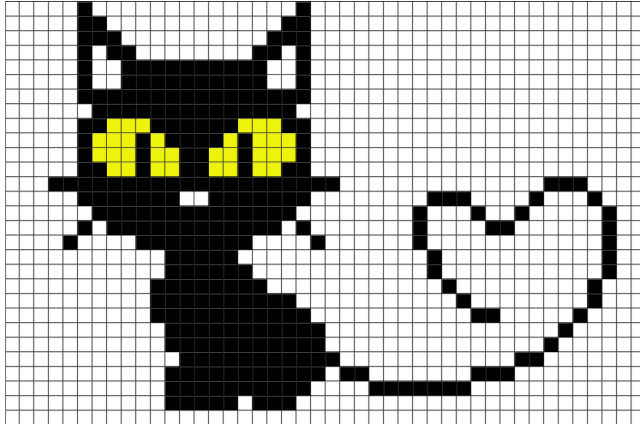


iki Boyutlu Diziler Matrisler



Fidan KAYA GÜLAĞIZ
Onur GÖK

Matrix

| | | |
|---|---|---|
| 0 | 1 | 2 |
|---|---|---|

| | | |
|---|---|---|
| 3 | 4 | 5 |
|---|---|---|

| | | |
|---|---|---|
| 6 | 7 | 8 |
|---|---|---|

| | | |
|---|---|---|
| 0 | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

Matris

isim
↑
int **matris**[4][3]={35,33,42,10,14,19,27,44,26,31,35,33};
↑ ↙
tip boyut

```
int matris[M][N]  
matris[1][2]  
]
```

| | | | |
|----------|----|----|----|
| 1. satır | 35 | 33 | 42 |
| 2. satır | 10 | 14 | 19 |
| 3. satır | 27 | 44 | 26 |
| 4. satır | 31 | 35 | 33 |

| | icerik | adress |
|----------|--------|--------|
| | ... | ... |
| | | 0F1C |
| 1. satır | 35 | 0F20 |
| | 33 | 0F24 |
| | 42 | 0F28 |
| | 10 | 0F2C |
| 2. satır | 14 | 0F30 |
| | 19 | 0F34 |
| | 27 | 0F38 |
| 3. satır | 44 | 0F3C |
| | 26 | 0F40 |
| | 31 | 0F44 |
| 4. satır | 35 | 0F48 |
| | 33 | 0F4C |
| | ... | ... |

Matris

```
#include <stdio.h>
```

```
int main() {
```

```
    int matris[3][3] = { {0, 1, 2}, {3, 4, 5}, {6, 7, 8} };
```

```
    int satir, sutun;
```

```
    for (satir = 0 ; satir < 3 ; satir++) {
```

```
        for (sutun = 0 ; sutun < 3 ; sutun++)
```

```
            printf("%d ", matris[satir][sutun]);
```

```
        printf("\n");
```

```
    }
```

```
    printf("\n");
```

```
    matris[1][2] = 999;
```

```
    printf("0,2 -> %d\n\n", matris[0][2]);
```

```
    printf("2,0. elemana sayi girin:");
```

```
    scanf("%d", &matris[2][0]);
```

```
    printf("2,0 -> %d\n\n", matris[2][0]);
```

```
    for (satir = 0 ; satir < 3 ; satir++) {
```

```
        for (sutun = 0 ; sutun < 3 ; sutun++)
```

```
            printf("%d ", matris[satir][sutun]);
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

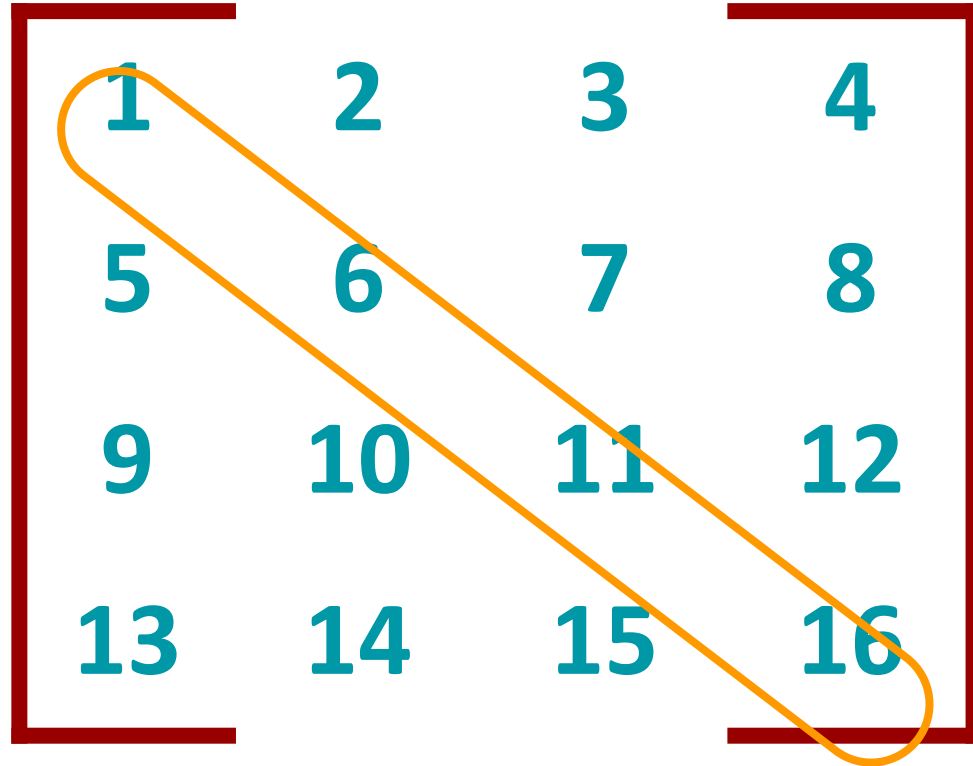
```
}
```

| | | |
|---|---|---|
| 0 | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

Matris Kösegeni

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

Matris Kösegeni

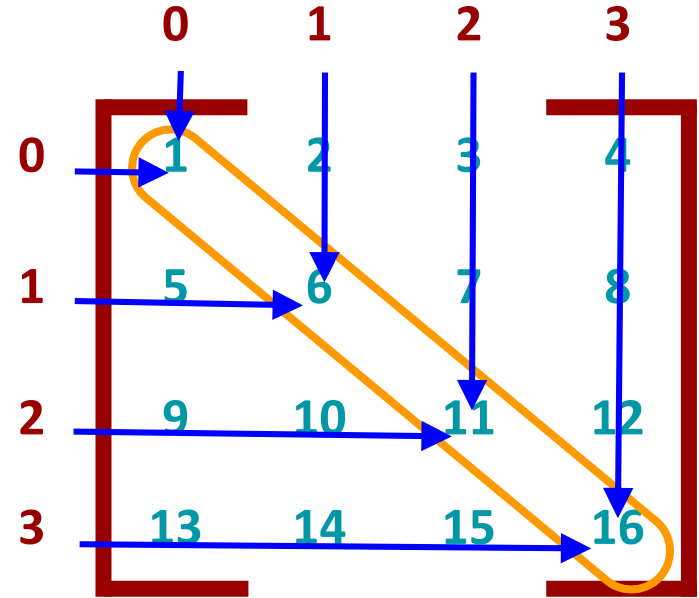


| | | | |
|----|----|----|----|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

Matris Kösegeni

```
#include <stdio.h>

int main() {
    int N = 4;
    int matris[4][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15, 16}};
    int i, j;
    printf("matris:\n");
    for (i = 0 ; i < N ; i++) {
        for (j = 0 ; j < N ; j++) {
            printf("%2d ", matris[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    printf("matrisin kosegeni: ");
    for (i = 0 ; i < N ; i++) {
        for (j = 0 ; j < N ; j++) {
            if (i == j)
                printf("%d ", matris[i][j]);
        }
    }
    printf("\n");
    return 0;
}
```

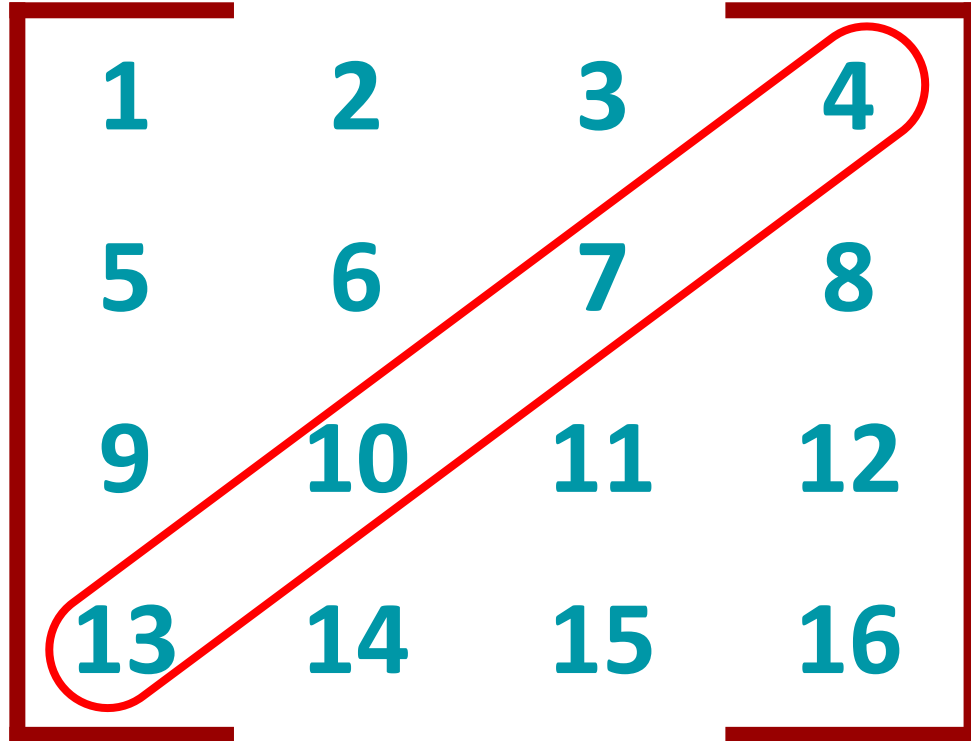


Matris Kösegeni

```
#include <stdio.h>
int main() {
    int N = 4;
    int matris[4][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15, 16}};
    int i, j;
    printf("matris:\n");
    for (i = 0 ; i < N ; i++) {
        for (j = 0 ; j < N ; j++) {
            printf("%2d ", matris[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    printf("matrisin kosegeni: ");
    for (i = 0 ; i < N ; i++) {
        printf("%d ", matris[i][i]);
    }
    printf("\n");
    return 0;
}
```

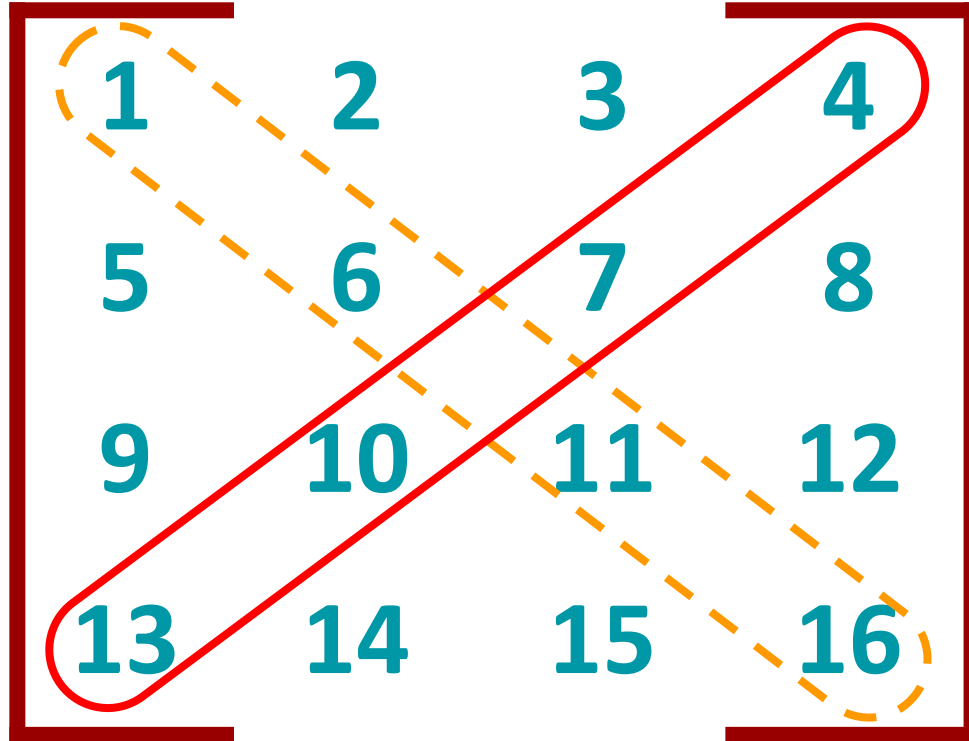
| | | | |
|----|----|----|----|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

Matris Ters Köşegeni



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

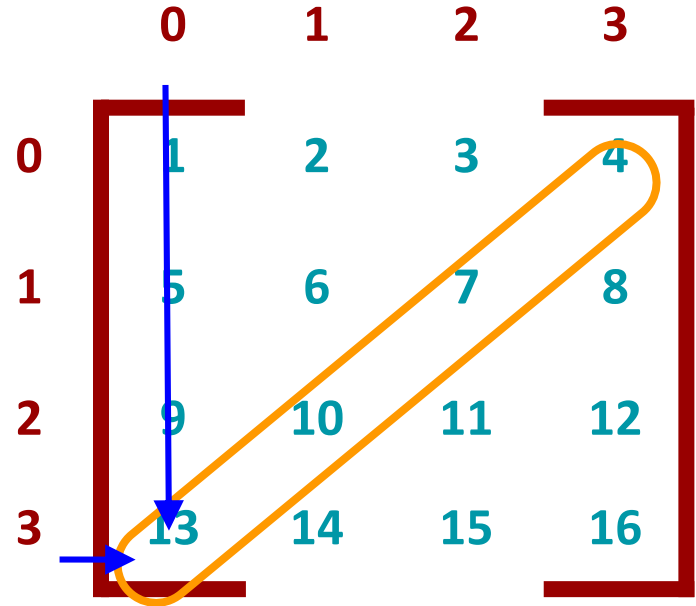
Matris Ters Köşegeni



Matris Ters Kösegeni

```
#include <stdio.h>

int main() {
    int N = 4;
    int matris[4][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15, 16}};
    int i, j;
    printf("matris:\n");
    for (i = 0 ; i < N ; i++) {
        for (j = 0 ; j < N ; j++) {
            printf("%2d ", matris[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    printf("matrisin ters kosegeni: ");
    for (i = 0 ; i < N ; i++) {
        for (j = 0 ; j < N ; j++) {
            if (i == N-1-j)
                printf("%d ", matris[i][j]);
        }
    }
    printf("\n");
    return 0;
}
```



Matris Ters Kösegeni

```
#include <stdio.h>
int main() {
    int N = 4;
    int matris[4][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15, 16}};
    int i, j;
    printf("matris:\n");
    for (i = 0 ; i < N ; i++) {
        for (j = 0 ; j < N ; j++) {
            printf("%2d ", matris[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    printf("matrisin ters kosegeni: ");
    for (i = 0 ; i < N ; i++) {
        printf("%d ", matris[i][N-1-i]);
    }
    printf("\n");
    return 0;
}
```

| | 0 | 1 | 2 | 3 |
|---|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 |
| 1 | 5 | 6 | 7 | 8 |
| 2 | 9 | 10 | 11 | 12 |
| 3 | 13 | 14 | 15 | 16 |

Matris transpozu alma

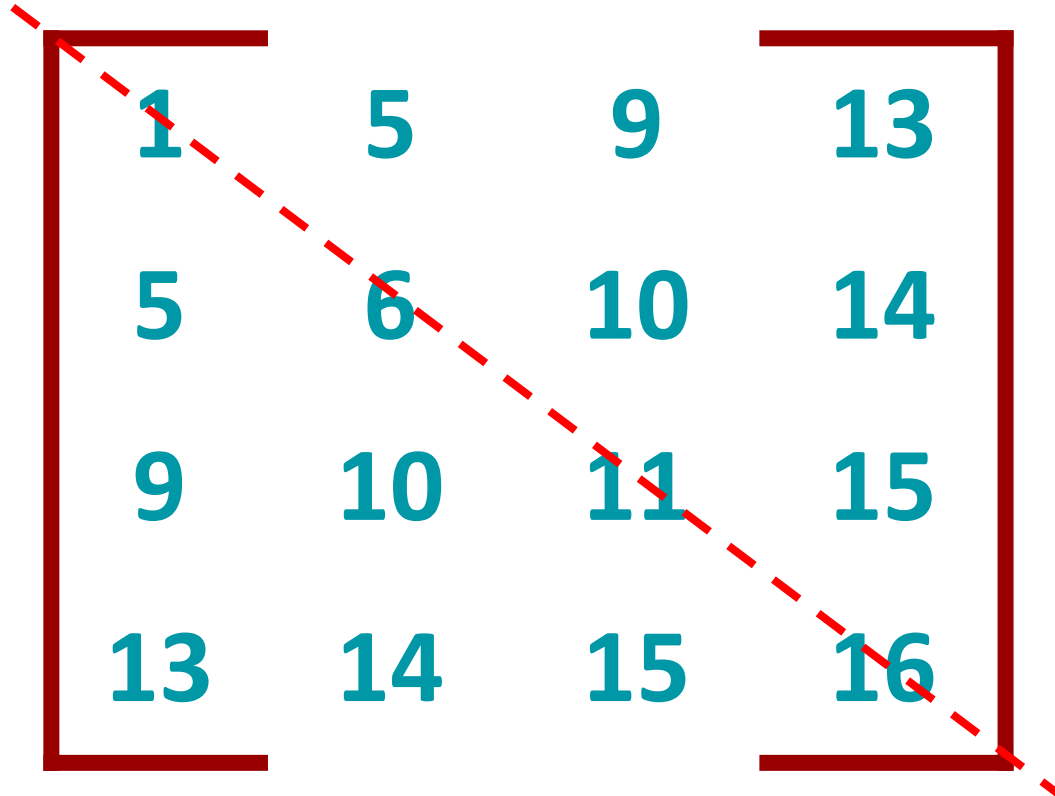
```
#include <stdio.h>

int main() {
    int matris[3][3];
    int transpose[3][3];
    int satir = 0, sutun = 0;
    printf("\n3*3'luk matrisin degerlerini giriniz \n\n");
    for( satir = 0 ; satir < 3; satir++ ) {
        for ( sutun = 0; sutun < 3; sutun++ ) {
            printf("Matrisin [%d][%d] elemani : ", satir+1, sutun+1);
            scanf("%d", &matris[satir][sutun]);
            transpose[sutun][satir] = matris[satir][sutun];
        }
    }
    printf("\nMatris\tve\tTranspozu\n");
    for( satir= 0 ; satir < 3; satir++ ) {
        for ( sutun = 0; sutun < 3; sutun++ ) {
            printf("%d ", matris[satir][sutun]);
        }
        printf("\t\t"); // 2 tab boyutu bosluk birakir
        for ( sutun = 0; sutun < 3; sutun++ ) {
            printf("%d ", transpose[satir][sutun]);
        }
        printf("\n");
    }
    return 0;
}
```

Simetrik Matriks

| | | | |
|----|----|----|----|
| 1 | 5 | 9 | 13 |
| 5 | 6 | 10 | 14 |
| 9 | 10 | 11 | 15 |
| 13 | 14 | 15 | 16 |

Simetrik Matriks



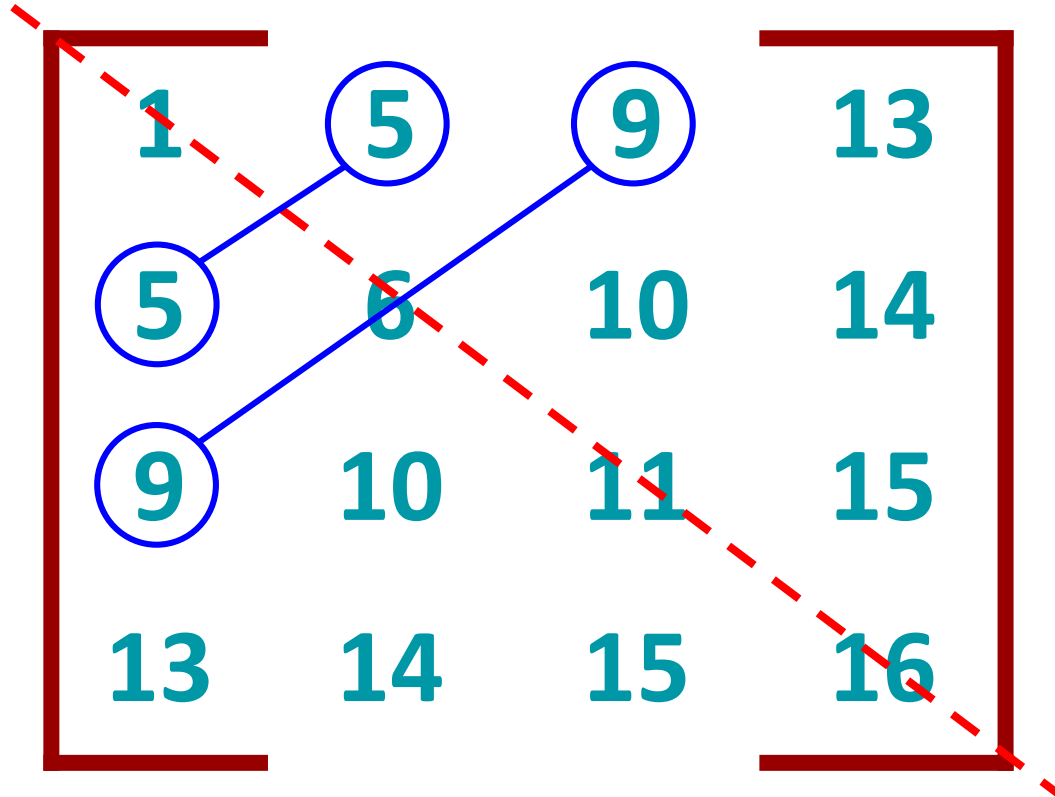
A 4x4 symmetric matrix is displayed, enclosed in a dark red border. The matrix elements are teal-colored numbers. A dashed red line runs diagonally from the top-left to the bottom-right, highlighting the symmetry of the matrix (where $a_{ij} = a_{ji}$).

| | | | |
|----|----|----|----|
| 1 | 5 | 9 | 13 |
| 5 | 6 | 10 | 14 |
| 9 | 10 | 11 | 15 |
| 13 | 14 | 15 | 16 |

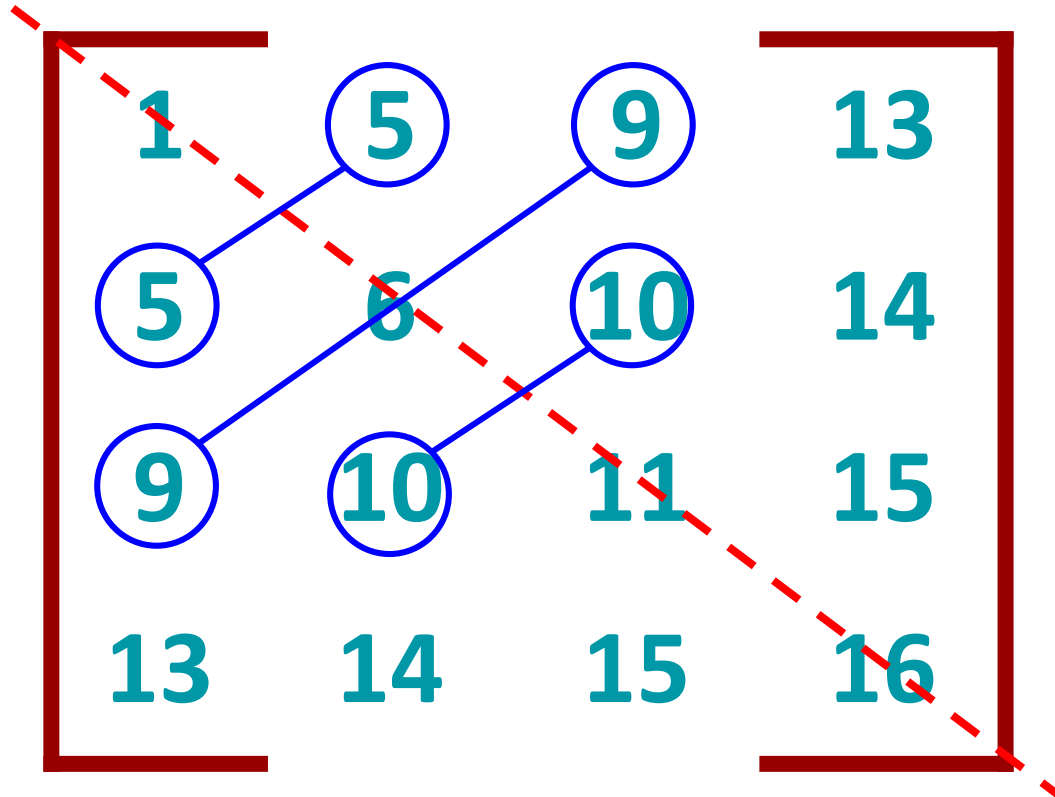
Simetrik Matris

| | | | |
|----|----|----|----|
| 1 | 5 | 9 | 13 |
| 5 | 6 | 10 | 14 |
| 9 | 10 | 11 | 15 |
| 13 | 14 | 15 | 16 |

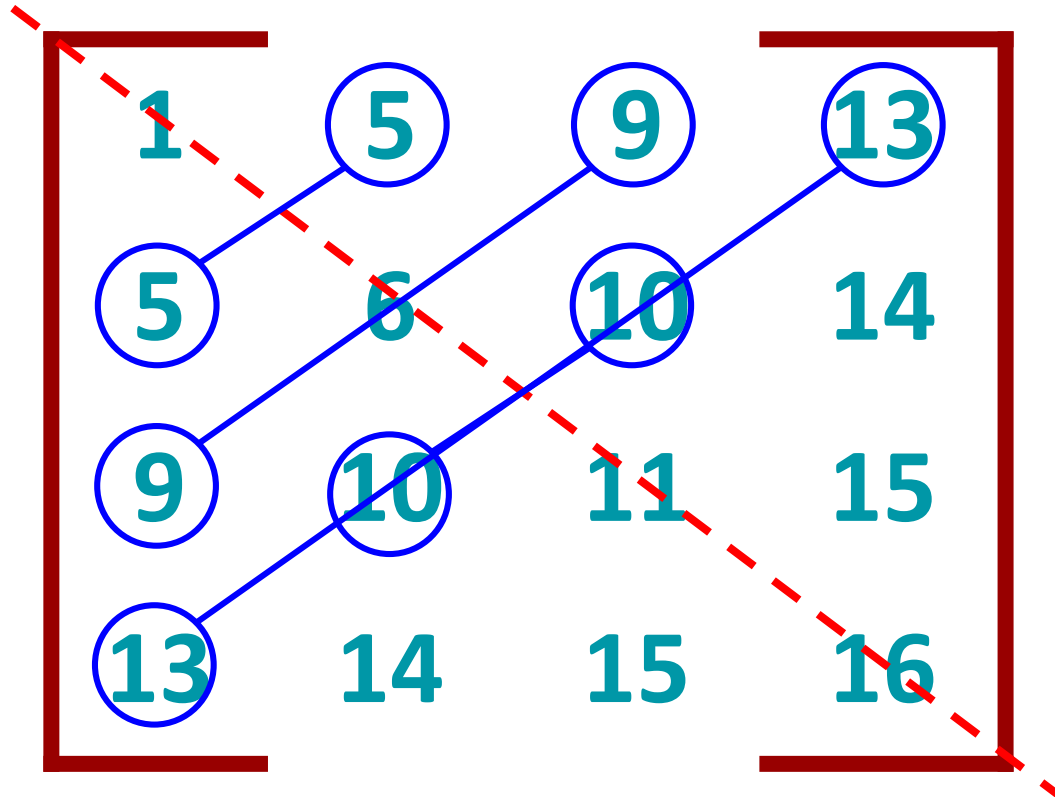
Simetrik Matriks



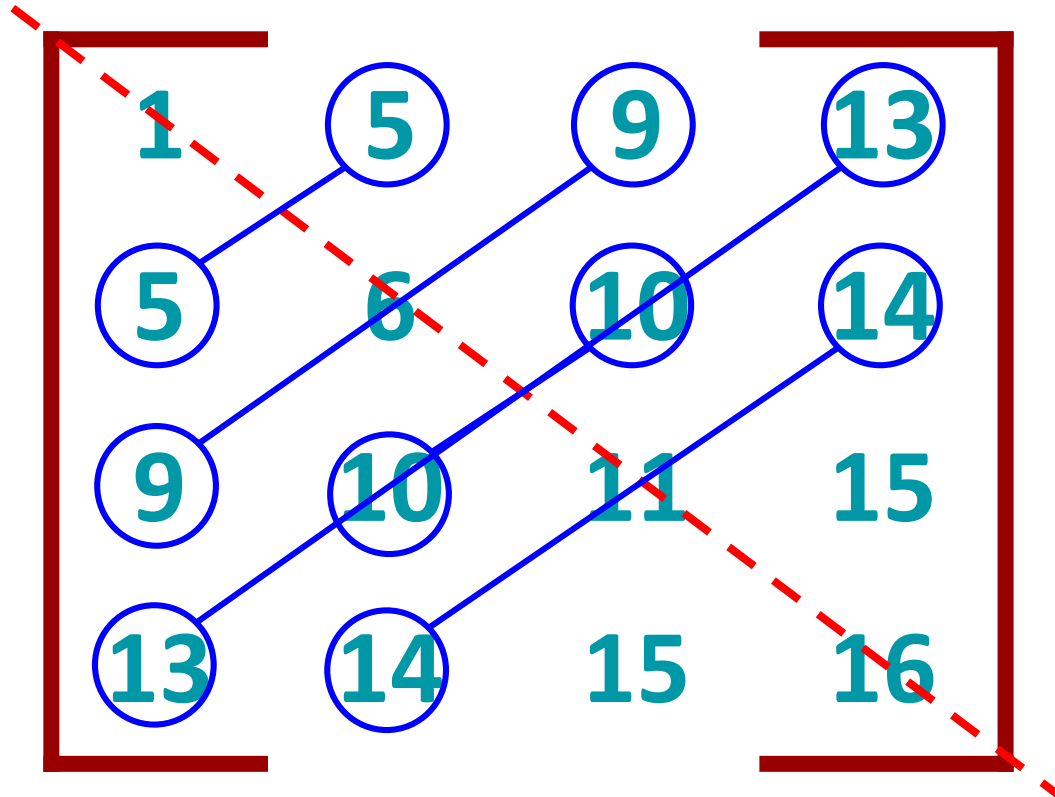
Simetrik Matriks



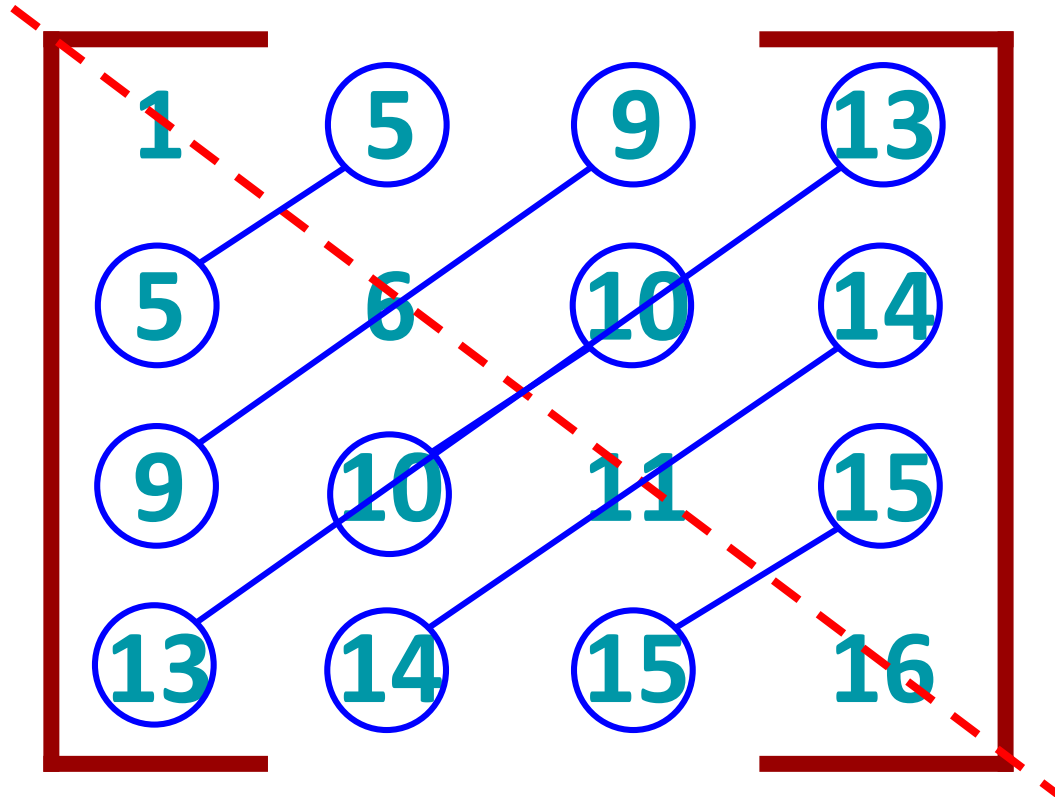
Simetrik Matriks



Simetrik Matris



Simetrik Matriks



Simetrik Matriks

$$A=A^T$$

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

\Rightarrow

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

```

#include <stdio.h>
int simetrik_mi_yontem_1(int matris[4][4]) {
    int i,j;
    for (i = 0 ; i < 4 ; i++) {
        for (j = 0 ; j < 4 ; j++) {
            if (matris[i][j] != matris[j][i]) {
                return 0;
            }
        }
    }
    return 1;
}

int simetrik_mi_yontem_2(int matris[4][4]) {
    int i,j;
    // kontrolu sadece ust ucgeni dolayarak yap
    for (i = 0 ; i < 4 ; i++) {
        for (j = i+1 ; j < 4 ; j++) {
            if (matris[i][j] != matris[j][i]) {
                return 0;
            }
        }
    }
    return 1;
}

void main() {
    int m1[4][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15, 16}};
    int m2[4][4] = { {1, 2, 3, 4}, {2, 6, 7, 8}, {3, 7, 11, 12}, {4, 8, 12, 16}};
    if (simetrik_mi_yontem_1(m1) == 1) printf("m1 simetriktir\n");
    else printf("m1 simetrik degildir\n");
    if (simetrik_mi_yontem_1(m2) == 1) printf("m2 simetriktir\n");
    else printf("m2 simetrik degildir\n");
    if (simetrik_mi_yontem_2(m1) == 1) printf("m1 simetriktir\n");
    else printf("m1 simetrik degildir\n");
    if (simetrik_mi_yontem_2(m2) == 1) printf("m2 simetriktir\n");
    else printf("m2 simetrik degildir\n");
}

```

Simetrik mi?

m1

| | 0 | 1 | 2 | 3 |
|---|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 |
| 1 | 5 | 6 | 7 | 8 |
| 2 | 9 | 10 | 11 | 12 |
| 3 | 13 | 14 | 15 | 16 |

m2

| | 0 | 1 | 2 | 3 |
|---|---|---|----|----|
| 0 | 1 | 2 | 3 | 4 |
| 1 | 2 | 6 | 7 | 8 |
| 2 | 3 | 7 | 11 | 12 |
| 3 | 4 | 8 | 12 | 16 |

Matriste Arama

```
#include <stdio.h>

void ara(int fmatris[][4],int boy,int bul){
    int i,j;
    for (i = 0 ; i < boy ; i++) {
        for (j = 0 ; j < boy ; j++) {
            if (fmatris[i][j]==bul){
                printf("%d sayisi matriste %d .satir %d. sutundadir",bul,i,j);
                break;
            }
        }
    }
}

int main() {
    int matris[4][4] = {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15, 16}};
    int i, aranan;
    printf("aramak istediginiz sayiyi giriniz:");
    scanf("%d", &aranan);
    ara(matris,4,aranan);
    return 0;
}
```

| | 0 | 1 | 2 | 3 |
|---|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 |
| 1 | 5 | 6 | 7 | 8 |
| 2 | 9 | 10 | 11 | 12 |
| 3 | 13 | 14 | 15 | 16 |

A diagram illustrating a search for the value 5 in a 4x4 matrix. A red box containing the number 5 is connected by a red line to the cell at row 1, column 0 of the matrix, which also contains the number 5. The matrix is enclosed in a large red bracket on the right side.

Satırları Sıralama (b -> k)

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

Satırları Sıralama (b -> k)

```
#include <stdio.h>
void sirala(int fmatris[][4],int boy)
{
    int i,j,k,degis;
    for (i = 0 ; i < boy ; i++) {
        for (j = 0 ; j < boy ; j++) {
            for (k = 0 ; k < boy-1 ; k++) {
                if(fmatris[i][k]<fmatris[i][k+1]){
                    degis=fmatris[i][k];
                    fmatris[i][k]=fmatris[i][k+1];
                    fmatris[i][k+1]=degis;
                }
            }
        }
    }
}

int main() {
    int matris[4][4] = {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15, 16}};
    sirala(matris,4);
    int i, j;
    printf("matris:\n");
    for (i = 0 ; i < 4 ; i++) {
        for (j = 0 ; j < 4 ; j++) {
            printf("%2d ", matris[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    return 0;
}
```

| | | k | | | |
|---|---|----|----|----|----|
| | | j | | | |
| | | 0 | 1 | 2 | 3 |
| i | 0 | 2 | 1 | 3 | 4 |
| | 1 | 5 | 6 | 7 | 8 |
| | 2 | 9 | 10 | 11 | 12 |
| | 3 | 13 | 14 | 15 | 16 |