

Algoritmalar ve Programlama I – BLM 103

Hafta 10: Diziler



Fenerbahçe Üniversitesi

10. Hafta İçeriği

- Dizi (Array) Tanımlaması
- Dizi Parametresi
- String'ler
- Dizi ve İşaretçiler(Pointer) Arasındaki İlişki

Diziler

- Bellekteki ardışık adresleri nasıl gruplayabiliriz?
 - Bir metnin karakterleri
 - Çeşitli sayılar
- Bu yaklaşımla yapılabilir mi?
- Az sayı olduğunda belki...
 - Ancak ya 100 sayı olsaydı?
 - Bir döngü içerisinde farklı farklı sayılara erişmek isteseydik?
- C dilinde diziler denen kavram bu sorunlara çözüm getirmektedir.
- `int num[4];`
- Tanımlaması ile 4 adet tamsayının tanımlaması yapılmıştır.

```
int num0;  
int num1;  
int num2;  
int num3;
```

`num[0], num[1], num[2], num[3].`

Dizilerin Kullanımı

- Tanımlama

- `tür değişken[eleman_sayisi];`

Dizi Türü, Tüm elemanlar aynı türden olmak zorundadır.

Eleman sayısı

- Dizi'nin elemanlarına ulaşmak için

- `değişken[index];`

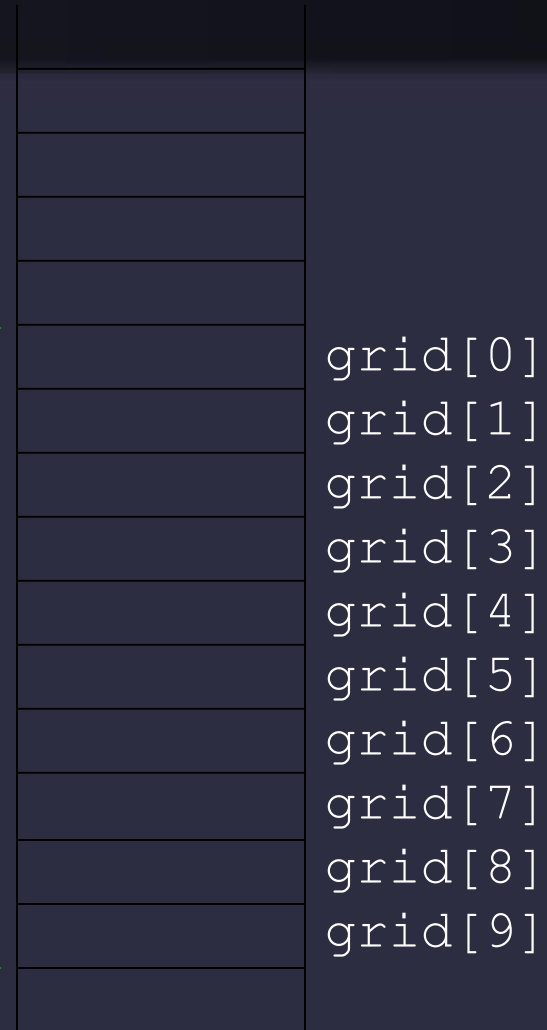
İndex isimli, değişkenin içerisindeki değere göre, ***değişken isimli dizinin, index' nolu elemanına erişim sağlanır***

Diziler

- Diziler bellekte ardışık olarak tutulurlar.

```
int grid[10];
```

- Dizinin ilk elemanı (`grid[0]`) dizinin başlangıç yerinden başlar.



Diziler

```
x = grid[3] + 1  
grid[6] = 5;
```

	x
	grid[0]
	grid[1]
	grid[2]
	grid[3]
	grid[4]
	grid[5]
	grid[6]
	grid[7]
	grid[8]
	grid[9]

Diziler

```
grid[x+1] = grid[x] + 2
```

x
grid[0]
grid[1]
grid[2]
grid[3]
grid[4]
grid[5]
grid[6]
grid[7]
grid[8]
grid[9]

Argüman olarak Dizinin Gönderilmesi

- C dilinde bir dizinin fonksiyona gönderilmesi esnasında
 - Dizinin ilk elemanının adresi gönderilir.
 - Yani fonksiyonda, dizinin kopyasıyla değil, kendisi ile işlem yapılır.

```
• main() {  
•   int numbers[5];  
•   ...  
•   mean = Average(numbers);  
•   ...  
• }  
• int Average(int inputValues[5]) {  
•   ...  
•   for (index = 0; index < 5; index++)  
•       sum = sum + inputValues[index];  
•   return (sum / 5);  
• }
```


Argüman olarak Dizinin Gönderilmesi Örnek

```
#include<stdio.h>
void abc(int abc[5]){

    for(int i=0;i<5;i++)
        abc[i] =abc[i] +1;
}
int main()
{
    int test [5] = {3,4,5,6,7};
    abc(test);

    for(int i=0;i<5;i++)
        printf("%d\n",test[i]);
    return 0;
}
```

Karakter dizileri

- Metin (string) saklamak için karakter dizileri kullanılabilir:

- `char outputString[16];`

- Metin'in içerisine atama yapmak için:

- `char outputString[16] = "Result = ";`

- ...diğer bir ifade ile:

- `outputString[0] = 'R';`
`outputString[1] = 'e';`
`outputString[2] = 's';`
`...`

Metinler ile Giriş/Çıkış

- Ekrana bastırmak ve kullanıcıdan metin formatında değer almak için "%s" kullanılır. Bir metnin son karakteri '\0' dır.
- Printf – \0 karakterini görene kadar ekrana basar
- Scanf – ilk boşluk karakterini görene kadar alır ve sonuna \0 karakteri ekleyerek diziye yazar.

```
printf("%s", outputString);
```

```
scanf("%s", inputString);
```

Diziler ve İşaretçiler Arasındaki İlişki

- Bir dizi tanımlandığında, aslında onun ismi, dizinin ilk elemanının adresini tutmaktadır.

```
char word[10];  
char *cptr;  
  
cptr = word; /* word[0]'ın adresini tutar*/
```

İşaretçi ve Dizi Gösterimi

- Bir önceki slaytta anlatıldığı gibi, her bir satırdaki ifadeler aslında aynıdır:

- `cptr` `word` `&word[0]`
- `(cptr + n)` `word + n` `&word[n]`
- `*cptr` `*word` `word[0]`
- `*(cptr + n)` `*(word + n)` `word[n]`

2D Diziler

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<time.h>

int main(void) {
    srand(time(NULL));
    int numbers[5][5], i, j;
    for (i = 0; i < 5; i++) {
        for (j = 0; j < 5; j++) {
            numbers[i][j] = rand() % 10;
            printf("\nrandom_array[%d][%d] =%d", i, j, numbers[i][j]);
        }
    }
    getch();
    return 0;
}
```

Fonksiyon Dizi Argümanı

```
#include <stdio.h>

float calculateSum(float age[]);

int main() {
    float result, age[] = {23.4, 55, 22.6, 3, 40.5, 18};
    // age array is passed to calculateSum()
    result = calculateSum(age);
    printf("Result = %.2f", result);
    return 0;
}

float calculateSum(float age[]) {
    float sum = 0.0;
    for (int i = 0; i < 6; ++i) {
        sum += age[i];
    }
    return sum;
}
```

C Dilinde Diziler ile Alakalı Tehlikeler

- **Dizinin sınırlarını aşma**

- Derleme zamanında dizinin limitlerinin aşılp aşılmadığı kontrol edilmez. Dizi dışına eriştiğinizde uygulamanız çökebilir.

- ```
int array[10];
int i;
for (i = 0; i <= 10; i++) array[i] = 0;
```

- **Değişken boyutlu tanımlama**

- Derleme zamanında dizi boyutu tanımlanmalıdır. Değişken boyutlu olamaz.

```
void SomeFunction(int num_elements) {
 int temp[num_elements];
 ...
}
```