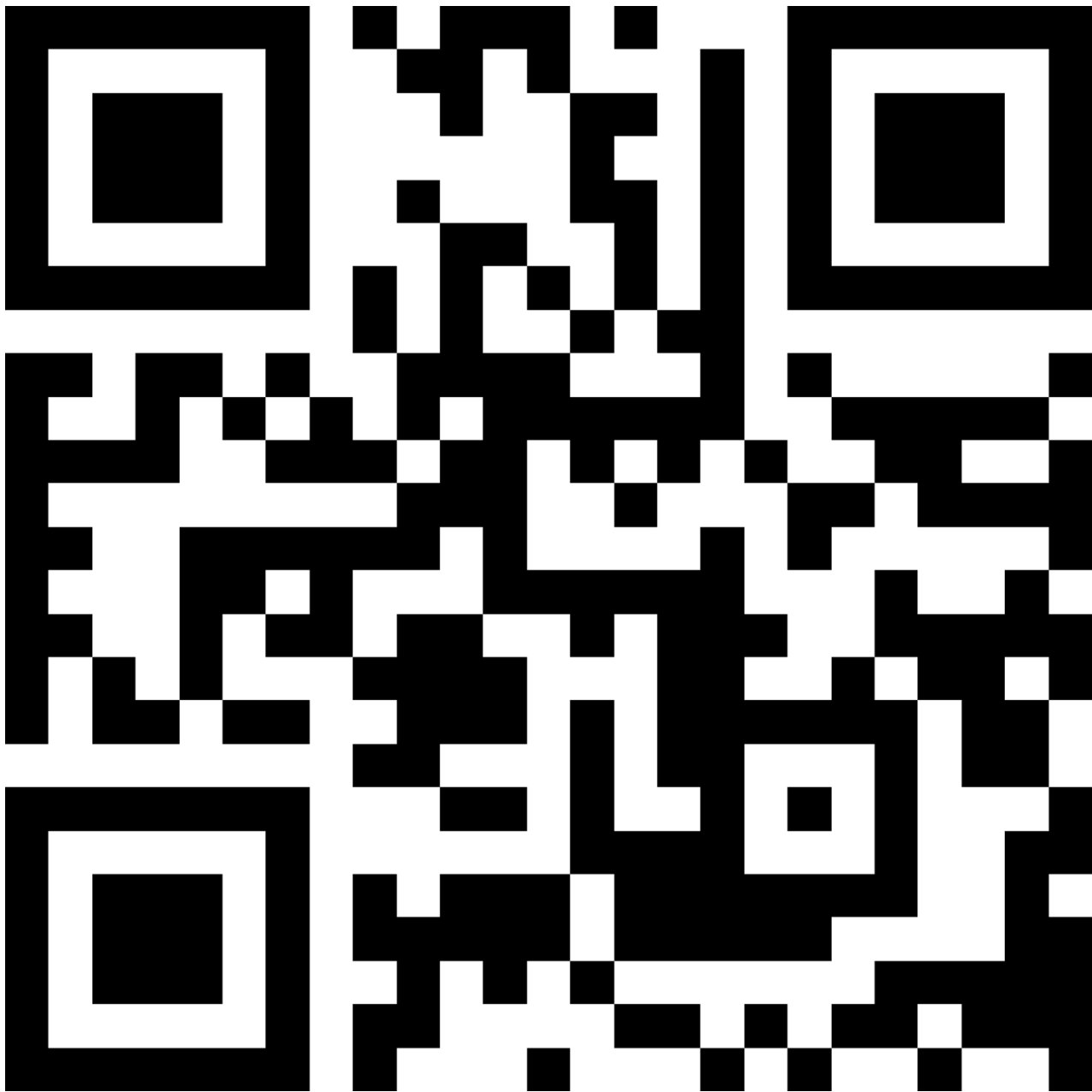


ALGORİTMALAR VE PROGRAMLAMA II

Hafta 5

String Fonksiyonlar



YOKLAMA



Thur

<https://tinyurl.com/algo-w5-t>

String

String NULL karakter '\0' ile biten bir karakter dizisidir.

Örnek: `char str[8];`

- ▶ En çok 8 karakter alabilen bir dizi oluşturur.
- ▶ Eğer `str` dizisi string olarak kullanılacak ise en fazla 7 karakter alabilir ve sonu NULL karakter '\0' ile bitmek zorundadır.

String İşlemleri

- ▶ C standard kütüphanesi stringleri manipüle etmek için birçok fonksiyon içerir.
- ▶ Bu fonksiyonları kullanmak için `<string.h>` ifadesini eklemeniz gerekir.

```
#include <string.h>
```

- ▶ Bazı önemli fonksiyonlar:
 - `strcpy(char *str1, const char *str2);`
 - `strlen(const char *str);`
 - `strcat(char *str1, const char *str2);`
 - `strcmp(const char *str1, const char *str2);`

Strcpy Fonksiyonu

```
strcpy(char *str1, const char *str2)
```

- `strcpy`, `str2` string'ini `str1` 'e kopyalar. `str1` 'in yeterli belleğe sahip olması gerekir.

```
char *strcpy(char *str1, const char *str2)
{
    char *p = str1;

    while (*str2)
        *p++ = *str2++;

    *p = '\0';
    return str1;
} /* end-strcpy */
```

Strcpy Fonksiyonu

```
#include <stdio.h>
#include <string.h>
int main () {
    char src[40];
    char dest[100];
    strcpy(src, "This is tutorialspoint.com");
    strcpy(dest, src);
    printf("Final copied string : %s\n", dest);
    return(0);
}
```

Output:

Final copied string : This is tutorialspoint.com



Strncpy Fonksiyonu

```
strncpy(char *str1, const char *str2, size_t n)
```

- `str2` string'inin ilk `n` karakterini `str1` 'e kopyalar.

```
#include <stdio.h>
#include <string.h>
int main()
{ char src[] = "Merhaba Dünya";
  char dest[20];
  strncpy(dest, src, 7);
  dest[7] = '\0';
  // strncpy eklediğinden Null karakter eklenir
  printf("Kopyalanan string: %s\n", dest); return 0;
}
```

Çıktı:

Kopyalanan string: Merhaba

```
dest[0] = 'M'
dest[1] = 'e'
dest[2] = 'r'
dest[3] = 'h'
dest[4] = 'a'
dest[5] = 'b'
dest[6] = 'a'
dest[7] = '\0' // Null terminator added here
```



Strlen Fonksiyonu

```
strlen(const char *str)
```

- String'in uzunluğunu döner (`\0` hariç).

```
int strlen(const char *str)
{
    int len = 0; // Initialize a variable to store the string length.

    while(*str++) // Loop through the string one character at a time.
        len++; // Increment the length each time a non-null character is encountered

    return len; // Return the final length of the string.
} /* end-strlen */
```


Strlen Fonksiyonu

```
#include <stdio.h>
#include <string.h>
```

```
int main () {
    char str[50]; int len;
    strcpy(str, "This is tutorialspoint.com");
    len = strlen(str);
    printf("Length of |%s| is |%d|\n", str, len);
    return(0);
}
```

Output:

Length of |This is tutorialspoint.com| is |26|

Strcat Fonksiyonu

```
strcat(char *str1, const char *str2)
```

- `str2` string'ini `str1` 'in sonuna ekler.

```
char *strcat(char *str1, const char *str2)
{
    char *p = str1;           // Pointer p initialized to the start of str1
    while(*p)                 // Traverse through str1 until the null character
        p++;                 // Move the pointer to the end of str1

    while(*str2)              // Traverse through str2 until the null character
        *p++ = *str2++;       // Copy each character from str2 to the end of str1

    *p = '\0';                // Null-terminate the concatenated string
    return str1;              // Return the pointer to the start of str1
}
```

Strcat Fonksiyonu

```
#include <stdio.h>
#include <string.h>

int main() {
    char src[50], dest[50];

    // Copy the string "This is source" into src
    strcpy(src, "This is source");

    // Copy the string "This is destination" into dest
    strcpy(dest, "This is destination");

    // Concatenate src to dest
    strcat(dest, src);

    // Print the final concatenated string
    printf("Final destination string : %s\n", dest);

    return 0;
}
```

Kodun çıktısı ne olur ?

Output:

Final destination string : |This is destinationThis is source|

Strncat Fonksiyonu

```
strncat(char *str1, const char *str2, size_t n)
```

- `str2` string'inin ilk `n` karakterini `str1`'in sonuna ekler.

```
#include <stdio.h>
#include <string.h>

int main() {
    // Declare and initialize the first string
    char str1[30] = "Merhaba ";

    // Declare and initialize the second string
    char str2[] = "Dünya";

    // Concatenate the first 3 characters of str2 to str1
    strncat(str1, str2, 3);

    // Print the concatenated result
    printf("Sonuç: %s\n", str1);

    return 0;
}
```

→ Kodun çıktısı ne olur ?

Output:

Sonuç: Merhaba Dün

Strcmp Fonksiyonu

```
strcmp(const char *str1, const char *str2)
```

- İki string'i karşılaştırır.

```
int strcmp(const char *str1, const char *str2)
{
    // Loop through both strings until a difference is found or both strings end
    while (*str1 && *str2 && *str1 == *str2) {
        str1++; // Move to the next character in str1
        str2++; // Move to the next character in str2
    }

    // Return the difference between the first unmatched character or null terminator
    return *str1 - *str2;
}
```

Strcmp Fonksiyonu

```
#include <stdio.h>
#include <string.h>

int main() {
    char str1[15], str2[15];
    int ret;

    strcpy(str1, "abcdef");
    strcpy(str2, "ABCDEF");

    ret = strcmp(str1, str2);

    if (ret < 0)
        printf("str1 is less than str2");
    else if (ret > 0)
        printf("str2 is less than str1");
    else
        printf("str1 is equal to str2");

    return 0;
}
```

Strncmp Fonksiyonu

```
strncmp(const char *str1, const char *str2, size_t n)
```

- İlk `n` karakteri karşılaştırır.

```
#include <stdio.h>
#include <string.h>

int main() {
    char str1[] = "Merhaba Dünya";
    char str2[] = "Merhaba Mars";

    if (strncmp(str1, str2, 7) == 0) {
        printf("İlk 7 karakter eşittir.\n");
    } else {
        printf("Farklıdır.\n");
    }

    return 0;
}
```

Strcasecmp Fonksiyonu

```
strcasecmp(const char *str1, const char *str2)
```

- Harf duyarsız karşılaştırma (Unix sistemlerinde).

```
#include <stdio.h>
#include <strings.h>

int main() {
    const char *str1 = "Merhaba";
    const char *str2 = "MERHABA";
    int result = strcasecmp(str1, str2);

    if (result == 0) {
        printf("İki string eşittir.\n");
    } else if (result < 0) {
        printf("str1, str2'den alfabetik olarak önce gelir.\n");
    } else {
        printf("str1, str2'den alfabetik olarak sonra gelir.\n");
    }

    return 0;
}
```

→ Çıktı ne olur?

Output: İki string eşittir.

Strchr Fonksiyonu

```
strchr(const char *str, int c)
```

- Belirli bir karakteri arar.

```
#include <stdio.h>
#include <string.h>

int main() {
    char str[] = "Merhaba Dünya";
    char *pos = strchr(str, 'D');

    if (pos != NULL) {
        printf("Karakter bulundu: %s\n", pos);
    }

    return 0;
}
```

Çıktı ne olur?

Output:
Karakter bulundu: Dünya

Strchr Fonksiyonu

```
strchr(const char *str, int c)
```

- Son geçen karakteri arar.
- string içinde aranan karakterin **son geçtiği yerin adresini** döner.
- Eğer karakter bulunamazsa NULL döner.
- Son geçtiği yerden itibaren string'in geri kalan kısmını gösterir.
- Bu fonksiyon, genellikle dosya uzantısı bulma gibi işlemlerde çok kullanışlıdır.

Strchr Fonksiyonu

```
strchr(const char *str, int c)
```

- Son geçen karakteri arar.

```
#include <stdio.h>
#include <string.h>

int main() {
    const char *str = "Merhaba dünya!";
    char ch = 'a';
    char *result = strchr(str, ch);

    if (result != NULL) {
        printf("%c karakterinin son geçtiği yer: %s\n", ch, result);
    } else {
        printf("Karakter bulunamadı.\n");
    }

    return 0;
}
```

Çıktı ne olur?

Output: 'a' karakterinin son geçtiği yer: a!

Strstr Fonksiyonu

- bir string içinde **alt string'i arar** ve bulunduğu yerin adresini döner.
- Eğer alt string bulunamazsa NULL döner.
- Bulunan kısımdan itibaren string'in geri kalanını gösterir.
- Bu fonksiyon, metin arama işlemlerinde çok kullanışlıdır.

Strstr Fonksiyonu

```
#include <stdio.h>
#include <string.h>

int main() {
    const char *text = "C programlama dilini öğreniyorum.";
    const char *search = "dil";
    char *result = strstr(text, search);

    if (result != NULL) {
        printf("'%s' ifadesi bulundu: %s\n", search, result);
    } else {
        printf("İfade bulunamadı.\n");
    }

    return 0;
}
```

Çıktı ne olur?

Output: 'dil' ifadesi bulundu: dilini öğreniyorum.

Strspn Fonksiyonu

```
strspn(const char *str1, const char *str2)
```

- `str1` 'in başında `str2` karakterlerinden oluşan kısmın uzunluğunu döner.
- string'in **başından itibaren**, belirtilen karakter kümesine **ait olan karakterleri sayar**.
- İlk uygun olmayan karaktere gelince durur.

```
#include <stdio.h>
#include <string.h>

int main() {
    const char *str = "12345abc678";
    const char *accept = "1234567890";

    // Using strspn to get the length of the initial segment of str that contains
    size_t result = strspn(str, accept);

    printf("Sadece rakamlardan oluşan kısmın uzunluğu: %zu\n", result);

    return 0;
}
```

only characters from accept

Çıktı ne olur?

Output: Sadece rakamlardan oluşan kısmın uzunluğu: 5

Atoi Fonksiyonu

- **ASCII to Integer** anlamına gelir.
- String içinde bulunan sayısal karakterleri tamsayıya çevirir.
- Eğer string sayısal olmayan karakter içeriyorsa, ilk sayısal olmayan karaktere kadar olan kısmı dönüştürür.

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

int main() {
    const char *str = "12345";
    int number;

    // Convert the string to an integer using atoi
    number = atoi(str);

    // Print the string and the converted integer
    printf("String olarak: %s\n", str);
    printf("Tamsayı olarak: %d\n", number);

    return 0;
}
```

Output:

String olarak: 12345

Tamsayı olarak: 12345

Çıktı ne olur ?

Atof Fonksiyonu

- **ASCII to Float** anlamına gelir.
- String içindeki **ondalıklı sayıyı, double türüne dönüştürür.**
- Eğer string, sayısal olmayan karakter içeriyorsa, ilk geçerli kısmı alır.
- **hatalı girişlerde 0 döner ve hata kontrolü yapmaz.**

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

int main() {
    const char *str = "3.14159";
    double number;

    number = atof(str);

    printf("String olarak: %s\n", str);
    printf("Ondalıklı sayı olarak: %f\n", number);

    return 0;
}
```

Output:

String olarak: 3.14159

Ondalıklı sayı olarak: 3.141590

Çıktı ne olur ?

Strtol Fonksiyonu

- string içindeki geçerli sayı kısmını dönüştürür ve geçersiz kısmın adresini endPtr üzerinden döner.

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

int main() {
    const char *str = "1234abc";
    char *endPtr;
    long number;

    // Convert the string to a long integer using strtol with base 10
    number = strtol(str, &endPtr, 10);

    // Print the converted number
    printf("Çevrilen sayı: %ld\n", number);

    // Check if there is any invalid part after the number
    if (*endPtr != '\0') {
        printf("Geçersiz kısım: %s\n", endPtr);
    }

    return 0;
}
```

Output:

Çevrilen sayı: 1234
Geçersiz kısım: abc

Çıktı ne olur?

Strtod Fonksiyonu

- **ASCII to Double** anlamına gelir.
- **Ondalıklı sayının geçerli kısmını çevirir ve geçersiz kısmı endPtr üzerinden döner.**

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

int main() {
    const char *str = "3.14159abc";
    char *endPtr;
    double number;

    // Convert the string to a double using strtod
    number = strtod(str, &endPtr);

    // Print the converted number
    printf("Çevrilen sayı: %lf\n", number);

    // Check if there is any invalid part after the number
    if (*endPtr != '\0') {
        printf("Geçersiz kısım: %s\n", endPtr);
    }

    return 0;
}
```

Output:


Çevrilen sayı: 3.141590
Geçersiz kısım: abc

Çıktı ne olur?

Doç. Dr. Caner ÖZCAN, KBÜ Yazılım Mühendisliği www.canerozcan.net*

- ▶ Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- ▶ Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- ▶ Paul J. Deitel, “C How to Program”, Harvey Deitel.
- ▶ “A book on C”, All Kelley, İra Pohl

* Bu dersin slaytları genelde bu kaynaktan türetilmiştir.



Ders içerikleri ve duyurular için dersin web sitesine
aşağıdaki adresten ulaşabilirsiniz.

hru-algpro.github.io

S o r u l a r

?



Dinlediğiniz için teşekkürler

