

PAMUKKALE ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
CENG 241 BİLİMSEL HESAPLAMA DERSİ VİZE SINAVI

Soru1	Soru2	Soru3	Soru4	Soru5	TOPLAM
40	15	15	15	15	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 Kth 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.

isprime(x) true if x is prime else false

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,I]=sort(X) Return a copy of X with the elements arranged in increasing order.

M = mod(X, Y) Compute the modulo of X and Y.

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.

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) A=[16 2 3 13; 5 11 10 8; 9 7 6 12; 4 14 15 1]; N = 3; K = [4 5]; x=[1 9 3 6 2 4 5 8 7]; çıktıları yazınız.

a) `A([6 12 15])`

`ans = [11 15 12]`

b) `m = max(x(:))`

`m=9`

c) `n=length(A); A(:, [n-2+1:n 1:n-2])`

`ans =`

3	13	16	2
10	8	5	11
6	12	9	7
15	1	4	14

d) `y = (abs(x)>=5) .* x`

`y = [0 9 0 6 0 0 5 8 7]`

e) `y=x; I = find(abs(y)<4); y(I) = 0;`

`y = [0 9 0 6 0 4 5 8 7]`

f) `x(mod((1:end)+2-1, end)+1)`

`ans = [3 6 2 4 5 8 7 1 9]`

g) `[v,I]=sort(abs(x(:))); x(I(1:end-4))=0`

`x = [0 9 0 6 0 0 0 8 7]`

h) `out = K(ones(1,N),:); out = out(:)'`
`out = [4 4 4 5 5 5]`

i) `A(1:2,1:3) .* K'`

`ans =`

64	8	12
25	55	50

j) `diag(diag(A,2))`

`ans =`

3	0
0	8

k) `A(1 + A > 12 == int8(pi) / 3 | false)'`

`ans = [16 14 15 13 12]`

l) `A(end-1:-1:2,end-2:-1:1)`

`ans =`

7	9
11	5

n) `[C,D]=meshgrid([1 2],[3 4]); C+D`

`ans =`

4	5
5	6

o) `poly(x(3:4))`

`[1 -9 18]`

SORU 2) Bir $A = [16 \text{ NaN } 3 \text{ } 13; \text{ NaN } 11 \text{ } 10 \text{ } 8; 9 \text{ } 7 \text{ } 6 \text{ NaN}; 4 \text{ } 14 \text{ } 15 \text{ NaN}]$; matrisinde NaN değerlerini geçici olarak sıfırlarla değiştirmek, diğer elemanların 3 katını hesaplamak ve ardından NaN değerlerini tekrar orijinal konumlarına geri koymak istiyoruz. Bu örnekte, NaN bilgisini tutmak için bir (Konum_NaN) ve A matrisi dışında başka bir değişken tutulmayacaktır. Herhangi bir akış kontrolü komutu kullanılmayacaktır.

:: Önce konum_nan değişkeninde NaN konum bilgisini tutmamız ve buraya sonra NaN değerlerini geri atamamız gerekir.

```
>>konum_nan = find(isnan(A));  
>>A(konum_nan) = 0;  
>>A = A ./sum(A,2);  
>>A(konum_nan) = NaN;
```

SORU 3) Asal sayılar $\text{primes}(29)=[2 \ 3 \ 5 \ 7 \ 11 \ 13 \ 17 \ 19 \ 23 \ 29]$ örneğinde olduğu gibi düzensiz şekilde devam eder. Bir n sayısı asal değilse asal olmayan bu n sayısından önceki ilk asal sayıyı veren, n sayısı asal sayı ise asal olan bu n sayısından önceki asal sayıyı veren işlemi akış kontrolü kullanmadan yapınız.

Örnek girişler:

n=5 ise çıkış=3

n=6 ise çıkış=5

```
>> clear all
```

```
>> n= input('n') %n sayısı kullanıcı tarafından giriliyor.
```

"kullanıcı girişi"

CEVAP:

```
n = 150 girilmiş olsun; P = primes(n); output = P(end- 1*isprime(n) );
```

SORU 4): $y = 9x^4 - 6x^3 + 3x^2 - 4x + 2$ fonksiyonu anonim fonksiyon olarak elde ediniz. Daha sonra $-2 \leq x \leq 2$ aralığında 1000 nokta alarak çizdiriniz. Başlığa 'Sınav Sorusunun Cevabı', giriş değerleri eksenine ('giris degeri'), çıkış değerleri eksenine 'cikis degeri' yazdırınız.

```
>>clear
```

```
>>clc
```

```
>>myfun = @(x) 9*x.^4 - 6*x.^3 + 3*x.^2 - 4*x + 2;
```

```
>>x = linspace(-2,2, 1000);
```

```
>>y = myfun(x);
```

```
>>plot(x, y, 'r-'), xlabel('giris degeri'), ylabel('cikis degeri')
```

```
>>title('Sınav Sorusunun Cevabı')
```

SORU 5) Başlangıçta tanımlı skaler bir değerimizin o anki değerinin üstüne, ayrı birer giriş argümanı olarak verdiğimiz a_1, a_2, \dots, a_n değerlerini toplayan ve bunu farklı n değerleri için yapabilen myfun fonksiyonunu yazınız. ai değerlerinin bir vektör girişi olarak girilmesi kabul edilmeyecektir. Not: Sabit giriş argümanları girilecekse (burada olduğu gibi), önce sabit giriş argümanları girişi yapılması gerekir.

Örnek: $s=2 \rightarrow \text{out}=\text{myfun}(s,a_1,a_2) \rightarrow \text{out}=2+a_1+a_2$

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