

DİZİ DEĞİŞKENLER

Aynı türden değişkenleri daha kolay yazabilmek ve programlamak için indisli değişkenler kullanılır. Bilindiği gibi aralarında ortak bir ilişki bulunan elemanlar topluluğuna küme denmektedir. Örneğin bir sınıftaki öğrenciler, bir ülkenin kentleri, bir polinomun katsayıları veya bir matrisin elemanları birer küme teşkil etmektedir. Elemanlar küme içinde belirli bir sıra içinde iseler, başka bir deyişle küme içindeki yerleri belli ise, küme dizi adını almaktadır.

Bir dizide herhangi bir elemanın yeri elemana bir sıra numarası vererek belirlenir. Elemanın dizi içindeki yerini belirleyen bu bilgiye indis denir. Diziler bir boyutlu veya çok boyutlu olabilirler. Örneğin bir sınıftaki öğrencilerin numaraları bir boyutlu bir dizi olup NO1, NO2, NO3,... şeklinde gösterilebilir. Bunun yerine hepsi daha kısa NO(I) şeklinde de gösterilebilir. Burada I indis olup 1,2,3,... değerlerini olmaktadır. Benzer şekilde alfasayısal değişkenler de indisli değişken olabilirler. AD1, AD2 , AD3,...yerine AD(I) yazılabilir.

Dizi kullanımı bazı işlemlerin anlatımına kolaylık ve sadelik gösterir. Örnek olarak bir polinom,

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_n x^n$$

şeklinde yazılabilir. Burada a_i , ($i=0,1,2,\dots,n$) polinomun katsayılarıdır. Bilindiği gibi bu polinom dizi elemanlarının bir toplamı olarak

$$p(x) = \sum_{i=0}^n a_i x^i$$

şeklinde gösterilebilir. Görüldüğü gibi anlatım böylece kısalmış (kolaylaşmış) ve sadeleşmiştir. Benzer şekilde n bilinmeyenli bir denklem sistemi matris yazma tarzıyla şu şekilde gösterilebilir:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & \dots & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Daha kısa olarak $A.X = B$ şeklinde yazmak da mümkündür. Burada A katsayı matrisi, X bilinmeyenler vektörü, B ise sabitler vektörüdür.

Dizi kullanımı yardımıyla dizi şeklinde verilebilen değişkenlerle ilgili işlemler kolaylaşmıştır. Çünkü dizilerin getirdiği anlatım kolaylıkları algoritmaların ve programların çok daha kolay yapılmasını sağlamaktadır. Hatta bazı problemlerin çözümü, örneğin yukarıda verilen denklem sisteminin çözümünü dizi kullanmadan programlamak hemen hemen imkansızdır.

Dizilerin Bildirimi

Bir dizi çok sayıda değişken barındırdığından, bunları birbirinden ayırtmak için indis adı verilen bir bilgiye ihtiyaç vardır. C programlama dilinde, bir dizi hangi tipte tanımlanmış olursa olsun başlangıç indisi her zaman 0'dır.

Bir dizinin bildirim işleminin genel biçimi şöyledir:

➤ veriTipi dizi_adı[eleman_sayısı];

Örneğin, 5 elemanlı, kütle verilerini bellekte tutmak için, kütle dizisi şöyle tanımlanabilir:

➤ float kütle[5];

Bu dizinin elemanlarına bir değer atama işlemi şöyle yapılabilir:

➤ kütle[0] = 8.47
➤ kütle[1] = 3.68
➤ kütle[2] = 9.10
➤ kütle[3] = 4.73
➤ kütle[4] = 3.91

Bir dizinin bellekte kapladığı alanın bayt cinsinden karşılığı sizeof operatörü ile öğrenilebilir.

➤ int a[5], b, c;
➤ ...
➤ b = sizeof(a); /* bellekte kapladığı alan: b = 4*5 = 20 bayt */
➤ c = sizeof(a) / sizeof(int); /* Dizinin boyutu : c = 20/4 = 5 */

Dizilere Başlangıç Değeri Atama

Bir diziye başlangıç değerleri küme parantezi içinde kısa formda atanabilir. Örneğin yukarıdaki kütle dizisinin değerleri aşağıdaki gibi verilebilir:

➤ float kütle[5] = { 8.47, 3.68, 9.10, 4.73, 3.91 };
➤ int maliyet[3] = { 25, 72, 94 };
➤ double a[4] = { 10.0, 5.2, 7.5, 0.0};

Küme parantezlerinin sonlandırıcı ; karakteri ile bittiğine dikkat edilmelidir. Bir dizinin uzunluğu (eleman sayısı) belirtilmeden de başlangıç değerleri atamak mümkündür.

➤ int a[] = { 100, 200, 300, 400 };
➤ float v[] = { 9.8, 11.0, 7.5, 0.0, 12.5};

Bu durumda derleyici bu şekilde bir atama ile karşılaştığında, küme parantezi içindeki eleman sayısını hesaplar ve dizinin o uzunlukta açıldığını varsayar. Yukarıdaki örnekte, a dizisinin 4, v dizisinin 5 elemanlı olduğu varsayılır.

Dizileri Yazdırma/Okuma

Printf() ve scanf() fonksiyonları bir dizinin okunması ve yazdırılması için de kullanılır. Örneğin bir A dizisinin aşağıdaki gibi bildirildiğini varsayalım;

- `int A[10];`

Bu dizinin elemanlarını klavyeden okumak için;

- `for(i=0; i<10; i++)`
- `scanf("%d",&A[i]);`

komutları kullanılabilir. Daha sonra bu değerlerin ekrana yazılması istenirse,

- `for(i=0; i<10; i++)`
- `printf("%d\n",A[i]);`

şeklinde yazılabilir.

Karakter Dizileri (Strings)

C dilinde, karakter dizileri oldukça sık kullanılır. Sadece karakter dizilerine özel olarak, karakter dizilerinin sonuna sonlandırıcı karakter olarak adlandırılan bir simge eklenir. Sonlandırıcı karakter, işlemlerin hızlı ve etkin bir biçimde yapılabilmesine olanak sağlar.

Sonlandırıcı karakter:

- dizinin bittiği yeri gösterir,
- ASCII tablosunun sıfır numaralı (' \0') karakteridir.

Karakter dizilerine iki şekilde başlangıç değeri verilebilir. Örneğin

- `char s[7] = {'d','e','n','e','m','e','\0'};`

ya da

- `char s[7] = "deneme";`

Birinci tanımlamada sonlandırıcı karakter programcı tarafından konmalıdır. İkinci tanımlamada ise buna gerek yoktur. Çünkü, sonlandırıcı karakter bu atamayla, derleyici tarafından eklenir.

Karakter dizileri gets() fonksiyonu ile klavyeden okunabilir.

- `char ad[20];`
- `...`
- `gets(ad);`

şeklinde yazılabilir.