

Pascal Programlama Dili ile elementer matris işlemleri;

<pre> Uses crt; Var A : Array[1..3,1..3] Of real; i,j,k : Integer; carpan,det : real; procedure matris_yaz; begin For i:=1 To 3 Do Begin For j:=1 To 3 Do Write (A[i,j]:7:2); WriteLn; End; writeln; end; Begin WriteLn('3x3 boyutlu Matris :'); A[1,1] := 1 ; A[1,2] := 2 ; A[1,3] := 3 ; A[2,1] := 3 ; A[2,2] := 1 ; A[2,3] := 2 ; A[3,1] := 2 ; A[3,2] := 3 ; A[3,3] := 1 ; matris_yaz; WriteLn('1. satirin 2. satira eklenmesi :'); i:=1; For j:=1 To 3 Do Begin A [i+1,j] := A [i,j] + A [i+1,j] ; End; matris_yaz; WriteLn('3. satirin -2 ile carpilmasi :'); i:=3; For j:=1 To 3 Do Begin A [i,j] := A [i,j] * (-2) ; End; matris_yaz; WriteLn('a11 elemani altindakilerin sifirlanmasi :'); For i:=1 To 3 Do begin if (i=1) then continue; carpan := (1/A [1,1]) * A [i,1] * (-1); For j:=1 To 3 Do Begin A [i,j] := carpan * A [1,j] + A [i,j] ; End; end; matris_yaz; </pre>	<pre> WriteLn('a22 elemani altindakilerin sifirlanmasi :'); For i:=1 To 3 Do begin if (i<=2) then continue; carpan := (1/A [2,2]) * A [i,2] * (-1); For j:=1 To 3 Do Begin A [i,j] := carpan * A [2,j] + A [i,j] ; End; end; matris_yaz; { WriteLn('Kosegen altindakilerin sifirlanmasi :'); for k:=1 to 3 do For i:=k To 3 Do begin if (i=k) then continue; carpan := (1/A [k,k]) * A [i,k] * (-1); For j:=1 To 3 Do Begin A [i,j] := carpan * A [k,j] + A [i,j] ; End; end; matris_yaz; WriteLn('Determinant :'); det := 1; for k:=1 to 3 do det :=det*A [k,k] ; writeln ('Determinant =',det:6:2); } readln; End. </pre>
---	---

Örnek :

$f(x) = e^{x-2}$ fonksiyonun $x=2$ noktasındaki 1. ve 2. türevini, $h=0.1$ alarak yaklaşık olarak tüm yöntemleri kullanarak bulunuz.

```
%Matlab%
```

```
clc;
```

```
h=0.1; a=2;
```

```
syms x %x degiskenini tanimla
```

```
f1 = exp(x-2); %fonksiyon tanimlandi  
f = inline(char(f1)); %f(x) olarak atandi
```

```
%Basit ileri fark
```

```
bif = (1/h)*(f(a+h)-f(a));
```

```
%Basit geri fark
```

```
bgf = (1/h)*(f(a)-f(a-h));
```

```
%Merkez fark
```

```
mf = (1/(2*h))*(f(a+h)-f(a-h));
```

```
%2 noktali ileri fark taylor 1.turev
```

```
t2ileri = (1/(2*h))*(-3*f(a)-f(a+2*h)+4*f(a+h));
```

```
%2 noktali geri fark taylor 1.turev
```

```
t2geri = (1/(2*h))*(3*f(a)+f(a-2*h)-4*f(a-h));
```

```
%3 noktali ileri fark taylor 1.turev
```

```
t3ileri = (1/(6*h))*(-11*f(a)+18*f(a+h)-9*f(a+2*h)+2*f(a+3*h));
```

```
%2 noktali ileri fark taylor 2.turev
```

```
t2ileri2 = (1/(h*h))*(f(a)-2*f(a+h)+f(a+2*h));
```

```
%2 noktali geri fark taylor 2.turev
```

```
t2geri2 = (1/(h*h))*(f(a)-2*f(a-h)+f(a-2*h));
```

```
fprintf('\n\te^(x-2) fonksiyonun sayisal turevi\n');
```

```
fprintf('\tAnalitik 1. ve 2. turev = %.5f \n\n',f(2));
```

```
fprintf('basit ileri fark: %.5f hata=%%%.5f\n',bif, 100*abs(1-bif));
```

```
fprintf('basit geri fark: %.5f hata=%%%.5f\n',bgf, 100*abs(1-bgf));
```

```
fprintf('merkez fark: %.5f hata=%%%.5f\n\n',mf,100*abs(1-mf) );
```

```
fprintf('2 noktali ileri Taylor serisi ile 1.turev: %.5f hata=%%%.5f\n',t2ileri, 100*abs(1-t2ileri));
```

```
fprintf('2 noktali geri Taylor serisi ile 1.turev: %.5f hata=%%%.5f\n\n',t2geri, 100*abs(1-t2geri));
```

```
fprintf('3 noktali ileri Taylor serisi ile 1.turev: %.5f hata=%%%.5f\n\n',t3ileri, 100*abs(1-t3ileri));
```

```
fprintf('2 noktali ileri Taylor serisi ile 2.turev: %.5f hata=%%%.5f\n',t2ileri2, 100*abs(1-t2ileri2));
```

```
fprintf('2 noktali geri Taylor serisi ile 2.turev: %.5f hata=%%%.5f\n',t2geri2, 100*abs(1-t2geri2));
```