

Yıldız Teknik Üniversitesi
Elektrik Elektronik Fakültesi
Bilgisayar Mühendisliği Bölümü



Dönem Projesi
(25.12.2024 – 05.01.2025)

Ad Soyad: Nasrullah Toprak
Öğrenci Numarası:
Ders Adı & Kodu: Bilgisayar Bilimlerine Giriş (BLM1011)
Grup No: 2
Sınıf: 1
E-posta:

Problem Tanımı ve Kodlaması:

Aşağıdaki kodumda, üç kişinin sırayla oynayabileceği bir **Triversi** oyunu bulunmaktadır. **Birinci oyuncunun** taşları **kırmızı (K)**, **ikinci oyuncunun** taşları **sarı (S)** ve **üçüncü oyuncunun** taşları **mavi (M)** renktedir ve her oyuncunun taş sayısı sınırsızdır.

Oyun, $N \times N$ boyutlu bir tahtada oynanır. Tahtanın boyutu belirlendikten sonra, birinci oyuncu taşını tahtanın ortasına en yakın koordinata koyarak oyunu başlatır. Diğer oyuncular sırayla taşlarını, mevcut taşlara dokunacak şekilde yerleştirirler. Eğer yerleştirilen taşın yatay, dikey veya çapraz hizasında farklı bir renk taş varsa, bu taş ile koyulan taş arasındaki tüm taşlar, koyulan taşın rengine dönüşür.

Çalışmam (kodum) şu aşamalardan oluşur.

1. Başla
2. Kullanıcılardan $N \times N$ boyutlu tahtanın N satır sayısını iste ($3 \leq N \leq 23$).
 - 2.1. Eğer $N > 23$ ya da $N < 3$ tekrar kullanıcıdan iste.
3. Tahtayı yazdır.
4. Birinci oyuncu (K) tahtanın ortasındaki koordinatı iste.
 - 4.1. Tahtanın boyutuna bağlı ortalamasını hesapla
 - 4.2. Girilen koordinatlar oyunun ortasında değil ise tekrar iste.
 - 4.3. Tahtayı yazdır.
5. Sırasıyla diğer oyuncuların x ve y koordinatları iste.
 - 5.1. Girilen koordinatlar hiçbir taşla değmiyor ise (etrafı boş ise) tekrar kullanıcıdan iste.
 - 5.2. Tahtayı yazdır.
 - 5.3. Bir hamlenin renk değişimine sebep olup olmayacağını kontrol et.
 - 5.4. Hamle var (yatay/dikey/çapraz) ise renkleri koyulan taşın rengine dönüştür.
 - 5.5. Tahtayı yazdır.
6. Tahta dolduğu anda her oyuncunun kaç tane taşın olduğunu say ve kazananı yazdır.
7. Bitir

Kodlamaya ilişkin ayrıntılı RAPOR (detaylar ve örnekler):

```
//Kullanıcıdan Matrisin Satir/Sutun talebi (Eger N<=0 veya N>23 tekrar kullanıciidan talebi)
printf("1. Oyuncu -> Kirmizi (K)\n2. Oyuncu -> Sari (S)\n3. Oyuncu -> Mavi (M)\n\n");
printf("Triversi oyunu icin kare tahtanın satir/sutun sayisini giriniz: ");
scanf("%d", &N);

while (N<3 || N>23){
    printf("Kare tahtanın boyutu 3-23 arasinda olmalidir. Tekrar kare tahtanın satir sayisini giriniz: ");
    scanf("%d", &N);
}

//Bos kare tahtayi (N*N) yazdirma
printf("\nBaslangic Kare Tahtasi: \n");
for (i=0; i<N; i++){
    for (j=0; j<N; j++){
        A[i][j]='-';
        printf("%c ", A[i][j]);
    }
    printf("\n");
}
printf("\n");
(21-40. satirlar)
```

Kullanıcıdan 'N' değişkenin boyutunu istenmesi ($3 \leq N \leq 23$ olma şartıyla).

Boş tahtayı yazdırmak ($A[i][j]='-'$)

-: o koordinatın boş olduğunu göstermektedir

Boş matrisi yazdırmak

Birinci oyuncu tahtanın ortasına en yakın koordinatına koyması.

N çift ise: ortaya yakın dört tane koordinat bulunmaktadır.

N tek ise: matrisin ortasını belirten sadece bir tane koordinat bulunmaktadır.

$((0\%3)+1 = 0+1 = 1)$

```
i=0;
printf("%d. Oyuncu (K) rengi icin hamlenizi yapiniz: \nx ve y koordinatlari tahtanın ortasında olacak sekilde giriniz: ", (i%3)+1);
scanf("%d %d", &x, &y);
if (N%2!=0){
    orta1=(N/2)+1;
    orta2=orta1;
}
else{
    orta1=(N/2);
    orta2=(N/2)+1;
}

while (x<orta1 || x>orta2 || y<orta1 || y>orta2){
    printf("x ve/veya y degerleri hatali tekrar giriniz: ");
    scanf("%d %d", &x, &y);
}

A[x-1][y-1]='K';

printf("\nOlusan matris:\n");
for (p=0; p<N; p++){
    for (q=0; q<N; q++){
        printf("%c ", A[p][q]);
    }
    printf("\n");
}
printf("\n");
i++;
(43-70. satirlar)
```

Örnek (kullanıcı değerleri 1 ila N arasında girecek c dilindeki indislerde 0 ila N-1 arasında olacak):

N = 6

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

$orta1 = (6/2) = 3$
 $orta2 = (6/2)+1 = 4$
x ve y değerleri 3 veya 4 olmalı

N = 7

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

$orta1 = 7/2 + 1 = 4$
 $orta2 = orta1 = 4$
x ve y sadece 4 olmalı (tahtanın ortası)

c dilinde indisler 0 dan başlayıp N-1'e kadar olduğu için kullanıcı x ve y değerlerin karşılığı x-1 ve y-1 olacaktır ve 1. oyuncu rengi $A[x-1][y-1]$ indisine yazılacaktır.

K rengini o koordinata yazmak

```

while (i<sayac){
    renk degisim=0;
    //(i%3)+1: Hangi oyuncunun sirasi oldugunu belirtmek ve hangi renk (K,S,M) oldugunu belirtmek.
    //Belirtilen oyuncudan tasin koordinatlarini istemek.
    //x ve y koordinatlarini negatif, N'den buyuk, yada baska bir tasin oldugu yere olursa tekrardan x ve y'yi kullanicidan istemek.
    //x ve y [1-N] arasin da ama c dilinde indisler [0,N-1] arasin da. Kullanci x ve y giriyor ama matriste v ve w indislerine A[v][w] yaziliyor (v=x-1 & w=y-1).
    //Yatay/dik/çapraz kontrolleri için m ve n degiskenleri kullanılacak (v ve w degerleri sabit kalmali için).
    //Kullancinin girdigi rengi (K, S, M) matrise yazdirmak

    // Sirali olarak oyuncular dan x ve y koordinatlar talebi
    if (i%3==0) renk = 'K'; // 1. oyuncu renk kirmizi
    else if (i%3==1) renk = 'S'; // 2. oyuncu renk sarı
    else if (i%3==2) renk = 'M'; // 3. oyuncu renk mavı

    printf("%d. Oyuncu (%c) rengi için hamlenizi yapınız: \n x ve y koordinatları giriniz: ",(i%3)+1,renk);
    scanf("%d %d",&x,&y);

    v=x-1;
    w=y-1;
    while (x <= 0 || x > N || y <= 0 || y > N || A[v][w] != '-' || // Kullancinin degerleri [1-N] arasin da oldugunu kontrol et ve o konumda baska bir tasin olup olmadigi
    /*
    Tasin butun komsularini kontrol etmek (ortalarda olan tasin komsuluklari 8.. Kenarda olan tasların komsuluklari ortakilerden daha az ona gore kontrol)
    Eger tasin butun komsuluklari nda '-' var ise o zaman etrafı bastur ve kurallara gore en az 1 tane komsusu olması gerektiği x ve y tekrar kullanicidan istemek
    */
    ((v - 1 < 0 || w - 1 < 0 || A[v - 1][w - 1] == '-') &&
    (v - 1 < 0 || A[v - 1][w] == '-') &&
    (v - 1 < 0 || w + 1 >= N || A[v - 1][w + 1] == '-') &&
    (w - 1 < 0 || A[v][w - 1] == '-') &&
    (w + 1 >= N || A[v][w + 1] == '-') &&
    (v + 1 >= N || w - 1 < 0 || A[v + 1][w - 1] == '-') &&
    (v + 1 >= N || A[v + 1][w] == '-') &&
    (v + 1 >= N || w + 1 >= N || A[v + 1][w + 1] == '-')) {
    //x ve y hatalı olma durumunda tekrar istemek
    printf("x ve/veya y degerleri hatalı tekrar giriniz: ");
    scanf("%d %d",&x, &y);
    v=x-1;
    w=y-1;
    }
    A[v][w]=renk;
}

```

(72-11.....3. satırlar)

$((i\%3)+1)$ oyuncuların sırayla $(1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 2 \dots)$ ve o oyuncunun taşların rengi.

$((1 \% 3) + 1) = (1+1) = 2 \rightarrow$ renk: S

$((2 \% 3) + 1) = (2+2) = 3 \rightarrow$ renk: M

$((3 \% 3) + 1) = (0+1) = 1 \rightarrow$ renk: K

$((4 \% 3) + 1) = (1+1) = 2 \rightarrow$ renk: S ...

Her bir oyuncu için girilen koordinatlar 0 ila N arası olmalı ve o koordinatta başka bir taş olmaması gerek, aksi taktirde while döngüsüne girip tekrar kullanicidan istenmesi.

Ayrıca bu taşın diğer taşlara en az dikey/yatay/çapraz olma zorunda... Eğer taşın etrafındaki bütün koordinatlarda '-' var ise o zaman koordinatlar tekrar kullanicidan istenmektedir. En az bir tane '-' farklı bir taş koyulan taşın dikey/yatay/çaprazında ise o zaman eklenen taşın koordinatı geçerlidir.

Örnek:

-	-	-	-	-
-	-	-	-	-
-	-	K	S	-
-	-	-	-	M
-	-	K	-	-

Kırmızı K geçerli değil (tekrar başka bir koordinat istenecektir)

-	-	-	-	-
-	-	-	-	-
-	-	K	S	-
-	-	-	-	M
-	-	-	K	-

Kırmızı K'nın sağ üst çaprazında '-' yok (M var)... Geçerli...K o koordinata koyulacaktır

KONTROLLER:

Not: v ve w değerleri c dilindeki indisler gibi 0 ila N-1 arasında... m ve n kontrol için değişecekler

YATAY

```
// Yatay olarak bir hamle olup olmadigi kontrol:
//Yatay sol kontrol (satir sayisi (v) sabit ama sutun sayisi (n) bir bir azalacak)
m=v;
n=w-2;
if (A[m][n+1]!='-' && A[m][n+1]!=renk){
    while (n>=0 && ctrl==0 && A[m][n]!='-'){
        if (A[m][n]==renk){
            for (p=n;p<w;p++){
                A[m][p] = renk;
            }
            ctrl=1;
            renk_degisim=1;
        }
        n--;
    }
}
ctrl=0;

//Yatay sag kontrol (satir sayisi (v) sabit ama sutun sayisi (n) bir bir artacak)
m=v;
n=w+2;
if (A[m][n-1]!='-' && A[m][n-1]!=renk){
    while (n<N && ctrl == 0 && A[m][n]!='-'){
        if (A[m][n]==renk){
            for (p=n;p>w;p--){
                A[m][p] = renk;
            }
            ctrl=1;
            renk_degisim=1;
        }
        n++;
    }
}
ctrl=0;
```

(124-157. satırlar)

Örnek (Sol yatay): (kırmızı ile gösterilen taş en son koyulan taş olup ve ona bağlı hamle olup olmayacağı kontrol edilmektedir)

	0	1	2	3	4	5	6
0	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-
3	-	-	-	K	S	M	K
4	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-

v=3 ; w = 6; renk = 'K'

m=v=3; n=w-2=4

A[m][n+1]=A[3][5] ≠ '-' && A[3][5] ≠ renk

A[m][n]=A[3][4]≠renk... n-- → n=3 → ctrl=1(döngüden çık)

A[3][3] = renk (bulundu) → A[3][3]=A[3][4]=A[3][5]=renk

→ Renk değişimin olduğunu belirtmek için renk_degisim değişkeni 1 olarak atandı if bloğuna giriş yapması ve renk değişimin olduğunu göstermesi.

Sağ Yatay:

Sol yatay kontrol ile aynı sadece satır sayısı artacak (sağ doğru n++;) ve A[m][n-1] aynı renge eşit olmadığı ve '-' eşit olmadığı sürece devam edilecek.

	0	1	2	3	4	5	6
0	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-
2	-	-	-	-	-	M	-
3	-	-	-	K	S	- K	K
4	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-

v=3 ; w = 6; renk = 'K'

m=v=3; n=w-2=4

A[m][n+1]=A[3][5] = '-' || A[3][5] = renk
(KONTROL YAPILMIYOR)

Dik kontroller:

Yukarı kontrol: Sütün sayısı sabit (aynı sütun) ve satır sayısı azalıyor (yukarı kontrol)

Aşağı kontrol: Sütün sayısı sabit (aynı sütun) ve satır sayısı artıyor (aşağı kontrol)

```
//Dik kontroller
//Dik yukari kontrol (sutun (w) sayisi sabit ama satir sayisi degisiyor)
m=v-2;
n=w;
if (A[m+1][n]!='-' && A[m+1][n]!=renk){
    while (m>=0 && ctrl==0 && A[m][n]!='-'){
        if (A[m][n]==renk){
            for (p=m;p<v;p++){
                A[p][n]=renk;
            }
            ctrl=1;
            renk_degisin=1;
        }
        m--;
    }
}
ctrl=0;

//Dik asagi kontrol (sutun (w) sayisi sabit ama satir sayisi degisiyor )
m=v+2;
n=w;
if (A[m-1][n]!='-' && A[m-1][n]!=renk){
    while (m<N && ctrl==0 && A[m][n]!='-'){
        if (A[m][n]==renk){
            for (p=m;p>v;p--){
                A[p][n]=renk;
            }
            ctrl=1;
            renk_degisin=1;
        }
        m++;
    }
}
ctrl=0;
```

(159-192. satırlar)

Örnek (dik yukarı kontrol):

	0	1	2	3	4	5	6
0	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-
3	-	-	-	K	-	-	-
4	-	-	-	S	-	-	-
5	-	-	-	M	-	-	-
6	-	-	-	K	-	-	-

v=6 ; w=3; renk = 'K'

m=v-2=4; n=w=3

A[m+1][n]=A[5][3] ≠ '-' && A[3][5] ≠ renk

A[m][n]=A[4][3]≠renk... m-- → m=3 → ctrl=1(döngüden çık)

A[3][3] = renk (bulundu) → A[3][3]=A[4][3]=A[5][3]=renk

	0	1	2	3	4	5	6
0	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-
3	-	-	-	K	-	-	-
4	-	-	-	S	-	-	-
5	-	-	M	- K	-	-	-
6	-	-	-	K	-	-	-

v=6 ; w=3; renk = 'K'

m=v-2=4; n=w=3

A[m][n+1]=A[5][3] = '-' || A[5][3] = renk
(KONTROL YAPILMIYOR)

Dik Aşağı kontrol:

Yukarı yatay kontrol ile aynı sadece satır sayısı artacak (aşağı doğru m++;) ve A[m-1][n] aynı renge eşit olmadığı ve '-' eşit olmadığı sürece devam edilecek

ÇAPRAZ KONTROLLER:

```
//Sag Yukrai capraz kontrolu:
```

```

n=v-2;
n=w+2;
if (A[n+1][n-1]!='-' && A[n+1][n-1]!=renk){
    while (n>=0 && n<N && ctrl==0 && A[n][n]!='-'){
        if (A[n][n]==renk){
            p=n;
            q=n;
            while (p<v && q>w){
                A[p][q]=renk;
                p++;
                q--;
            }
            ctrl=1;
            renk_degisin=1;
        }
        n--;
        n++;
    }
}
ctrl=0;

```

```
//Sol Asagi capraz kontrolu:
```

```

n=v+2;
n=w-2;
if (A[n-1][n+1]!='-' && A[n-1][n+1]!=renk){
    while (n<N && n>=0 && ctrl==0 && A[n][n]!='-'){
        if (A[n][n]==renk){
            p=n;
            q=n;
            while (p>v && q<w){
                A[p][q]=renk;
                p--;
                q++;
            }
            ctrl=1;
            renk_degisin=1;
        }
        n++;
        n--;
    }
}
ctrl=0;

```

```
//Sol Yukari capraz kontrolu:
```

```

n=v-2;
n=w-2;
if (A[n+1][n+1]!='-' && A[m+1][n+1]!=renk){
    while (m>=0 && n>=0 && ctrl==0 && A[m][n]!='-'){
        if (A[m][n]==renk){
            p=n;
            q=m;
            while (p<v && q<w){
                A[p][q]=renk;
                p++;
                q++;
            }
            ctrl=1;
            renk_degisin=1;
        }
        m--;
        n--;
    }
}
ctrl=0;

```

```
//Sag Asagi capraz kontrolu:
```

```

n=v+2;
n=w+2;
if (A[n-1][n-1]!='-' && A[n-1][n-1]!=renk){
    while (n<N && n<N && ctrl==0 && A[n][n]!='-'){
        if (A[n][n]==renk){
            p=n;
            q=n;
            while (p>v && q>w){
                A[p][q]=renk;
                p--;
                q--;
            }
            ctrl=1;
            renk_degisi=n-1;
        }
        n++;
        n++;
    }
}
ctrl=0;

```

(194-280. satırlar)

Örnekler (kırmızı renk ile gösterilen renk en son eklenen renk ve ona bağlı yapılan kontrol):

Bütün örneklerde en son koyulan taşın aynı taş bulana kadar devam etmesi ve o taşı bulduğu anda döngüden çıkması.

Sağ Yukarı Kontrol

		n++ →					
		0	1	2	3	4	5
0	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-
2	-	-	-	K	-	-	-
3	-	-	S	-	-	-	-
4	-	-	M	-	-	-	-
5	K	-	-	-	-	-	-

m-- (m>=0)
n++ (n<N)
A[m][n]='-' Dur
A[m][n]≠renk... Devam et

Sol Yukarı Kontrol

Diagram illustrating a 6x6 matrix with row and column indices. The matrix is labeled with $m++$ on the left and $n++$ on the top. The matrix contains values: Row 0: [-, -, -, -, -, -]; Row 1: [-, -, -, -, -, -]; Row 2: [-, -, K, -, -, -]; Row 3: [-, -, -, S, -, -]; Row 4: [-, -, -, -, M, -]; Row 5: [-, -, -, -, -, K]. A diagonal arrow points from the top-right towards the bottom-left, passing through the cell (2,3) which contains 'K'.

Sol Aşağı Kontrol

	n++ →					
	0	1	2	3	4	5
0	-	-	-	-	-	K
1	-	-	-	-	S	-
2	-	-	-	M	-	-
3	-	-	K	-	-	-
4	-	-	-	-	-	-
5	-	-	-	-	-	-

m++ (m < N)
 n-- (n >= 0)
 A[m][n] = ' ' Dur
 A[m][n] ≠ renk ... Devam et

Sağ Aşağı Kontrol

		n++					
		0	1	2	3	4	5
0	K	-	-	-	-	-	-
1	-	M	-	-	-	-	-
2	-	-	S	-	-	-	-
3	-	-	-	K	-	-	-
4	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-

m++ (m<N)
n++ (n<N)
A[m][n]=''.... Dur
A[m][n]≠renk... Devam et

```
//Eger renk degisimi oldu ise yeni tahtayi yazdirma:
if (renk_degisim==1){
    printf("RENK DEGISIMI:\n");
    for (p=0;p<N;p++){
        for (q=0;q<N;q++){
            printf("%c ",A[p][q]);
        }
        printf("\n");
    }
    printf("\n");
}
```

```
i++;
/*while bitti (kullanici girdileri ve her biri icin kontrol) (282-297. satirlar)
```

Kontrollerden renk deęiřimi oldu ise (renk_degisim==1) renklerin deęiřtięi matrisi yazdır ve sıradaki oyuncuya ge.

```
//Tahtada kac tane renk oldugunu bulmak ve renk sayisini yazdirmek
for (i=0;i<N;i++){
    for (j=0;j<N;j++){
        if (A[i][j]=='K') kirmizisayac++;
        else if (A[i][j]=='S') sarisayac++;
        else if (A[i][j]=='M') mavisayac++;
    }
}

printf("Kirmizi Taslar Sayisi: %d\nSari Taslar Sayisi: %d\nMavi Taslar Sayisi: %d\n",kirmizisayac,sarisayac,mavisayac);

//En cok renge sahip olan oyuncu oyunu kazanir:
if (kirmizisayac > sarisayac && kirmizisayac>mavisayac)
    printf("1. OYUNCU KAZANDI");
else if (sarisayac>kirmizisayac && sarisayac >mavisayac)
    printf("2. OYUNCU KAZANDI");
else if (mavisayac> kirmizisayac && mavisayac>sarisayac)
    printf("3. OYUNCU KAZANDI");
else printf("BERABERLIK");

return 0;
}
```

(299-320. satirlar)

Renklerin sayımı ve renklere baęlı kullanıcılardan kazanını belirtmek.

Video Linki (kodun nasıl alıřtıęına dair):

https://drive.google.com/file/d/1eTwfbgTSat6cZWBj7GoYqzIRzEbrYd98/view?usp=drive_link

Örnek Oyundan Ekran Görüntüleri (7*7 matris):

```
C:\Users\user\Desktop\240119  X + v
1. Oyuncu -> Kirmizi (K)
2. Oyuncu -> Sari (S)
3. Oyuncu -> Mavi (M)

Triversi oyunu icin kare tahtanin satir/sutun sayisini giriniz: 7

Baslangic Kare Tahtasi:
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

1. Oyuncu (K) rengi icin hamlenizi yapiniz:
x ve y koordinatlari tahtanin ortasinda olacak sekilde giriniz: 4 4

Olusan matris:
- - - - -
- - - - -
- - - - -
- - K - -
- - - - -
- - - - -
- - - - -

2. Oyuncu (S) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 4 5

Olusan matris:
- - - - -
- - - - -
- - - - -
- - K S -
- - - - -
- - - - -
- - - - -

3. Oyuncu (M) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 4 3

Olusan matris:
- - - - -
- - - - -
- - - - -
- - M K S -
- - - - -
- - - - -
- - - - -
```


1. Oyuncu (K) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 4 2

Olusun matris:

```
- - - - -  
- - - - -  
- - - - -  
- K M K S - -  
- - - - -  
- - - - -  
- - - - -
```

RENK DEGISIMI:

```
- - - - -  
- - - - -  
- - - - -  
- K K K S - -  
- - - - -  
- - - - -  
- - - - -
```

2. Oyuncu (S) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 4 1

Olusun matris:

```
- - - - -  
- - - - -  
- - - - -  
S K K K S - -  
- - - - -  
- - - - -  
- - - - -
```

RENK DEGISIMI:

```
- - - - -  
- - - - -  
- - - - -  
S S S S S - -  
- - - - -  
- - - - -  
- - - - -
```

3. Oyuncu (M) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 3 3

Olusun matris:

```
- - - - -  
- - - - -  
- - M - - - -  
S S S S S - -  
- - - - -  
- - - - -  
- - - - -
```

1. Oyuncu (K) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 3 5

Olusun matris:

```
- - - - -  
- - - - -  
- - M - K - -  
S S S S S - -  
- - - - -  
- - - - -  
- - - - -
```

2. Oyuncu (S) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 2 6

Olusun matris:

```
- - - - -  
- - - - - S -  
- - M - K - -  
S S S S S - -  
- - - - -  
- - - - -  
- - - - -
```

RENK DEGISIMI:

```
- - - - -  
- - - - - S -  
- - M - S - -  
S S S S S - -  
- - - - -  
- - - - -  
- - - - -
```

3. Oyuncu (M) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 5 3

Olusun matris:

```
- - - - -  
- - - - - S -  
- - M - S - -  
S S S S S - -  
- - M - - - -  
- - - - -  
- - - - -
```

RENK DEGISIMI:

```
- - - - -  
- - - - - S -  
- - M - S - -  
S S M S S - -  
- - M - - - -  
- - - - -  
- - - - -
```

1. Oyuncu (K) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 2 2

Olusan matris:

```
- - - - -
- K - - - S -
- - M - S - -
S S M S S - -
- - M - - - -
- - - - -
- - - - -
```

2. Oyuncu (S) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 6 2

Olusan matris:

```
- - - - -
- K - - - S -
- - M - S - -
S S M S S - -
- - M - - - -
- S - - - - -
- - - - -
```

RENK DEGISIMI:

```
- - - - -
- K - - - S -
- - M - S - -
S S M S S - -
- - S - - - -
- S - - - - -
- - - - -
```

3. Oyuncu (M) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 5 5

Olusan matris:

```
- - - - -
- K - - - S -
- - M - S - -
S S M S S - -
- - S - M - -
- S - - - - -
- - - - -
```

RENK DEGISIMI:

```
- - - - -
- K - - - S -
- - M - S - -
S S M M S - -
- - S - M - -
- S - - - - -
- - - - -
```

1. Oyuncu (K) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 6 6

Olusan matris:

```
- - - - -
- K - - - S -
- - M - S - -
S S M M S - -
- - S - M - -
- S - - - K -
- - - - -
```

RENK DEGISIMI:

```
- - - - -
- K - - - S -
- - K - S - -
S S M K S - -
- - S - K - -
- S - - - K -
- - - - -
```

2. Oyuncu (S) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 2 3

Olusan matris:

```
- - - - -
- K S - - S -
- - K - S - -
S S M K S - -
- - S - K - -
- S - - - K -
- - - - -
```

RENK DEGISIMI:

```
- - - - -
- K S - - S -
- - S - S - -
S S S K S - -
- - S - K - -
- S - - - K -
- - - - -
```

3. Oyuncu (M) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 1 1

Olusan matris:

```
M - - - - -
- K S - - S -
- - S - S - -
S S S K S - -
- - S - K - -
- S - - - K -
- - - - -
```



C:\Users\user\Desktop\240115



1. Oyuncu (K) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 1 7

Olusan matris:

```
M - - - - K
- K S - - S -
- - S - S - -
S S S K S - -
- - S - K - -
- S - - - K -
- - - - - -
```

RENK DEGISIMI:

```
M - - - - K
- K S - - K -
- - S - K - -
S S S K S - -
- - S - K - -
- S - - - K -
- - - - - -
```

2. Oyuncu (S) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 2 1

Olusan matris:

```
M - - - - K
S K S - - K -
- - S - K - -
S S S K S - -
- - S - K - -
- S - - - K -
- - - - - -
```

RENK DEGISIMI:

```
M - - - - K
S S S - - K -
- - S - K - -
S S S K S - -
- - S - K - -
- S - - - K -
- - - - - -
```

3. Oyuncu (M) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 7 7

Olusan matris:

```
M - - - - K
S S S - - K -
- - S - K - -
S S S K S - -
- - S - K - -
- S - - - K -
- - - - - M
```

RENK DEGISIMI:

```
M - - - - K
S M S - - K -
- - M - K - -
S S S M S - -
- - S - M - -
- S - - - M -
- - - - - M
```



C:\Users\user\Desktop\240115



1. Oyuncu (K) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 7 1

Olusan matris:

```
M - - - - K
S M S - - K -
- - M - K - -
S S S M S - -
- - S - M - -
- S - - - M -
K - - - - M
```

RENK DEGISIMI:

```
M - - - - K
S M S - - K -
- - M - K - -
S S S K S - -
- - K - M - -
- K - - - M -
K - - - - M
```

2. Oyuncu (S) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 7 3

Olusan matris:

```
M - - - - K
S M S - - K -
- - M - K - -
S S S K S - -
- - K - M - -
- K - - - M -
K - S - - - M
```

3. Oyuncu (M) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 2 5

Olusan matris:

```
M - - - - K
S M S - M K -
- - M - K - -
S S S K S - -
- - K - M - -
- K - - - M -
K - S - - - M
```

RENK DEGISIMI:

```
M - - - - K
S M S - M K -
- - M - M - -
S S S K M - -
- - K - M - -
- K - - - M -
K - S - - - M
```



C:\Users\user\Desktop\240119



1. Oyuncu (K) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 2 4

Olusun matris:

```
M - - - - K
S M S K M K -
- - M - M - -
S S S K M - -
- - K - M - -
- K - - - M -
K - S - - - M
```

RENK DEGISIMI:

```
M - - - - K
S M S K K K -
- - M - M - -
S S S K M - -
- - K - M - -
- K - - - M -
K - S - - - M
```

2. Oyuncu (S) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 6 3

Olusun matris:

```
M - - - - K
S M S K K K -
- - M - M - -
S S S K M - -
- - K - M - -
- K S - - M -
K - S - - - M
```

RENK DEGISIMI:

```
M - - - - K
S M S K K K -
- - M - M - -
S S S K M - -
- - S - M - -
- K S - - M -
K - S - - - M
```

3. Oyuncu (M) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 5 1

Olusun matris:

```
M - - - - K
S M S K K K -
- - M - M - -
S S S K M - -
M - S - M - -
- K S - - M -
K - S - - - M
```

RENK DEGISIMI:

```
M - - - - K
S M S K K K -
- - M - M - -
S M S K M - -
M - S - M - -
- K S - - M -
K - S - - - M
```



C:\Users\user\Desktop\240119



1. Oyuncu (K) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 4 6

Olusun matris:

```
M - - - - K
S M S K K K -
- - M - M - -
S M S K M K -
M - S - M - -
- K S - - M -
K - S - - - M
```

RENK DEGISIMI:

```
M - - - - K
S M S K K K -
- - M - K - -
S M S K K K -
M - S - M - -
- K S - - M -
K - S - - - M
```

2. Oyuncu (S) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 1 6

Olusun matris:

```
M - - - - S K
S M S K K K -
- - M - K - -
S M S K K K -
M - S - M - -
- K S - - M -
K - S - - - M
```

3. Oyuncu (M) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 6 4

Olusun matris:

```
M - - - - S K
S M S K K K -
- - M - K - -
S M S K K K -
M - S - M - -
- K S M - M -
K - S - - - M
```

RENK DEGISIMI:

```
M - - - - S K
S M S K K K -
- - M - K - -
S M S K K K -
M - M - M - -
- K S M - M -
K - S - - - M
```



C:\Users\user\Desktop\240115



1. Oyuncu (K) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 2 7

Oluşan matris:

```
M - - - S K
S M S K K K K
- - M - K - -
S M S K K K -
M - M - M - -
- K S M - M -
K - S - - - M
```

2. Oyuncu (S) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 3 1

Oluşan matris:

```
M - - - S K
S M S K K K K
S - M - K - -
S M S K K K -
M - M - M - -
- K S M - M -
K - S - - - M
```

3. Oyuncu (M) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 1 5

Oluşan matris:

```
M - - - M S K
S M S K K K K
S - M - K - -
S M S K K K -
M - M - M - -
- K S M - M -
K - S - - - M
```

RENK DEĞİŞİMİ:

```
M - - - M S K
S M S M M K K
S - M - M - -
S M S K M K -
M - M - M - -
- K S M - M -
K - S - - - M
```

1. Oyuncu (K) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 1 2

Oluşan matris:

```
M K - - M S K
S M S M M K K
S - M - M - -
S M S K M K -
M - M - M - -
- K S M - M -
K - S - - - M
```



C:\Users\user\Desktop\240115



2. Oyuncu (S) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 1 3

Oluşan matris:

```
M K S - M S K
S M S M M K K
S - M - M - -
S M S K M K -
M - M - M - -
- K S M - M -
K - S - - - M
```

RENK DEĞİŞİMİ:

```
M K S - M S K
S S S M M K K
S - M - M - -
S M S K M K -
M - M - M - -
- K S M - M -
K - S - - - M
```

3. Oyuncu (M) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 6 7

Oluşan matris:

```
M K S - M S K
S S S M M K K
S - M - M - -
S M S K M K -
M - M - M - -
- K S M - M M
K - S - - - M
```

1. Oyuncu (K) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 6 5

Oluşan matris:

```
M K S - M S K
S S S M M K K
S - M - M - -
S M S K M K -
M - M - M - -
- K S M K M M
K - S - - - M
```

RENK DEĞİŞİMİ:

```
M K S - M S K
S S S M M K K
S - M - M - -
S M S K M K -
M - M - M - -
- K K K K M M
K - S - - - M
```



C:\Users\user\Desktop\240115



2. Oyuncu (S) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 3 7

Oluşan matris:

```
M K S - M S K
S S S M M K K
S - M - M - S
S M S K M K -
M - M - M - -
- K K K K M M
K - S - - - M
```

RENK DEĞİŞİMİ:

```
M K S - M S K
S S S M M K K
S - M - M - S
S M S K M S -
M - M - S - -
- K K S K M M
K - S - - - M
```

3. Oyuncu (M) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 6 1

Oluşan matris:

```
M K S - M S K
S S S M M K K
S - M - M - S
S M S K M S -
M - M - S - -
M K K S K M M
K - S - - - M
```

RENK DEĞİŞİMİ:

```
M K S - M S K
S S S M M K K
S - M - M - S
S M S K M S -
M - M - S - -
M M M M M M M
K - S - - - M
```

1. Oyuncu (K) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 1 4

Oluşan matris:

```
M K S K M S K
S S S M M K K
S - M - M - S
S M S K M S -
M - M - S - -
M M M M M M M
K - S - - - M
```

RENK DEĞİŞİMİ:

```
M K K K K K K
S S S M M K K
S - M - M - S
S M S K M S -
M - M - S - -
M M M M M M M
K - S - - - M
```



C:\Users\user\Desktop\240115



2. Oyuncu (S) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 5 7

Oluşan matris:

```
M K K K K K K
S S S M M K K
S - M - M - S
S M S K M S -
M - M - S - S
M M M M M M M
K - S - - - M
```

3. Oyuncu (M) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 4 7

Oluşan matris:

```
M K K K K K K
S S S M M K K
S - M - M - S
S M S K M S M
M - M - S - S
M M M M M M M
K - S - - - M
```

RENK DEĞİŞİMİ:

```
M K K K K K K
S S S M M K K
S - M - M - S
S M S K M M M
M - M - S - M
M M M M M M M
K - S - - - M
```

1. Oyuncu (K) rengi için hamlenizi yapınız:
x ve y koordinatları giriniz: 3 4

Oluşan matris:

```
M K K K K K K
S S S M M K K
S - M K M - S
S M S K M M M
M - M - S - M
M M M M M M M
K - S - - - M
```

RENK DEĞİŞİMİ:

```
M K K K K K K
S S K K K K K
S - M K M - S
S M S K M M M
M - M - S - M
M M M M M M M
K - S - - - M
```

```
C:\Users\user\Desktop\240119 X +
2. Oyuncu (S) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 3 6

Olusan matris:
M K K K K K K
S S K K K K K
S - M K M S S
S M S K M M M
M - M - S - M
M M M M M M M
K - S - - - M

3. Oyuncu (M) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 3 2

Olusan matris:
M K K K K K K
S S K K K K K
S M M K M S S
S M S K M M M
M - M - S - M
M M M M M M M
K - S - - - M

1. Oyuncu (K) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 7 6

Olusan matris:
M K K K K K K
S S K K K K K
S M M K M S S
S M S K M M M
M - M - S - M
M M M M M M M
K - S - - K M

2. Oyuncu (S) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 7 5

Olusan matris:
M K K K K K K
S S K K K K K
S M M K M S S
S M S K M M M
M - M - S - M
M M M M M M M
K - S - S K M

RENK DEGISIMI:
M K K K K K K
S S K K K K K
S M M K M S S
S S S K M M M
M - S - S - M
M M M S S M M
K - S - S K M
```

```
C:\Users\user\Desktop\240119 X +
3. Oyuncu (M) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 5 4

Olusan matris:
M K K K K K K
S S K K K K K
S M M K M S S
S S S K M M M
M - S M S - M
M M M S S M M
K - S - S K M

RENK DEGISIMI:
M K K K K K K
S S K K K K K
S M M K M S S
S S M K M M M
M - S M S - M
M M M S S M M
K - S - S K M

1. Oyuncu (K) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 5 6

Olusan matris:
M K K K K K K
S S K K K K K
S M M K M S S
S S M K M M M
M - S M S K M
M M M S S M M
K - S - S K M

RENK DEGISIMI:
M K K K K K K
S S K K K K K
S M M K M K S
S S M K K K M
M - S M S K M
M M M S S K M
K - S - S K M

2. Oyuncu (S) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 5 2

Olusan matris:
M K K K K K K
S S K K K K K
S M M K M K S
S S M K K K M
M S S M S K M
M M M S S K M
K - S - S K M
```



C:\Users\user\Desktop\240119



3. Oyuncu (M) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 7 2

Olusan matris:

```
M K K K K K K
S S K K K K K
S M M K M K S
S S M K K K M
M S S M S K M
M M M S S K M
K M S - S K M
```

1. Oyuncu (K) rengi icin hamlenizi yapiniz:
x ve y koordinatlari giriniz: 7 4

Olusan matris:

```
M K K K K K K
S S K K K K K
S M M K M K S
S S M K K K M
M S S M S K M
M M M S S K M
K M S K S K M
```

RENK DEGISIMI:

```
M K K K K K K
S S K K K K K
S M M K M K S
S S M K K K M
M S S K S K M
M M M K K K M
K K K K K K M
```

Kirmizi Taslar Sayisi: 27

Sari Taslar Sayisi: 9

Mavi Taslar Sayisi: 13

1. OYUNCU KAZANDI