

Programlamaya Giriş

Diziler

Hüseyin Ahmetođlu

Dizi Bildirimi ve Kullanımı

- ▶ 0 öğrencilik bir sınıf için notların ortalamasını bulmak istediğinizi varsayalım, 30 değişken oluşturmak istemezsiniz: mark1, mark2, ..., mark30. Bunun yerine, dizi adı verilen, 30 öğeli tek bir değişken kullanabilirsiniz.
- ▶ Bir dizi, bir çift köşeli parantez [] ile tanımlanan aynı türdeki öğelerin listesidir. Bir dizi kullanmak için, diziyi 3 şeyle bildirmeniz gerekir: bir ad, bir tür ve bir boyut (veya boyut veya uzunluk).

```
type arrayName[arrayLength];
```

Dizi Bildirimi ve Kullanımı

```
int marks[5];           // Declare an int array called marks with 5 elements
double numbers[10];    // Declare an double array of 10 elements

// Use #define to specify the length
#define SIZE 9
int numbers[SIZE];

// Some compilers support an variable as array length, e.g.,
const int SIZE = 9;
float temps[SIZE];     // Use const int as array length

int size;
printf("Enter the length of the array: ");
scanf("%d", &size);
float values[size];
```

Dizi Bildirimi ve Kullanımı

```
int marks[5];           // Declare an int array called marks with 5 elements
double numbers[10];    // Declare an double array of 10 elements

// Use #define to specify the length
#define SIZE 9
int numbers[SIZE];

// Some compilers support an variable as array length, e.g.,
const int SIZE = 9;
float temps[SIZE];     // Use const int as array length

int size;
printf("Enter the length of the array: ");
scanf("%d", &size);
float values[size];
```

Dizi Bildirimi ve Kullanımı

```
// Declare and initialize an int array of 3 elements
int numbers[3] = {11, 33, 44};
// If length is omitted, the compiler counts the elements
int numbers[] = {11, 33, 44};
// Number of elements in the initialization shall be equal to or less than length
int numbers[5] = {11, 33, 44}; // Remaining elements are zero. Confusing! Don't do this
int numbers[2] = {11, 33, 44}; // ERROR: too many initializers

// Use {0} or {} to initialize all elements to 0
int numbers[5] = {0}; // First element to 0, the rest also to zero
int numbers[5] = {}; // All element to 0 too
```

```

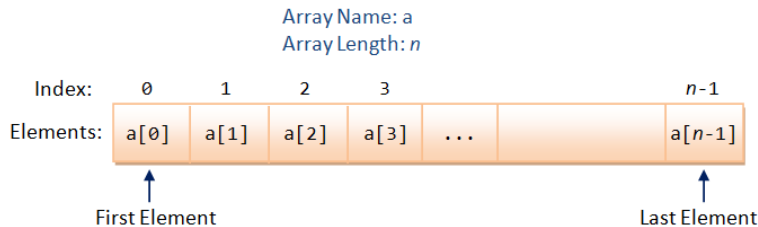
1  /* Test local array initialization (TestArrayInit.c) */
2  #include <stdio.h>
3  #define SIZE 5
4
5  int main() {
6      int i;
7
8      int a1[SIZE]; // Uninitialized
9      for (i = 0; i < SIZE; ++i) printf("%d ", a1[i]);
10     printf("\n"); // ? ? ? ? ?
11
12     int a2[SIZE] = {21, 22, 23, 24, 25}; // All elements initialized
13     for (i = 0; i < SIZE; ++i) printf("%d ", a2[i]);
14     printf("\n"); // 21 22 23 24 25
15
16     int a3[] = {31, 32, 33, 34, 35}; // Size deduced from init values
17     int a3Size = sizeof(a3)/sizeof(int);
18     printf("Size is %d\n", a3Size); // 5
19     for (i = 0; i < a3Size; ++i) printf("%d ", a3[i]);
20     printf("\n"); // 31 32 33 34 35
21
22     int a4[5] = {41, 42}; // Leading elements initialized, the rests to 0
23     for (i = 0; i < SIZE; ++i) printf("%d ", a4[i]);
24     printf("\n"); // 41 42 0 0 0
25
26     int a5[5] = {0}; // First elements to 0, the rests to 0 too
27     for (i = 0; i < SIZE; ++i) printf("%d ", a5[i]);
28     printf("\n"); // 0 0 0 0 0
29
30     int a6[5] = {}; // All elements to 0 too
31     for (i = 0; i < SIZE; ++i) printf("%d ", a6[i]);
32     printf("\n"); // 0 0 0 0 0
33
34     // Using variable as the length of array
35     const int SIZE_2 = 5;
36     int a7[SIZE_2]; // okay without initialization
37     int a8[SIZE_2] = {5, 4, 3, 2, 1}; // error: variable-sized object may
38 }

```

Dizi elemanlarına erişim

- Köşeli parantez [] içine alınmış bir index (veya alt simge) aracılığıyla bir dizinin bir ögesine başvurabilirsiniz. C'nin dizi indeksi sıfırla başlar. Örneğin, eleman sayısı 5 olan bir int dizisi olduğunu ve bu durumda 5 ögenin: dizi [0], dizi [1], işaretler [2], dizi [3] ve dizi [4] olduğunu varsayalım.

```
// Declare & allocate a 5-element int array
int marks[5];
// Assign values to the elements
marks[0] = 95;
marks[1] = 85;
marks[2] = 77;
marks[3] = 69;
marks[4] = 66;
printf("%d\n", marks[0]);
printf("%d\n", marks[3] + marks[4]);
```



Dizi öğelerinin Değerini Değiştirme

```
int mark[5] = {19, 10, 8, 17, 9}  
  
// make the value of the third element to -1  
mark[2] = -1;  
  
// make the value of the fifth element to 0  
mark[4] = 0;
```


Array Input/Output

```
// Program to take 5 values from the user and store them in an array
// Print the elements stored in the array
#include <stdio.h>

int main() {
    int values[5];

    printf("Enter 5 integers: ");

    // taking input and storing it in an array
    for(int i = 0; i < 5; ++i) {
        scanf("%d", &values[i]);
    }

    printf("Displaying integers: ");

    // printing elements of an array
    for(int i = 0; i < 5; ++i) {
        printf("%d\n", values[i]);
    }
    return 0;
}
```

Output

```
Enter 5 integers: 1
-3
34
0
3
Displaying integers: 1
-3
34
0
3
```

Ortalamayı Hesapla

```
// Program to find the average of n numbers using arrays

#include <stdio.h>
int main()
{
    int marks[10], i, n, sum = 0, average;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    for(i=0; i<n; ++i)
    {
        printf("Enter number%d: ",i+1);
        scanf("%d", &marks[i]);

        // adding integers entered by the user to the sum variable
        sum += marks[i];
    }

    average = sum/n;
    printf("Average = %d", average);

    return 0;
}
```

Output

```
Enter n: 5
Enter number1: 45
Enter number2: 35
Enter number3: 38
Enter number4: 31
Enter number5: 49
Average = 39
```

Öğelere sınırlarının dışında erişim

- ▶ 10 öğeden oluşan bir dizi tanımladığınızı varsayalım
- ▶ Dizi öğelerine `testArray[0]` 'dan `testArray [9]`' a erişebilirsiniz.

Şimdi `testArray [12]` 'ye erişmeye çalışırsanız, olmayan öğeye erişmeye çalışırsınız. Bu, beklenmeyen çıktılara (tanımsız davranış) neden olabilir. Bazen bir hata alabilirsiniz veya başka bir zamanda programınız doğru şekilde çalışabilir.

C dizi index'inin sınırlarını kontrol etmez. Bu nedenle, bir dizinin sınırlarının dışındaki öğelerine asla erişmemelisiniz.

```
int testArray[10];
```

Dizi boyutu

- ▶ Bir dizi oluşturmak için dizinin uzunluğunu (veya boyutunu) önceden bilmeniz ve buna göre ayırmanız gerekir. Bir dizi oluşturulduktan sonra uzunluğu sabittir ve değiştirilemez. Bazen bir dizinin uzunluğunu belirlemek zordur (örneğin, bir sınıfta kaç öğrenci var?). Bununla birlikte, uzunluğu tahmin etmeniz ve bir üst sınır belirlemeniz gerekir. Bu muhtemelen bir dizi kullanmanın en büyük dezavantajıdır.

Dizi uzunluğunu `sizeof (diziAdı) / sizeof (diziAdı [0])` ifadesini kullanarak bulabilirsiniz, burada `sizeof (diziAdı)` dizinin toplam baytını döndürür ve `sizeof (diziAdı [0])` ilk öğenin baytlarını döndürür.

Diziler ve Döngüler

- Diziler, döngülerle el ele çalışır. Bir dizinin tüm elemanları bir döngü aracılığıyla işlenebilir.

```
1  /*
2  * Find the mean and standard deviation of numbers kept in an array (MeanStdArray.c).
3  */
4  #include <stdio.h>
5  #include <math.h>
6  #define SIZE 7
7
8  int main() {
9      int marks[] = {74, 43, 58, 60, 90, 64, 70};
10     int sum = 0;
11     int sumSq = 0;
12     double mean, stdDev;
13     int i;
14     for (i = 0; i < SIZE; ++i) {
15         sum += marks[i];
16         sumSq += marks[i] * marks[i];
17     }
18     mean = (double)sum/SIZE;
19     printf("Mean is %.21f\n", mean);
20
21     stdDev = sqrt(((double)sumSq/SIZE - mean*mean));
22     printf("Std dev is %.21f\n", stdDev);
23
24     return 0;
25 }
```

Çok boyutlu diziler

```
int[2][3] = { {11, 22, 33}, {44, 55, 66} };
```

	Column 0	Column 1	Column 2	Column 3	
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]	...
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]	...

Row Index Column Index

Çok boyutlu diziler

```
// Different ways to initialize two-dimensional array  
  
int c[2][3] = {{1, 3, 0}, {-1, 5, 9}};  
  
int c[][3] = {{1, 3, 0}, {-1, 5, 9}};  
  
int c[2][3] = {1, 3, 0, -1, 5, 9};
```

```
int test[2][3][4] = {  
    {{3, 4, 2, 3}, {0, -3, 9, 11}, {23, 12, 23, 2}},  
    {{13, 4, 56, 3}, {5, 9, 3, 5}, {3, 1, 4, 9}}};
```

Çok boyutlu diziler

```
// C program to store temperature of two cities of a week and display it.
#include <stdio.h>
const int CITY = 2;
const int WEEK = 7;
int main()
{
    int temperature[CITY][WEEK];

    // Using nested loop to store values in a 2d array
    for (int i = 0; i < CITY; ++i)
    {
        for (int j = 0; j < WEEK; ++j)
        {
            printf("City %d, Day %d: ", i + 1, j + 1);
            scanf("%d", &temperature[i][j]);
        }
    }
    printf("\nDisplaying values: \n\n");

    // Using nested loop to display vlues of a 2d array
    for (int i = 0; i < CITY; ++i)
    {
        for (int j = 0; j < WEEK; ++j)
        {
            printf("City %d, Day %d = %d\n", i + 1, j + 1, temperature[i][j]);
        }
    }
    return 0;
}
```

```
City 1, Day 1: 33
City 1, Day 2: 34
City 1, Day 3: 35
City 1, Day 4: 33
City 1, Day 5: 32
City 1, Day 6: 31
City 1, Day 7: 30
City 2, Day 1: 23
City 2, Day 2: 22
City 2, Day 3: 21
City 2, Day 4: 24
City 2, Day 5: 22
City 2, Day 6: 25
City 2, Day 7: 26
```

Displaying values:

```
City 1, Day 1 = 33
City 1, Day 2 = 34
City 1, Day 3 = 35
City 1, Day 4 = 33
City 1, Day 5 = 32
City 1, Day 6 = 31
City 1, Day 7 = 30
City 2, Day 1 = 23
City 2, Day 2 = 22
City 2, Day 3 = 21
City 2, Day 4 = 24
City 2, Day 5 = 22
City 2, Day 6 = 25
City 2, Day 7 = 26
```



```
// C program to find the sum of two matrices of order 2*2
```

```
#include <stdio.h>
int main()
{
    float a[2][2], b[2][2], result[2][2];

    // Taking input using nested for loop
    printf("Enter elements of 1st matrix\n");
    for (int i = 0; i < 2; ++i)
        for (int j = 0; j < 2; ++j)
        {
            printf("Enter a%d%d: ", i + 1, j + 1);
            scanf("%f", &a[i][j]);
        }

    // Taking input using nested for loop
    printf("Enter elements of 2nd matrix\n");
    for (int i = 0; i < 2; ++i)
        for (int j = 0; j < 2; ++j)
        {
            printf("Enter b%d%d: ", i + 1, j + 1);
            scanf("%f", &b[i][j]);
        }
}
```

```
// adding corresponding elements of two arrays
for (int i = 0; i < 2; ++i)
    for (int j = 0; j < 2; ++j)
    {
        result[i][j] = a[i][j] + b[i][j];
    }

// Displaying the sum
printf("\nSum Of Matrix:");

for (int i = 0; i < 2; ++i)
    for (int j = 0; j < 2; ++j)
    {
        printf("%.1f\t", result[i][j]);

        if (j == 1)
            printf("\n");
    }
return 0;
}
```

```
Enter elements of 1st matrix
Enter a11: 2;
Enter a12: 0.5;
Enter a21: -1.1;
Enter a22: 2;
Enter elements of 2nd matrix
Enter b11: 0.2;
Enter b12: 0;
Enter b21: 0.23;
Enter b22: 23;
```

```
Sum Of Matrix:
2.2    0.5
-0.9   25.0
```

Soru

- ▶ Kullanıcıdan aldığı öğrenci sayısına göre, yine kullanıcı tarafından girilen bir dersin notlarını bir dizide saklayan ve bu diziyi kullanarak;
 - 10'ar puanlık aralıklar için yatay histogramı grafiğini veren
 - En yüksek, en düşük ve ortalama notu veren
 - 60 başarı notuna göre başarılı başarısız öğrenci sayılarını ve yüzdelerini veren
- ▶ C console programını kodlayınız.

KAYNAKLAR

- ▶ Goel, A., & Mittal, A. (2016). Computer Fundamentals and Programming in C (RMK). Pearson Education India. Retrieved from <https://www.oreilly.com/library/view/computer-fundamentals-and/9789332579200>
- ▶ yet another insignificant Programming Notes. (2021, April 08). Retrieved from <https://www3.ntu.edu.sg/home/ehchua/programming/index.html#Cpp>