

2_ncci_kod

March 28, 2024

```
[ ]: [a b] = pol2cart(53.1301*pi/180,5)
```

a =

3.0000

b =

4.0000

```
[ ]: x = [1 2 3 4 5]
```

```
f = 2*x + 4
```

```
f2 = 2*x.^2 + 4
```

x =

1 2 3 4 5

f =

6 8 10 12 14

f2 =

6 12 22 36 54

1. Soru

```
[ ]: x = -2:5
```

x =

-2	-1	0	1	2	3	4	5
----	----	---	---	---	---	---	---

```
[ ]: x = -2:0.5:5
```

x =

Columns 1 through 13

-2.0000	-1.5000	-1.0000	-0.5000		0	0.5000	1.0000	1.5000
2.0000	2.5000	3.0000	3.5000	4.0000				

Columns 14 through 15

4.5000	5.0000
--------	--------

```
[ ]: x = linspace(-2, 5, 250)
```

x =

Columns 1 through 13

-2.0000	-1.9719	-1.9438	-1.9157	-1.8876	-1.8594	-1.8313	-1.8032
-1.7751	-1.7470	-1.7189	-1.6908	-1.6627			

Columns 14 through 26

-1.6345	-1.6064	-1.5783	-1.5502	-1.5221	-1.4940	-1.4659	-1.4378
-1.4096	-1.3815	-1.3534	-1.3253	-1.2972			

Columns 27 through 39

-1.2691	-1.2410	-1.2129	-1.1847	-1.1566	-1.1285	-1.1004	-1.0723
-1.0442	-1.0161	-0.9880	-0.9598	-0.9317			

Columns 40 through 52

-0.9036	-0.8755	-0.8474	-0.8193	-0.7912	-0.7631	-0.7349	-0.7068
-0.6787	-0.6506	-0.6225	-0.5944	-0.5663			

Columns 53 through 65

-0.5382	-0.5100	-0.4819	-0.4538	-0.4257	-0.3976	-0.3695	-0.3414
-0.3133	-0.2851	-0.2570	-0.2289	-0.2008			

Columns 66 through 78

-0.1727	-0.1446	-0.1165	-0.0884	-0.0602	-0.0321	-0.0040	0.0241
0.0522	0.0803	0.1084	0.1365	0.1647			

Columns 79 through 91

0.1928	0.2209	0.2490	0.2771	0.3052	0.3333	0.3614	0.3896
0.4177	0.4458	0.4739	0.5020	0.5301			

Columns 92 through 104

0.5582	0.5863	0.6145	0.6426	0.6707	0.6988	0.7269	0.7550
0.7831	0.8112	0.8394	0.8675	0.8956			

Columns 105 through 117

0.9237	0.9518	0.9799	1.0080	1.0361	1.0643	1.0924	1.1205
1.1486	1.1767	1.2048	1.2329	1.2610			

Columns 118 through 130

1.2892	1.3173	1.3454	1.3735	1.4016	1.4297	1.4578	1.4859
1.5141	1.5422	1.5703	1.5984	1.6265			

Columns 131 through 143

1.6546	1.6827	1.7108	1.7390	1.7671	1.7952	1.8233	1.8514
1.8795	1.9076	1.9357	1.9639	1.9920			

Columns 144 through 156

2.0201	2.0482	2.0763	2.1044	2.1325	2.1606	2.1888	2.2169
2.2450	2.2731	2.3012	2.3293	2.3574			

Columns 157 through 169

2.3855	2.4137	2.4418	2.4699	2.4980	2.5261	2.5542	2.5823
2.6104	2.6386	2.6667	2.6948	2.7229			

Columns 170 through 182

2.7510	2.7791	2.8072	2.8353	2.8635	2.8916	2.9197	2.9478
2.9759	3.0040	3.0321	3.0602	3.0884			

Columns 183 through 195

3.1165	3.1446	3.1727	3.2008	3.2289	3.2570	3.2851	3.3133
3.3414	3.3695	3.3976	3.4257	3.4538			

Columns 196 through 208

3.4819	3.5100	3.5382	3.5663	3.5944	3.6225	3.6506	3.6787
3.7068	3.7349	3.7631	3.7912	3.8193			

Columns 209 through 221

3.8474	3.8755	3.9036	3.9317	3.9598	3.9880	4.0161	4.0442
4.0723	4.1004	4.1285	4.1566	4.1847			

Columns 222 through 234

4.2129	4.2410	4.2691	4.2972	4.3253	4.3534	4.3815	4.4096
4.4378	4.4659	4.4940	4.5221	4.5502			

Columns 235 through 247

4.5783	4.6064	4.6345	4.6627	4.6908	4.7189	4.7470	4.7751
4.8032	4.8313	4.8594	4.8876	4.9157			

Columns 248 through 250

4.9438	4.9719	5.0000
--------	--------	--------

```
[ ]: size(x)
```

ans =

```
1    250
```

```
[ ]: length(x)
```

ans =

```
250
```

```
[ ]: y = (((3*x.^2 - 4*x + 2).^2)/(20*x.^3))
```

y =

1.0512

```
[ ]: x = -2:5
```

x =

-2 -1 0 1 2 3 4 5

parantezleri kullanmak oldukça önemli doğru sonuçları bulmak için

```
[ ]: f = x./(4.*x)
```

f =

0.2500 0.2500 NaN 0.2500 0.2500 0.2500 0.2500 0.2500

```
[ ]: x.^x
```

ans =

1.0e+03 *

0.0003 -0.0010 0.0010 0.0010 0.0040 0.0270 0.2560 3.1250

```
[ ]:
```

Matlabda işlem Öncelikleri 1- Parantez 2- Kuvvet 3- Dört işlem 4- Fonksiyon

```
[ ]: A = [1 2 3 4; 5 6 7 8]
```

A =

1	2	3	4
5	6	7	8

```
[ ]: flipud(A)
      fliplr(A)
```

ans =

5	6	7	8
1	2	3	4

ans =

4	3	2	1
8	7	6	5

```
[ ]: B = rot90(A)
```

B =

4	8
3	7
2	6
1	5

```
[ ]: C = rot90(A,2)
```

C =

8	7	6	5
4	3	2	1

```
[ ]: D = repmat(A, 2)
```

D =

1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8

```
[ ]: E = repmat(A,2,3)
```

E =

1	2	3	4	1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8	5	6	7	8
1	2	3	4	1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8	5	6	7	8

```
[ ]: eye
```

ans =

1

```
[ ]: ones
```

ans =

1

```
[ ]: diag
```

Not enough input arguments.

```
[ ]: diag(2)
```

ans =

2

```
[ ]: diag(2,3)
```

ans =

0	0	0	2
0	0	0	0
0	0	0	0
0	0	0	0

```
[ ]: diag(2,3,4)
```

Too many input arguments.

```
[ ]: diag(0,4)
```

ans =

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

```
[ ]: B = diag([1 -2 4])
```

B =

1	0	0
0	-2	0
0	0	4

```
[ ]: B = diag([1 -2 4],2)
```

B =

0	0	1	0	0
0	0	0	-2	0
0	0	0	0	4

0	0	0	0	0
0	0	0	0	0

```
[ ]: B = diag([1 -2 4],-2)
```

B =

0	0	0	0	0
0	0	0	0	0
1	0	0	0	0
0	-2	0	0	0
0	0	4	0	0

```
[ ]: magic(3)
```

ans =

8	1	6
3	5	7
4	9	2

```
[ ]: sum(magic(4))
```

ans =

34	34	34	34
----	----	----	----

```
[ ]: magic(4)
```

ans =

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

```
[ ]: sum(magic(5))
```

ans =

65	65	65	65	65
----	----	----	----	----

```
[ ]: rand(3)
```

ans =

0.8147	0.9134	0.2785
0.9058	0.6324	0.5469
0.1270	0.0975	0.9575

```
[ ]: rand(2,3)
```

ans =

0.9649	0.9706	0.4854
0.1576	0.9572	0.8003

```
[ ]: rand(2,3)*25
```

ans =

3.5472	22.8934	23.9873
10.5440	19.8052	16.3935

```
[ ]: randn
```

ans =

0.4889

maksimum 50' 'ye kadar rastgele sayılar üretir

```
[ ]: randi(50)
```

ans =

33

```
[ ]: rand
```

ans =

0.0462

```
[ ]: perms([1 2 3])
```

ans =

3	2	1
3	1	2
2	3	1
2	1	3
1	3	2
1	2	3

```
[ ]: P = perms([1 2 3; 4 5 6])
```

P =

6	3	5	2	4	1
6	3	5	2	1	4
6	3	5	4	2	1
6	3	5	4	1	2
6	3	5	1	2	4
6	3	5	1	4	2
6	3	2	5	4	1
6	3	2	5	1	4
6	3	2	4	5	1
6	3	2	4	1	5
6	3	2	1	5	4
6	3	2	1	4	5

6	3	4	5	2	1
6	3	4	5	1	2
6	3	4	2	5	1
6	3	4	2	1	5
6	3	4	1	5	2
6	3	4	1	2	5
6	3	1	5	2	4
6	3	1	5	4	2
6	3	1	2	5	4
6	3	1	2	4	5
6	3	1	4	5	2
6	3	1	4	2	5
6	5	3	2	4	1
6	5	3	2	1	4
6	5	3	4	2	1
6	5	3	4	1	2
6	5	3	1	2	4
6	5	3	1	4	2
6	5	2	3	4	1
6	5	2	3	1	4
6	5	2	4	3	1
6	5	2	4	1	3
6	5	2	1	3	4
6	5	2	1	4	3
6	5	4	3	2	1
6	5	4	3	1	2
6	5	4	2	3	1
6	5	4	2	1	3
6	5	4	1	3	2
6	5	4	1	2	3
6	5	1	3	2	4
6	5	1	3	4	2
6	5	1	2	3	4
6	5	1	2	4	3
6	5	1	4	3	2
6	5	1	4	2	3
6	2	3	5	4	1
6	2	3	5	1	4
6	2	3	4	5	1
6	2	3	4	1	5
6	2	3	1	5	4
6	2	3	1	4	5
6	2	5	3	4	1
6	2	5	3	1	4
6	2	5	4	3	1
6	2	5	4	1	3
6	2	5	1	3	4
6	2	5	1	4	3

6	2	4	3	5	1
6	2	4	3	1	5
6	2	4	5	3	1
6	2	4	5	1	3
6	2	4	1	3	5
6	2	4	1	5	3
6	2	1	3	5	4
6	2	1	3	4	5
6	2	1	5	3	4
6	2	1	5	4	3
6	2	1	4	3	5
6	2	1	4	5	3
6	4	3	5	2	1
6	4	3	5	1	2
6	4	3	2	5	1
6	4	3	2	1	5
6	4	3	1	5	2
6	4	3	1	2	5
6	4	5	3	2	1
6	4	5	3	1	2
6	4	5	2	3	1
6	4	5	2	1	3
6	4	5	1	3	2
6	4	5	1	2	3
6	4	2	3	5	1
6	4	2	3	1	5
6	4	2	5	3	1
6	4	2	5	1	3
6	4	2	1	3	5
6	4	2	1	5	3
6	4	1	3	5	2
6	4	1	3	2	5
6	4	1	5	3	2
6	4	1	5	2	3
6	4	1	2	3	5
6	4	1	2	5	3
6	1	3	5	2	4
6	1	3	5	4	2
6	1	3	2	5	4
6	1	3	2	4	5
6	1	3	4	5	2
6	1	3	4	2	5
6	1	5	3	2	4
6	1	5	3	4	2
6	1	5	2	3	4
6	1	5	2	4	3
6	1	5	4	3	2
6	1	5	4	2	3

6	1	2	3	5	4
6	1	2	3	4	5
6	1	2	5	3	4
6	1	2	5	4	3
6	1	2	4	3	5
6	1	2	4	5	3
6	1	4	3	5	2
6	1	4	3	2	5
6	1	4	5	3	2
6	1	4	5	2	3
6	1	4	2	3	5
6	1	4	2	5	3
3	6	5	2	4	1
3	6	5	2	1	4
3	6	5	4	2	1
3	6	5	4	1	2
3	6	5	1	2	4
3	6	5	1	4	2
3	6	2	5	4	1
3	6	2	5	1	4
3	6	2	4	5	1
3	6	2	4	1	5
3	6	2	1	5	4
3	6	2	1	4	5
3	6	4	5	2	1
3	6	4	5	1	2
3	6	4	2	5	1
3	6	4	2	1	5
3	6	4	1	5	2
3	6	4	1	2	5
3	6	1	5	2	4
3	6	1	5	4	2
3	6	1	2	5	4
3	6	1	2	4	5
3	6	1	4	5	2
3	6	1	4	2	5
3	5	6	2	4	1
3	5	6	2	1	4
3	5	6	4	2	1
3	5	6	4	1	2
3	5	6	1	2	4
3	5	6	1	4	2
3	5	2	6	4	1
3	5	2	6	1	4
3	5	2	4	6	1
3	5	2	4	1	6
3	5	2	1	6	4
3	5	2	1	4	6

3	5	4	6	2	1
3	5	4	6	1	2
3	5	4	2	6	1
3	5	4	2	1	6
3	5	4	1	6	2
3	5	4	1	2	6
3	5	1	6	2	4
3	5	1	6	4	2
3	5	1	2	6	4
3	5	1	2	4	6
3	5	1	4	6	2
3	5	1	4	2	6
3	2	6	5	4	1
3	2	6	5	1	4
3	2	6	4	5	1
3	2	6	4	1	5
3	2	6	1	5	4
3	2	6	1	4	5
3	2	5	6	4	1
3	2	5	6	1	4
3	2	5	4	6	1
3	2	5	4	1	6
3	2	5	1	6	4
3	2	5	1	4	6
3	2	4	6	5	1
3	2	4	6	1	5
3	2	4	5	6	1
3	2	4	5	1	6
3	2	4	1	6	5
3	2	4	1	5	6
3	2	1	6	5	4
3	2	1	6	4	5
3	2	1	5	6	4
3	2	1	5	4	6
3	2	1	4	6	5
3	2	1	4	5	6
3	4	6	5	2	1
3	4	6	5	1	2
3	4	6	2	5	1
3	4	6	2	1	5
3	4	6	1	5	2
3	4	6	1	2	5
3	4	5	6	2	1
3	4	5	6	1	2
3	4	5	2	6	1
3	4	5	2	1	6
3	4	5	1	6	2
3	4	5	1	2	6

3	4	2	6	5	1
3	4	2	6	1	5
3	4	2	5	6	1
3	4	2	5	1	6
3	4	2	1	6	5
3	4	2	1	5	6
3	4	1	6	5	2
3	4	1	6	2	5
3	4	1	5	6	2
3	4	1	5	2	6
3	4	1	2	6	5
3	4	1	2	5	6
3	1	6	5	2	4
3	1	6	5	4	2
3	1	6	2	5	4
3	1	6	2	4	5
3	1	6	4	5	2
3	1	6	4	2	5
3	1	5	6	2	4
3	1	5	6	4	2
3	1	5	2	6	4
3	1	5	2	4	6
3	1	5	4	6	2
3	1	5	4	2	6
3	1	2	6	5	4
3	1	2	6	4	5
3	1	2	5	6	4
3	1	2	5	4	6
3	1	2	4	6	5
3	1	2	4	5	6
3	1	4	6	5	2
3	1	4	6	2	5
3	1	4	5	6	2
3	1	4	5	2	6
3	1	4	2	6	5
3	1	4	2	5	6
5	6	3	2	4	1
5	6	3	2	1	4
5	6	3	4	2	1
5	6	3	4	1	2
5	6	3	1	2	4
5	6	3	1	4	2
5	6	2	3	4	1
5	6	2	3	1	4
5	6	2	4	3	1
5	6	2	4	1	3
5	6	2	1	3	4
5	6	2	1	4	3

5	6	4	3	2	1
5	6	4	3	1	2
5	6	4	2	3	1
5	6	4	2	1	3
5	6	4	1	3	2
5	6	4	1	2	3
5	6	1	3	2	4
5	6	1	3	4	2
5	6	1	2	3	4
5	6	1	2	4	3
5	6	1	4	3	2
5	6	1	4	2	3
5	3	6	2	4	1
5	3	6	2	1	4
5	3	6	4	2	1
5	3	6	4	1	2
5	3	6	1	2	4
5	3	6	1	4	2
5	3	2	6	4	1
5	3	2	6	1	4
5	3	2	4	6	1
5	3	2	4	1	6
5	3	2	1	6	4
5	3	2	1	4	6
5	3	4	6	2	1
5	3	4	6	1	2
5	3	4	2	6	1
5	3	4	2	1	6
5	3	4	1	6	2
5	3	4	1	2	6
5	3	1	6	2	4
5	3	1	6	4	2
5	3	1	2	6	4
5	3	1	2	4	6
5	3	1	4	6	2
5	3	1	4	2	6
5	2	6	3	4	1
5	2	6	3	1	4
5	2	6	4	3	1
5	2	6	4	1	3
5	2	6	1	3	4
5	2	6	1	4	3
5	2	3	6	4	1
5	2	3	6	1	4
5	2	3	4	6	1
5	2	3	4	1	6
5	2	3	1	6	4
5	2	3	1	4	6

5	2	4	6	3	1
5	2	4	6	1	3
5	2	4	3	6	1
5	2	4	3	1	6
5	2	4	1	6	3
5	2	4	1	3	6
5	2	1	6	3	4
5	2	1	6	4	3
5	2	1	3	6	4
5	2	1	3	4	6
5	2	1	4	6	3
5	2	1	4	3	6
5	4	6	3	2	1
5	4	6	3	1	2
5	4	6	2	3	1
5	4	6	2	1	3
5	4	6	1	3	2
5	4	6	1	2	3
5	4	3	6	2	1
5	4	3	6	1	2
5	4	3	2	6	1
5	4	3	2	1	6
5	4	3	1	6	2
5	4	3	1	2	6
5	4	2	6	3	1
5	4	2	6	1	3
5	4	2	3	6	1
5	4	2	3	1	6
5	4	2	1	6	3
5	4	2	1	3	6
5	4	1	6	3	2
5	4	1	6	2	3
5	4	1	3	6	2
5	4	1	3	2	6
5	4	1	2	6	3
5	4	1	2	3	6
5	1	6	3	2	4
5	1	6	3	4	2
5	1	6	2	3	4
5	1	6	2	4	3
5	1	6	4	3	2
5	1	6	4	2	3
5	1	3	6	2	4
5	1	3	6	4	2
5	1	3	2	6	4
5	1	3	2	4	6
5	1	3	4	6	2
5	1	3	4	2	6

5	1	2	6	3	4
5	1	2	6	4	3
5	1	2	3	6	4
5	1	2	3	4	6
5	1	2	4	6	3
5	1	2	4	3	6
5	1	4	6	3	2
5	1	4	6	2	3
5	1	4	3	6	2
5	1	4	3	2	6
5	1	4	2	6	3
5	1	4	2	3	6
2	6	3	5	4	1
2	6	3	5	1	4
2	6	3	4	5	1
2	6	3	4	1	5
2	6	3	1	5	4
2	6	3	1	4	5
2	6	5	3	4	1
2	6	5	3	1	4
2	6	5	4	3	1
2	6	5	4	1	3
2	6	5	1	3	4
2	6	5	1	4	3
2	6	4	3	5	1
2	6	4	3	1	5
2	6	4	5	3	1
2	6	4	5	1	3
2	6	4	1	3	5
2	6	4	1	5	3
2	6	1	3	5	4
2	6	1	3	4	5
2	6	1	5	3	4
2	6	1	5	4	3
2	6	1	4	3	5
2	6	1	4	5	3
2	3	6	5	4	1
2	3	6	5	1	4
2	3	6	4	5	1
2	3	6	4	1	5
2	3	6	1	5	4
2	3	6	1	4	5
2	3	5	6	4	1
2	3	5	6	1	4
2	3	5	4	6	1
2	3	5	4	1	6
2	3	5	1	6	4
2	3	5	1	4	6

2	3	4	6	5	1
2	3	4	6	1	5
2	3	4	5	6	1
2	3	4	5	1	6
2	3	4	1	6	5
2	3	4	1	5	6
2	3	1	6	5	4
2	3	1	6	4	5
2	3	1	5	6	4
2	3	1	5	4	6
2	3	1	4	6	5
2	3	1	4	5	6
2	5	6	3	4	1
2	5	6	3	1	4
2	5	6	4	3	1
2	5	6	4	1	3
2	5	6	1	3	4
2	5	6	1	4	3
2	5	3	6	4	1
2	5	3	6	1	4
2	5	3	4	6	1
2	5	3	4	1	6
2	5	3	1	6	4
2	5	3	1	4	6
2	5	4	6	3	1
2	5	4	6	1	3
2	5	4	3	6	1
2	5	4	3	1	6
2	5	4	1	6	3
2	5	4	1	3	6
2	5	1	6	3	4
2	5	1	6	4	3
2	5	1	3	6	4
2	5	1	3	4	6
2	5	1	4	6	3
2	5	1	4	3	6
2	4	6	3	5	1
2	4	6	3	1	5
2	4	6	5	3	1
2	4	6	5	1	3
2	4	6	1	3	5
2	4	6	1	5	3
2	4	3	6	5	1
2	4	3	6	1	5
2	4	3	5	6	1
2	4	3	5	1	6
2	4	3	1	6	5
2	4	3	1	5	6

2	4	5	6	3	1
2	4	5	6	1	3
2	4	5	3	6	1
2	4	5	3	1	6
2	4	5	1	6	3
2	4	5	1	3	6
2	4	1	6	3	5
2	4	1	6	5	3
2	4	1	3	6	5
2	4	1	3	5	6
2	4	1	5	6	3
2	4	1	5	3	6
2	1	6	3	5	4
2	1	6	3	4	5
2	1	6	5	3	4
2	1	6	5	4	3
2	1	6	4	3	5
2	1	6	4	5	3
2	1	3	6	5	4
2	1	3	6	4	5
2	1	3	5	6	4
2	1	3	5	4	6
2	1	3	4	6	5
2	1	3	4	5	6
2	1	5	6	3	4
2	1	5	6	4	3
2	1	5	3	6	4
2	1	5	3	4	6
2	1	5	4	6	3
2	1	5	4	3	6
2	1	4	6	3	5
2	1	4	6	5	3
2	1	4	3	6	5
2	1	4	3	5	6
2	1	4	5	6	3
2	1	4	5	3	6
4	6	3	5	2	1
4	6	3	5	1	2
4	6	3	2	5	1
4	6	3	2	1	5
4	6	3	1	5	2
4	6	3	1	2	5
4	6	5	3	2	1
4	6	5	3	1	2
4	6	5	2	3	1
4	6	5	2	1	3
4	6	5	1	3	2
4	6	5	1	2	3

4	6	2	3	5	1
4	6	2	3	1	5
4	6	2	5	3	1
4	6	2	5	1	3
4	6	2	1	3	5
4	6	2	1	5	3
4	6	1	3	5	2
4	6	1	3	2	5
4	6	1	5	3	2
4	6	1	5	2	3
4	6	1	2	3	5
4	6	1	2	5	3
4	3	6	5	2	1
4	3	6	5	1	2
4	3	6	2	5	1
4	3	6	2	1	5
4	3	6	1	5	2
4	3	6	1	2	5
4	3	5	6	2	1
4	3	5	6	1	2
4	3	5	2	6	1
4	3	5	2	1	6
4	3	5	1	6	2
4	3	5	1	2	6
4	3	2	6	5	1
4	3	2	6	1	5
4	3	2	5	6	1
4	3	2	5	1	6
4	3	2	1	6	5
4	3	2	1	5	6
4	3	1	6	5	2
4	3	1	6	2	5
4	3	1	5	6	2
4	3	1	5	2	6
4	3	1	2	6	5
4	3	1	2	5	6
4	5	6	3	2	1
4	5	6	3	1	2
4	5	6	2	3	1
4	5	6	2	1	3
4	5	6	1	3	2
4	5	6	1	2	3
4	5	3	6	2	1
4	5	3	6	1	2
4	5	3	2	6	1
4	5	3	2	1	6
4	5	3	1	6	2
4	5	3	1	2	6

4	5	2	6	3	1
4	5	2	6	1	3
4	5	2	3	6	1
4	5	2	3	1	6
4	5	2	1	6	3
4	5	2	1	3	6
4	5	1	6	3	2
4	5	1	6	2	3
4	5	1	3	6	2
4	5	1	3	2	6
4	5	1	2	6	3
4	5	1	2	3	6
4	2	6	3	5	1
4	2	6	3	1	5
4	2	6	5	3	1
4	2	6	5	1	3
4	2	6	1	3	5
4	2	6	1	5	3
4	2	3	6	5	1
4	2	3	6	1	5
4	2	3	5	6	1
4	2	3	5	1	6
4	2	3	1	6	5
4	2	3	1	5	6
4	2	5	6	3	1
4	2	5	6	1	3
4	2	5	3	6	1
4	2	5	3	1	6
4	2	5	1	6	3
4	2	5	1	3	6
4	2	1	6	3	5
4	2	1	6	5	3
4	2	1	3	6	5
4	2	1	3	5	6
4	2	1	5	6	3
4	2	1	5	3	6
4	1	6	3	5	2
4	1	6	3	2	5
4	1	6	5	3	2
4	1	6	5	2	3
4	1	6	2	3	5
4	1	6	2	5	3
4	1	3	6	5	2
4	1	3	6	2	5
4	1	3	5	6	2
4	1	3	5	2	6
4	1	3	2	6	5
4	1	3	2	5	6

4	1	5	6	3	2
4	1	5	6	2	3
4	1	5	3	6	2
4	1	5	3	2	6
4	1	5	2	6	3
4	1	5	2	3	6
4	1	2	6	3	5
4	1	2	6	5	3
4	1	2	3	6	5
4	1	2	3	5	6
4	1	2	5	6	3
4	1	2	5	3	6
1	6	3	5	2	4
1	6	3	5	4	2
1	6	3	2	5	4
1	6	3	2	4	5
1	6	3	4	5	2
1	6	3	4	2	5
1	6	5	3	2	4
1	6	5	3	4	2
1	6	5	2	3	4
1	6	5	2	4	3
1	6	5	4	3	2
1	6	5	4	2	3
1	6	2	3	5	4
1	6	2	3	4	5
1	6	2	5	3	4
1	6	2	5	4	3
1	6	2	4	3	5
1	6	2	4	5	3
1	6	4	3	5	2
1	6	4	3	2	5
1	6	4	5	3	2
1	6	4	5	2	3
1	6	4	2	3	5
1	6	4	2	5	3
1	3	6	5	2	4
1	3	6	5	4	2
1	3	6	2	5	4
1	3	6	2	4	5
1	3	6	4	5	2
1	3	6	4	2	5
1	3	5	6	2	4
1	3	5	6	4	2
1	3	5	2	6	4
1	3	5	2	4	6
1	3	5	4	6	2
1	3	5	4	2	6

1	3	2	6	5	4
1	3	2	6	4	5
1	3	2	5	6	4
1	3	2	5	4	6
1	3	2	4	6	5
1	3	2	4	5	6
1	3	4	6	5	2
1	3	4	6	2	5
1	3	4	5	6	2
1	3	4	5	2	6
1	3	4	2	6	5
1	3	4	2	5	6
1	5	6	3	2	4
1	5	6	3	4	2
1	5	6	2	3	4
1	5	6	2	4	3
1	5	6	4	3	2
1	5	6	4	2	3
1	5	3	6	2	4
1	5	3	6	4	2
1	5	3	2	6	4
1	5	3	2	4	6
1	5	3	4	6	2
1	5	3	4	2	6
1	5	2	6	3	4
1	5	2	6	4	3
1	5	2	3	6	4
1	5	2	3	4	6
1	5	2	4	6	3
1	5	2	4	3	6
1	5	4	6	3	2
1	5	4	6	2	3
1	5	4	3	6	2
1	5	4	3	2	6
1	5	4	2	6	3
1	5	4	2	3	6
1	2	6	3	5	4
1	2	6	3	4	5
1	2	6	5	3	4
1	2	6	5	4	3
1	2	6	4	3	5
1	2	6	4	5	3
1	2	3	6	5	4
1	2	3	6	4	5
1	2	3	5	6	4
1	2	3	5	4	6
1	2	3	4	6	5
1	2	3	4	5	6

1	2	5	6	3	4
1	2	5	6	4	3
1	2	5	3	6	4
1	2	5	3	4	6
1	2	5	4	6	3
1	2	5	4	3	6
1	2	4	6	3	5
1	2	4	6	5	3
1	2	4	3	6	5
1	2	4	3	5	6
1	2	4	5	6	3
1	2	4	5	3	6
1	4	6	3	5	2
1	4	6	3	2	5
1	4	6	5	3	2
1	4	6	5	2	3
1	4	6	2	3	5
1	4	6	2	5	3
1	4	3	6	5	2
1	4	3	6	2	5
1	4	3	5	6	2
1	4	3	5	2	6
1	4	3	2	6	5
1	4	3	2	5	6
1	4	5	6	3	2
1	4	5	6	2	3
1	4	5	3	6	2
1	4	5	3	2	6
1	4	5	2	6	3
1	4	5	2	3	6
1	4	2	6	3	5
1	4	2	6	5	3
1	4	2	3	6	5
1	4	2	3	5	6
1	4	2	5	6	3
1	4	2	5	3	6

```
[ ]: size(P)
```

```
ans =
```

```
720    6
```

```
randperm(max,sayı miktarı)
```

```
[ ]: randperm(4,2)
```

ans =

```
     2     3
```

```
[ ]: sum([1 3 4])
```

ans =

```
     8
```

```
[ ]: sum([1;3;4])
```

ans =

```
     8
```

```
[ ]: A = [1 3 5; 2 4 6]
```

A =

```
     1     3     5
     2     4     6
```

```
[ ]: sum(A)
```

ans =

```
     3     7    11
```

```
[ ]: sort(A)
```

ans =

```
1 3 5
2 4 6
```

```
[ ]: sort([-2 10 0 6])
```

```
ans =
```

```
-2 0 6 10
```

```
[ ]: max(A)
```

```
ans =
```

```
2 4 6
```

```
[ ]: min(A)
```

```
ans =
```

```
1 3 5
```

```
[ ]: C = [1 2 3]
sum(C)/numel(C)
```

```
C =
```

```
1 2 3
```

```
ans =
```

```
2
```

```
[ ]: rank(C)
```

```
ans =
```

```
1
```

```
[ ]: rank(A)
```

```
ans =
```

```
2
```

```
[ ]: rank(B)
```

```
Unrecognized function or variable 'B'.
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```