

PAMUKKALE ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ  
BÖLÜMÜ

**CENG 241 BİLİMSEL HESAPLAMA DERSİ VİZE SINAVI**

Soru1	Soru2	Soru3	Soru4	Soru5	Soru6	Soru7		TOPLAM
8	32	10	10	10	10	10		100

**diag(V,K)** when V is a vector with N components is a square matrix of order N+ABS(K) with the elements of V on the K<sup>th</sup> diagonal. K=0 is the main diagonal, K>0 is above the main diagonal and K<0 is below the main diagonal. **diag(V)** is the same as **diag(V,0)** and puts V on the main diagonal. **diag(X,K)** when X is a matrix is a column vector formed from the elements of the K-th diagonal of X.

**exp(X)** is the exponential of the elements of X, e to the X.

**eye(M,N)** or **eye([M,N])** is an M-by-N matrix with 1's on the diagonal and zeros elsewhere.

**figure**, by itself, creates a new figure window, and returns its handle.

**fprintf** Write formatted data to text file or command window (if no file ID given).

**global X Y Z** defines X, Y, and Z as global in scope.

**length(X)** returns the length of vector X.

**linspace(X1, X2, N)** generates N points between X1 and X2.

**logical(X)** converts the elements of the array X into logicals

**nargin** Number of function input arguments.

**num = input(PROMPT)** displays the PROMPT string on the screen

**ones(M,N)** or **ones([M,N])** is an M-by-N matrix of ones.

**plot(X,Y)** plots vector Y versus vector X.

**rem(X,Y)** returns the remainder on integer division of X by Y. If this is 0, then Y is a factor of X.

**s = sum(X)** is the sum of the elements of the vector X

**I = uint8(X)** converts the elements of the array X into unsigned 8-bit integers.

**C = unique(A)** A dizisinin elemanlarını tekrarsız olarak küçükten büyüğe sıralı döndürür.

**varargin** Variable length input argument list. Allows any number of arguments to a function. The variable varargin is a cell array containing the optional arguments to the function.

**zeros(M,N)** or **ZEROS([M,N])** is an M-by-N matrix of zeros.

#### SORU 1)

Aşağıdaki fonksiyon komut satırından

```
>>a=[9 6 8 9 5 6 2 0 1 2 9 5 6]; b=[0 1 2 3 4 9 8 8 9 1 2 0];
```

```
>>y=myfun(a,b)
```

şeklinde çağrılmaktadır. Çalışırsa çıktısını çalışmazsa nedenini yazınız.

```
function [out] = myfun(a,b)
a = unique(a); b = unique(b);
out = [];
for j = 1:length(a)
    if(sum(b==a(j))) % if a(j) is in b
        out(end+1) = a(j);
    end
end
```

```
>> y =      0      1      2      8      9
```

**SORU 2)** `X = [ 11, 12, 13, 15 ; 21, 22, 23,24; 31, 32, 33,34; 41, 42, 43,44 ];` `f=@(x) ~logical(rem(x,2))`  
`x=[-1, 2, 3, -2], y=[0.2, 3.1, 0, -3], z=[3, 0, 1, 0.1], arr = {5, [1, 4, 3], 'r'}`  
 olarak verildiğine göre aşağıdaki işlemleri çözünüz:

a) `X(sum((X>=12 & X<=21)))`  
`ans= [11 11 11 11]`

b) `min(X(X>=13 & X<=21))`  
`ans=13`

c) `X(max((X>=13 & X<=21)))`  
`ans = 11 31 41`

d) `y=[];`  
`for x=1:5,`  
`if(x==4),continue; end,`  
`y=[x y x];`  
`end, disp(y)`  
`y = 5 3 2 1 1 2 3 5`

e) `X(end-4)`  
`ans = 43`

f) `X(end-2,end-3)`  
`ans = 21`

g) `[diag([1 2],1); diag(X,1)']`  
`ans =0 1 0`  
`0 0 2`  
`0 0 0`  
`12 23 34`

h) `arr{2}(2)`  
`ans=4`

i) `f(X)`  
`ans = 0 1 0 0`  
`0 1 0 1`  
`0 1 0 1`  
`0 1 0 1`

### SORU 3)

`M=[-2 1 1 1 1...1;1 -4 1 1 1... 1; 1 1 -6 1 ...1 1; 1 1 1 -8 1...1 1;...;1 1...1 1 -2N];`

şeklinde verilen matrisi N değerinin verildiğini varsayarak,

a) Çevrim kontrol komutları kullanarak elde ediniz

```
M=ones(N);
for i=1:N
    M(i,i)=-2*i;
end
```

b) Herhangi bir çevrim kullanmadan tek atama işlemi ile Matlab'ın `ones`, `zeros`, `eyes`, `diag` vb matris oluşturma komutlarını kullanarak elde ediniz.

```
M=ones(N)+diag(-2*(1:N))-eye(N)
```

j) `X ( (X < 24) & f(X) )'`  
`ans= 12 22`

k) `Y=X; Y(4:-2:1,:)=Y(:,2:2:5)'`  
`Y =11 12 13 15`  
`15 24 34 44`  
`31 32 33 34`  
`12 22 32 42`

l) `zeros(uint8( X(3,4)/10 ),4)`  
`ans =0 0 0 0`  
`0 0 0 0`  
`0 0 0 0`

m) `z(x + ~y > z)`  
`ans= 0 1`

n) `x ~= -5*y`  
`ans = 0 1 1 1`

o) `X([3 1 2],:)`  
`ans =31 32 33 34`  
`11 12 13 15`  
`21 22 23 24`

p) `X([3 1], :) =[]`  
`X = 21 22 23 24`  
`41 42 43 44`

**SORU 2)** Aşağıda DenizliSpor'un maçlarının bir bölümü tablo olarak verilmiştir. Bu bilgileri Denizli adında bir struct dizisi ve Rakip, EvSahibi, DenizliSkor, RakipSkor alanları ile tutulmaktadır. Alanların içerikleri sırası ile char, logical, double ve double olarak tutulmaktadır.

Rakip	EvSahibi	DenizliSkor	RakipSkor
Fenerbahçe	True	4	3
Galatasaray	False	2	2
Beşiktaş	True	2	1
Trabzonspor	False	0	3

a) struct yapısını oluşturan komut satırını struct(field,value,...) kullanarak yazınız.

```
>>Denizli=struct('Rakip',{'Fenerbahçe','Galatasaray','Beşiktaş','Trabzonspor'},
'EvSahibi',{true,false,true,false},
'DenizliSkor',{4,2,2,0},
'RakipSkor',{3,2,1,3})
```

b) Denizlisporun attığı golleri DS dizisine, rakiplerin attığı golleri RS dizisine aktarınız. (DS=[4 2 2 0] olmalı)

```
>> for i=1:4 DS(i)=Denizli(i).DenizliSkor;end
>> for i=1:4 RS(i)=Denizli(i).RakipSkor;end
```

c) DS ve RS'den yararlanarak Denizlispor'un galibiyet sayısını bulunuz.

```
>>sum(DS>RS)
```

**SORU 5)**  $x=1:0.1:20$ ; olarak alınması istenmektedir.  $y=x^2+2$ ,  $z=\sqrt[3]{x+5}$  fonksiyonları çizdirilecektir.

a) Fonksiyonları aynı şekil penceresinde aynı çizim eksen takımında kesikli çizgi ile farklı renklerde çiziniz.

```
x=1:0.1:20;
plot(x,x.^2+2,'- -'), hold on; plot(x,(x+5).^(1/3),'r- -');
```

b) Aynı şekil penceresinde ayrı eksen takımlarında alt alta istediğiniz gibi çiziniz.

```
figure
subplot(211); plot(x,x.^2+2,'- -'), subplot(212);
plot(x,(x+5).^(1/3),'r- -');
```

c) Ayrı iki şekil penceresinde çiziniz.

```
plot(x,x.^2+2,'- -'), figure, plot(x,(x+5).^(1/3),'r- -');
```

**SORU 6)** Başlangıçta tanımlı skaler s değerimizin o anki değerinin üstüne, giriş argümanı olarak verdiğimiz  $a_1, a_2, \dots, a_n$  değerlerini toplayan ve bunu farklı n değerleri için yapabilen myfun fonksiyonunu yazınız

```
function s=myfun(s,varargin)
for n=1:nargin-1
s=s+varargin{n};
end
```

**SORU 7)** Aşağıdaki kod myfctn.m dosyasında bulunmaktadır:

```
function myfctn(x)
```

```
global x; x=x-5; y = 0;
```

```
end
```

Komut satırında

```
>>x = 10; y = 1; global x; myfctn(x);
```

Yazdığımızda x = 5 y = 1 ne olur?

**SORU 8)**

```
for ii = 1:5
```

```
if ii == 3; continue; end
```

```
if ii>3; break; end
```

```
ii
```

```
end
```

```
disp('Bitti!');
```

**SORU 9)**

İki çıkış ve üç giriş argümanı olan hesapla adında bir fonksiyon yazılacaktır:

- 1) NARGOUT==0 Eğer fonksiyon çağrısında çıkış istenmezse fonksiyon hiçbir hesaplama yapmadan "çıkış isteği yok" yazdıracaktır.
- 2) NARGOUT==1 NARGIN==1 Eğer fonksiyon çağrısında tek çıkış istenir ve tek giriş varsa  $V = (4/3) \cdot \pi \cdot r^3$  ile küre hacmi döndürülecektir.
- 3) NARGOUT==1 ve NARGIN==3 Eğer fonksiyon çağrısında tek çıkış istenir ve üç giriş varsa  $C = s1 + s2 + s3$  ile üçgenin çevresi döndürülecektir.
- 4) NARGOUT==2 ve NARGIN==4 Eğer fonksiyon çağrısında iki çıkış istenirse ve 4 giriş varsa V ve C hesaplanır.

Tüm hesaplamalar için r değeri boş girildiğinde default değeri olan 0.5 alınmaktadır.

Bunlar dışında bir durumda "Bir hata olmalı!" mesajı ekrana yazdırılır.

```
function [V,C]=hesapla(r,s1,s2,s3)
```

```
if nargout == 0,
```

```
    disp('çıkış isteği yok'),
```

```
elseif nargout==1 & nargin==1,
```

```
    if isempty(r), r=0.5;end
```

```
    V=(4/3)*pi*r^3;
```

```
elseif nargout==1 & nargin==3,
```

```
    C=s1+s2+s3;
```

```
elseif nargout==2 & nargin==4,
```

```
    if isempty(r), r=0.5;end
```

```
    V=(4/3)*pi*r^3;
```

```
    C=s1+s2+s3;
```

```
else
```

```
    disp('Bir hata olmalı!')
```

```
end
```