

# BLM111 Programlama Dilleri I

## Hafta 11 Çok Boyutlu Diziler

Yrd. Doç. Dr. Caner ÖZCAN

# Çok Boyutlu Diziler

- ▶ Bir dizi birden fazla boyutlu olabilir.
- ▶ Örneğin 3x4 bir matris için 2 boyutlu bir dizi kullanırız.
- ▶ Üç boyutlu öklid uzayındaki x, y, z noktalarını saklamak için de 3 boyutlu bir diziyi tercih ederiz.
- ▶ i satırında ve j sütunundaki elemana ulaşmak için  $M[i][j]$  şeklinde yazarız.
  - Örneğin, 2-boyutlu dizi (matris) aşağıdaki gibi tanımlanır
  - `int M[5][9]; /* 5 satır ve 9 sütundan oluşuyor */`
  - Kavramsal olarak, M dizisi aşağıdakine benzer.

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

## 2 Boyutlu Dizilere Ulaşmak

```
/* değer atama */  
for (i=0; i<5; i++){  
    for (j=0; j<9; j++){  
        M[i][j] = 0;  
    }  
}
```

```
/* Toplama */  
toplama = 0;  
for (i=0; i<5; i++){  
    for (j=0; j<9; j++){  
        toplama += M[i][j];  
    }  
}
```

```
/* min ve max bulma */  
min = M[0][0];  
Max = M[0][0];  
for (i=0; i<5; i++){  
    for (j=0; j<9; j++){  
        if (M[i][j]<min) min=M[i][j];  
        if (M[i][j]>max) max=M[i][j];  
    }  
}  
printf("min: %d, max: %d\n", min, max);
```

# Çok Boyutlu Dizilere İlk Değer Atama

- ▶ Çok boyutlu dizilere ilk değer atamak için iç içe geçmiş bir boyutlu dizi yükleyicilerini kullanabiliriz.

```
int M[5][9] = { {1, 1, 1, 1, 0, 1, 1, 1, 1},  
                {0, 1, 0, 1, 0, 1, 0, 1, 0},  
                {1, 0, 0, 1, 1, 1, 0, 0, 1},  
                {0, 0, 0, 0, 1, 1, 1, 1, 1},  
                {1, 1, 1, 1, 0, 0, 0, 1, 1}};
```

- ▶ Eğer ilk değerler çok boyutlu dizinin elemanlarından az ise kalanlar 0 ile doldurulur.

```
int M[5][9] = { {1, 1, 2, 1, 2, 0, 0, 1, 1},  
                {0, 0, 0, 1, 1, 1, 1, 2, 1}};  
/* 2, 3 ve 4 üncü satırlar 0 ile doldurulacak*/
```

# Çok Boyutlu Dizilere İlk Değer Atama

- ▶ Eğer içerdeki liste bir satırın eleman sayısından az ise satırın kalan kısmı 0 ile doldurulur.

```
int M[5][9] = { {1, 1, 0, 0, 1, 1, 1, 1, 1},
                {0, 1, 1, 2, 1, 1},
                {1, 1, 2, 2, 3}};

/* M[1][6], M[1][7], M[1][8] 0 olacaklar*/
/* M[2][5], M[2][6], M[2][7], M[2][8] 0 sıfır olacaklar*/
/* 3 ve 4 üncü satırlar tümü 0 ile doldurulacaklar */
```

# Daha Yüksek Boyutlu Diziler

- Bir dizi herhangi bir boyutta tanımlanabilir

```
int Kup[8][8][8];      /* ebatları 8 olan bir küp */
int Prizma[4][6][10]; /* ebatları 4x6x10 olan bir
                       * dikdörtgenler prizması
                       */
/* ilk değerler atanabilir*/
float A[4][6][8] = {{{1, 2, 3}, {3, 4}}, {{3, 4}}};

Kup[2][3][4] = 2;
Prizma[3][5][8] = 6;
A[0][0][4] = 3.34;
```

# Çok Boyutlu Diziler

- ▶ 5 kişilik bir öğrenci grubu için 8 adet test uygulansın. Bunların sonuçlarını saklamak için 2 boyutlu bir dizi kullanalım.

```
#include<stdio.h>
int main( void ) {
    // 5 adet ogrenci icin 8 adet sinavi temsil etmesi icin bir ogrenci tablosu
    // olusturuyoruz. Bunun icin 5x8 bir matris yaratilmasi gerekiyor.
    int ogrenci_tablosu[ 5 ][ 8 ];
    int i, j;
    for( i = 0; i < 5; i++ ) {
        for( j = 0; j < 8; j++ ) {
            printf( "%d no.'lu ogrencinin ", ( i + 1 ) );
            printf( "%d no.'lu sinavi> ", ( j + 1 ) );
            // Tek boyutlu dizilerdeki gibi deger atiyoruz
            scanf( "%d", &ogrenci_tablosu[ i ][ j ] );
        }
    }
    return 0;
}
```

# Çok Boyutlu Diziler

- Bu programı çalıştırıp, öğrencilere çeşitli değerler atadığımızı düşünelim. Bunu görsel bir şekle sokarsak, aşağıdaki gibi bir çizelge oluşur:

		8 Sınav							
5 Öğrenci	1	80	76	58	90	27	60	85	95
	2	60	59	75	80	82	79	64	87
	3	77	...						
	4								
	5				...	67	60	84	

5.Öğrencinin 6.sınavı

- Tabloya bakarsak, 1.öğrenci sınavlardan, 80, 76, 58, 90, 27, 60, 85 ve 95 puan almış gözüküyor. Ya da 5.öğrencinin, 6.sınavından 67 aldığını anlıyoruz. Benzer şekilde diğer hücrelere gerekli değerler atanıp, ilgili öğrencinin sınav notları hafızada tutuluyor.



# Örnek: İki Matrisin Toplamı

```
#include <stdio.h>
#define SAT 2
#define SUT 3

int main(){
    int a[SAT][SUT] = {5, 3, 7, 0, 1, 2};

    int b[SAT][SUT] = {1, 2, 3, 4, 5, 6};
    int c[SAT][SUT];
    int i, j;

    puts("A Matrisi:");
    for(i=0; i<SAT; i++){
        for(j=0; j<SUT; j++)
            printf("%4d",a[i][j]);
        printf("\n");
    }
}
```

# Örnek: İki Matrisin Toplamı

```
puts("B Matrisi:");
for(i=0; i<SAT; i++){
    for(j=0; j<SUT; j++)
        printf("%4d",b[i][j]);
    printf("\n");
}
puts("\nC Matrisi:");
for(i=0; i<SAT; i++){
    for(j=0; j<SUT; j++){
        c[i][j] = a[i][j] + b[i][j];
        printf("%4d",c[i][j]);
    }
    printf("\n");
}
return 0;
}
```

# Örnek: Simetrik Matris

```
#include <stdio.h>
#include <conio.h>
int main(){
    int a[100][100];
    int simetri=1;
    int x, y, i, j;
    printf("Matrisin satir sayisini gir > ");
    scanf("%d",&x);
    printf("Matrisin sutun sayisini gir > ");
    scanf("%d",&y);
    printf("Matrisin degerlerini gir > ");
    for(i=0; i<x; i++) {
        for(j=0; j<y; j++) {
            printf("\n Deger [%d] [%d] --> ", i+1, j+1);
            scanf("%d", &a[i][j]);
        }
    }
}
```

# Örnek: Simetrik Matris

```
//NOT: a[i][j]==a[j][i] ise bu matris simetriktir.  
for(i=0; i<x; i++){  
    for(j=0; j<y; j++){  
        if(a[i][j]!=a[j][i])  
            simetri=0;  
        break;  
    }  
}  
}  
if(simetri==1)  
    printf("\nMatris Simetriktir.\n");  
else  
    printf("\nMatris Simetrik Degildir\n");  
  
return 0;  
}
```

# Örnek: Diziyi Matrise Çevirme

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int dizi[100];
    int a[100][100];
    int i, j, n, x, sat, sut;
    printf("Diziniz kac elemandan olusacak? > ");
    scanf("%d",&n);
    for(x=0; x<n; x++){
        printf("Dizinin [%d] . elemanini gir > ",x+1);
        scanf("%d",&dizi[x]);
    }
    printf("\nMatrisin Satir sayisini gir > ");
    scanf("%d",&sat);
    printf("matrisin sutun sayisini gir > ");
    scanf("%d",&sut);
```

# Örnek: Diziyi Matrise Çevirme

```
if(n%sat==0 && n%sut==0){
    x=0;
    for(i=0; i<sat; i++){
        for (j=0; j<sut; j++){
            a[i][j]=dizi[x];
            x++;
        }
    }
    printf("\n\nMATRIS > \n");
    for(i=0; i<sat; i++){
        for(j=0; j<sut; j++)
            printf("%3d", a[i][j]);
        printf("\n");
    }
} else
    printf("HATA! Dizi Eleman Sayisi Satir veya Sutuna Tam bolunmelidir");
return 0;
}
```

# Kaynaklar

- ▶ Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- ▶ J. G. Brookshear, “Computer Science: An Overview 10th Ed.”, Addison Wisley, 2009.
- ▶ Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- ▶ Paul J. Deitel, “C How to Program”, Harvey Deitel.
- ▶ Bayram AKGÜL, C Programlama Ders notları