını kopyalayın.

5-Jakobi iterasyonunu kullanarak 3 bilinmeyenli 3 denklemleri çözerek aynı işlemleri tekrarlayın. Denlemlerin yakınsamma koşulunu sağlamasına dikkat edin.

6- Tüm işlemleri kaydettiğiniz. DOCX dosyasını pdf olarak kayıt ederek okulnumarası.PDF ( örnek 100205055.pdf) şeklinde [tduru@kocaeli.edu.tr](mailto:tduru@kocaeli.edu.tr) adresine 7 Nisan 2013 Pazar 24:00' e kadar posta ile gönderiniz.