BLM111 Programlama Dilleri I

Hafta 10
Diziler

Yrd. Doç. Dr. Caner ÖZCAN

Diziler

- Bilgisayarlar yardımıyla yapılan işlemlerde, çok sayıda veri girilmesi ve girilen verilerin işlenerek belirli bir sistematiğe göre sıralanması gerekebilir.
- Belirli bir düzende olan verilerin işlenmesi hem daha kolay hem de daha pratik tir.
- Bu nedenle bilgisayar programlarında çoklu verileri işlemek için "dizi" olarak adlandırılan sıralı veri alanları kullanılır.
- Tek isimle adlandırılan bu veri alanları genel olarak belleğe ardışık olarak yerleşirler.

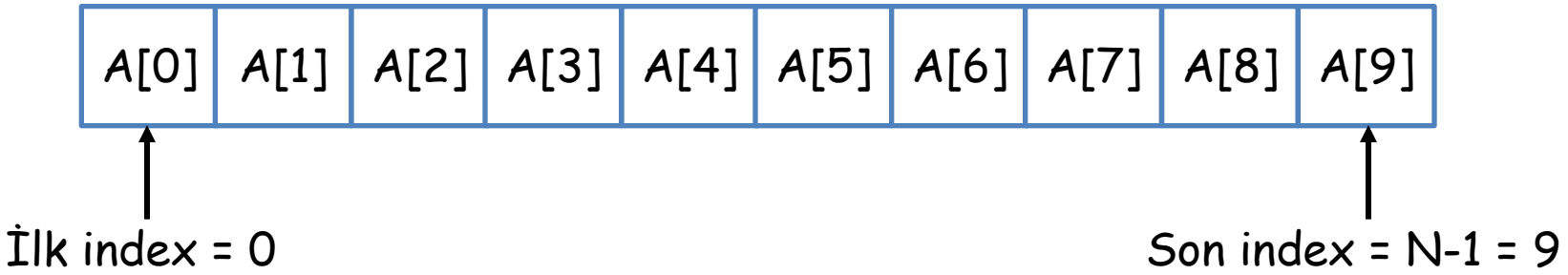
Diziler

- Aynı amaç için birden fazla aynı tip değişkene ihtiyacımız olur.
- Örneğin, 100 kişilik bir sınıfın "Programlama Dilleri" dersinden aldığı yılsonu notlarını tek tek değişkenlere aktarmak yerine (100 tane değişken adı gerekli), bunları ‘Notlar’ isimli bir diziye aktarılabilir.
- Bu şekilde birçok değişken adı ve alanı kullanılmaz.
- Bilgiler tek isim altında belirli bir düzen altında tutulur ve hızlı bir şekilde işlenirler.

Diziler

- Birden fazla aynı tip değişkeni bir arada tutan veri yapısıdır.
- En basit tipi bir boyutlu olanıdır.
  - Bir boyutlu dizinin elemanları bir satırda bir biri ardına dizilmiş şekilde kabul edilir.
- Kodlama:

```
#define N 10
…
int  A[N];
```



Diziler

- Dizinin n. elemanı c[n-1] ile gösterilir.
  - c[0]+ c[1]+ c[2]+…….c[n-1]
- Dizi elemanları normal değişkenler gibidir.
  - c[0] = 3;
  - printf(“%d”, c[0]);
- İndis numarası üzerinde işlemler gerçekleştirilebilir. a= 2, b=3 ise
  - c[a+b] += 8; // c[5] eleman değerine 8 ekler.
- Dizinin ilk üç elemanının değerleri toplamını yazdırmak için
  - printf(“%d”, c[0]+c[1]+c[2]);

Dizilere İlk Değer Atama

- Dizilere tanımlama sırasında ilk değer atanabilir.

```
int A[10]={8, 4, 10, 2, 5, 6, 7, 8, 9, 4};
```

- Eğer ilk değerler dizinin eleman sayısından az ise kalan elemanların değeri 0 olur.

```
int A[10]={1, 2, 3, 4};
/* A[10] dizisinin ilk değerleri {1, 2, 3, 4, 0, 0, 0, 0, 0, 0}*/
```

- Eğer ilk değerlerle bir dizi tanımlıyorsak, dizinin boyutunu boş bırakabiliriz.

```
int A[]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
/* A dizisinin 10 elemanı var A[0]..A[9] */
```

Dizilere İlk Değer Atama

- Eğer belirtilen eleman sayısı daha az ise kalan elemanların hepsinin değeri sıfır yapılır.

```
int n[5] = {0}; // tüm elemanların değeri 0 olur
```

- Dizi elemanlarının başlangıç değerleri otomatik olarak sıfır olmaz. Bunun için en azından ilk eleman değeri sıfır yapılmalıdır.

- Eğer gereğinden fazla başlangıç değeri varsa hata oluşur.

```
int n[5] = {1, 2, 3, 4, 5, 6}; //altı adet başlangıç değeri
```

Dizi Kullanımı

- Eğer belirtilen eleman sayısı daha az ise kalan elemanların hepsinin değeri sıfır yapılır.
- Dizilerin her bir elemanına ulaşmak için her elemanın indeksini kullanmamız gerekiyor.
- Bir indeks elemanın dizideki yerini ifade ediyor.
- Dizinin elemanları peş peşe sıralanmıştır. (arada boşluk yok)
- Dizinin her elamanı sırasıyla tanımlanır ve bu sıralama 0 dan başlar.

Dizi Kullanımı

- Örnek

```
#define MAX_OGR_SAYISI 5
…
int notlar[MAX_OGR_SAYISI];


…
notlar[0] = 98;
notlar[1] = 87;
notlar[2] = 92;
notlar[3] = 79;
notlar[4] = 85;
```

Dizi Kullanımı

- Uyarı!
  - C indexlerin doğru aralıkta olup olmadığını kontrol etmiyor. (yani, index değerleri, [ ] operatorü kullanılırken dizinin sınırları içerisinde mi değil mi diye kontrol edilmiyor.)

```
#define MAX_OGR_SAYISI 5
…
int notlar[MAX_OGR_SAYISI];

…

notlar[53] = 98;
notlar[5] = 98;
```

Dizi Kullanımı

- Dizinin elemanlarına ulaşılırken genelde döngüler kullanılır, ve döngünün her iterasyonunda dizinin bir elemanı üzerinde çalışılır.

- En sık kullanılan döngü for döngüsüdür. Çünkü döngü ifadesinde açıkça hem ilk değer atamaları hem de indeks değişkeni kullanılabiliyor:

```
notlar[0] = 0;
notlar[1] = 0;
notlar[2] = 0;
notlar[3] = 0;
notlar[4] = 0;
```

```
int i;
for(i = 0; i < MAX_OGR_SAYISI; i++)
        notlar[i] = 0;
```

Örnek: Okuma

```
#include <stdio.h>
#define SIZE 5

int main(void)
{
   int i;
   double a[SIZE];
   printf("%d tane dizi elemanı gir: ", SIZE);
   /* dizi elemanlarını oku*/
   for(i = 0; i < SIZE; i++)
        scanf("%lf", &a[i]);
   return 0;
}
```

```
5 tane dizi elemanı gir: 1.2 3.4 5.6 7.8 9.0
```

Örnek: Yazma

```
#include <stdio.h>
#define SIZE 5

int main(void)
{
   int i;
   double a[SIZE] = { 1.2, 3.4, 5.6, 7.8,
                      9.0 };
   /* Dizi elemanlarını göster*/
   for(i = 0; i < SIZE; i++)
       printf("a[%d] = %.2lf\n", i, a[i]);

   return 0;
}
```

```
a[0] = 1.20
a[1] = 3.40
a[2] = 5.60
a[3] = 7.80
a[4] = 9.00
```

Örnek: En Büyük Eleman

```
#include<stdio.h>

#define SIZE 5

int main(void)
{
    int i;
    double a[SIZE] = { 1.2, 3.4,
                       5.6, 7.8,
                       9.0 };
    double max = 0.0;
    /* Dizideki en büyük elemanı bul*/
    for(i = 0; i < SIZE; i++)
         if (a[i] > max)
                max = a[i];
    printf("max = %.2lf\n", max);
    return 0;
}
```

```
max = 9.00
```

Örnek: Dizi Toplamı

```
#include <stdio.h>

int main(void) {
   int i, N, A[100], B[100], C[100];
   printf("Dizinin eleman sayisini gir:\n");
   scanf("%d", &N);
   for(i = 0; i < N; i++){  /* Dizi elemanlarını oku*/
        printf("A[%d]=",i);
        scanf("%d",&A[i]);
   }
   for(i = 0; i < N; i++){  /* Dizi elemanlarını oku*/
        printf("B[%d]=",i);
        scanf("%d",&B[i]);
   }
   for(i = 0; i < N; i++){  /* Dizi elemanlarını oku*/
        C[i] = A[i] + B[i];
        printf("C[%d]=%d\n", i, C[i]);
   }
   return 0;
}
```

Örnek: Dizi Ortalama ve Standart Sapma

```
#include <stdio.h>
#include <math.h>
#define N 10
int main(){
   int   i;
   float x[N], toplam = 0.0, ort, std_sap = 0.0;
   /* ortalama hesabı */
   for(i=0; i<N; i++){
      printf("%d. sayi : ",i+1);
      scanf("%f",&x[i]);
      toplam += x[i];
   }
   ort = toplam/N;
   /* standart sapma hesabı */
   for(toplam = 0.0, i=0; i<N; i++)
       toplam += pow(x[i]-ort, 2.0);
   std_sap = sqrt( toplam/(N-1) );
   printf("Ortalama        = %f\n",ort);
   printf("Standart sapma = %f\n",std_sap);
   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