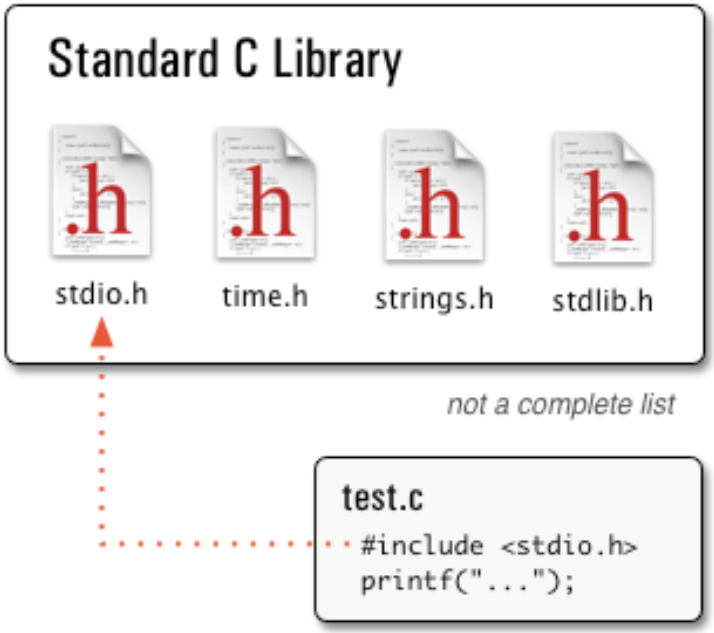
Hazır Fonksiyonlar

Suhap SAHIN
Onur GÖK

string kutuphaneleri



https://www.tutorialspoint.com/c_standard_library/ctype_h.htm

isdigit

```c
#include <stdio.h>
#include <ctype.h>
int main() {
    char c1;
                printf("karakter girin: ");
                scanf(" %c", &c1);
                if ( isdigit(c1) )
                                printf("%c karakteri rakamdir\n", c1);
                else
                                printf("%c karakter rakam degildir\n", c1);
                return 0;
}
```

ispunct

```c
#include <stdio.h>
#include <ctype.h>
int main() {
    char c1;
                printf("karakter girin: ");
                scanf(" %c", &c1);
                if ( ispunct(c1) )
                                printf("%c karakteri noktalama isaretidir\n", c1);
                else
                                printf("%c karakter noktalama isareti degildir\n", c1);
                return 0;
}
```

isalpha & (isupper, islower)

```c
#include <stdio.h>
#include <ctype.h>
int main() {
    char c1;
                printf("karakter girin: ");
                scanf(" %c", &c1);
                if ( isalpha(c1) ) {
                            if ( isupper(c1) ) // buyuk harf ise, kucuk harfi yaz
                              printf("kucuk harf karsiligi: %c\n", tolower(c1));
                            if ( islower(c1) ) // kucuk harf ise, buyuk harfi yaz
                              printf("buyuk harf karsiligi: %c\n", toupper(c1));
                }
                return 0;
}
```

String çevrime fonk.

atoi : string → integer

atol : string → long

atof : string → double

strtod(s, ptr) : s(string) → double

Çevrilemeyen ilk karakterin adresini ptr isimli pointera atar.

strtol(s, ptr) : s(string) → double

Çevrilemeyen ilk karakterin adresini ptr isimli pointera atar.

atoi & atol & atof

```c
#include <stdio.h>
#include <stdlib.h> // atio, atof, atol gibi fonksiyonlar stdlib'dedir
int main() {
        char s[50];
        int x;
        long y;
        double z;
        // kucuk sayilarla ve 15-20 basamakli sayilarla test edin
        // sonra kesirli sayilarla test edin
        printf("sayi girin: ");
        gets(s);

        x = atoi(s); // stringin icerigini integer'a cevirir
        y = atol(s); // stringin icerigini long'a cevirir
        z = atof(s); // stringin icerigini double'a cevirir

        printf("int: %d\n", x);
        printf("long: %ld\n", y);
        printf("double: %lf\n", z);
        return 0;
}
```

strtod & strtol

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

int main () {
   char str[30] = "20.30300 Bu kısım yazıdır";
   char *cevilemeyen_kisim;
   double cevrilen_kisim;

   cevrilen_kisim = strtod(str, &cevilemeyen_kisim);
   printf("Çevrilen kısım: |%f|\n", cevrilen_kisim);
   printf("Çevrilmeyen kısım: |%s|", cevilemeyen_kisim);

   return(0);
}
```

String karsılastırma fonk.

Kullanılısı;

strcmp(s1, s2)

Geri dönüs degeri;

s1 ve s2 string'lerini karsılastırır

Esitse 0

Alfabetik olarak s1 < s2 ise negatif sayı döndürür

Alfabetik olarak s1 > s2 ise negatif sayı döndürür

String karsılastırma fonk.

```c
#include <stdio.h>
#include <string.h>
int main() {
        char s1[100] = "aaaa";

        // yanlis kullanim, bu sekilde karsilastirma yapilamaz
        if (s1 == "aaaa")
                printf("esit\n");
        else
                printf("esit degil\n");

        printf("%u\n", s1);
        printf("%u\n", "aaaa");
        printf("\n");

        return 0;
}
```

strcmp

```c
#include <stdio.h>
#include <string.h>
int main() {
		char yazi1[10] = "test";
		char yazi2[10] = "test";
	printf("yazi1: %s\n", yazi1);
		printf("yazi2: %s\n",yazi2);
		if (yazi1 == yazi2)
		printf("yazi1 ve yazi2 esit\n");
	else
		printf("yazi1 ve yazi2 esit degil\n");
	printf("yazi1: %u\n", yazi1);
	printf("yazi2: %u\n", yazi2);
	int sonuc = strcmp(yazi1, yazi2);
	if (sonuc == 0)
		printf("yazi1 ve yazi2 esit\n");
	else if (sonuc < 0)
		printf("alfabetik olarak yazi1 < yazi2\n");
	else
		printf("alfabetik olarak yazi1 > yazi2\n");
	return 0;
}
```

strcmp

```c
#include <stdio.h>
#include <string.h>
int main() {
    int sonuc;
                sonuc = strcmp("telefon", "tablet");

                if (sonuc == 0)
                            printf("a ve b esit\n");
                else if (sonuc < 0)
                            printf("alfabetik olarak telefon <
                else
                            printf("alfabetik olarak telefon >
                printf("\n");

                return 0;
}
```

String kopyalama fonk.

strcpy(s2, s1): s1 stringini s2'ye kopyalar

strcat(s2, s1): s1 stringini s2'nin sonuna ekler

strcpy

```c
#include <stdio.h>
#include <string.h>
int main() {
		char s1[100] = "aa bbb cc";
		char s2[100];

		int karakter_sayisi = strlen(s1);
		printf("strlen(s1) : %d\n", karakter_sayisi

		// s1'i s2'ye kopyalar
		strcpy(s2, s1);

		printf("s1: %s\n", s1);
		printf("s2: %s\n", s2);


		return 0;
}
```

strcat

```
#include <stdio.h>
#include <string.h>
int main() {
		char s1[100] = "aa bbb cc";
		char s2[100];

		strcpy(s1, "test 1 ");
		strcpy(s2, "deneme 2 ");
		printf("s1: %s\n", s1);
		printf("s2: %s\n", s2);

		strcat(s1, s2);
		printf("strcat(s1, s2) sonrasinda\n");
		printf("s1: %s\n", s1);
		printf("s2: %s\n", s2);

		return 0;
}
```

strcpy

```c
#include <stdio.h>
#include <string.h>
int main() {

        char s1[100] = "aa bbb cc";
        char s2[100];

        strcpy(s1, "test 1 ");
        strcpy(s2, "deneme 2 ");
        strcpy(s1+strlen(s1), s2);

        printf("strcpy(s1+strlen(s1), s2) sonrasinda\n");
        printf("s1: %s\n", s1);
        printf("s2: %s\n", s2);

        return 0;
}
```

string arama

strchr(s, c) :

s içerisinde c karakterini arar.
Bulundugu yeri tutan isaretçiyi döndürür.
Yoksa **NULL** (0 adresi) pointer döndürür.

strstr(s1, s2) :

s1 içerisinde s2 stringini arar.
Bulundugu yeri tutan isaretçiyi döndürür.
Yoksa NULL pointer döndüdür.

strchr

```c
#include <stdio.h>
#include <string.h>
int main() {
		char cumle[100];
		char karakter;
		printf("cumle girin:\n");
		gets(cumle);
		printf("aranan karakteri girin:\n");
		scanf("%c", &karakter);
		char *ilk_gectigi_ptr = strchr(cumle, karakter);
		if (ilk_gectigi_ptr == NULL)
		printf("yok\n");
	else
		printf("ilk gectigi index: %d\n", ilk_gectigi_ptr-cumle);
	printf("|%c| den sonraki yazı - |%s|\n", karakter,ilk_gectigi_ptr);
	char *son_gectigi_ptr = strrchr(cumle, karakter);
	if (son_gectigi_ptr == NULL)
		printf("yok\n");
	else
		printf("son gectigi index: %d\n", son_gectigi_ptr-cumle);
		printf("|%c| den sonraki yazı - |%s|\n", karakter,son_gectigi_ptr);
	return 0;
}
```

strstr

```c
#include <stdio.h>
#include <string.h>
int main() {
		char cumle[200];
		char aranan[20];
		printf("cumle girin:\n");
		gets(cumle);
		printf("aranani girin:");
		scanf("%s", aranan);

		char *ara_ptr = strstr(cumle, aranan);
		if (ara_ptr == NULL)
			printf("\nyok\n");
		else
			printf("\nilk gectigi index: %d\n", ara_ptr-cumle);
		return 0;
}
```

strstr

```c
#include <stdio.h>
#include <string.h>
int main() {
		char cumle[200];
	char aranan[20];
		printf("cumle girin:\n");
		gets(cumle);
		printf("aranani girin:");
		scanf("%s", aranan);

		int sayac = 0;
		char *ara_ptr = strstr(cumle, aranan);

		while (ara_ptr != NULL) {
				printf("yer: %d\n", ara_ptr-cumle);
				sayac++;
				// bir sonraki karkaterden itibaren tekrar ariyoruz
				ara_ptr = strstr(ara_ptr+1, aranan);
		}
		printf("cumlede %d kere geciyor\n", sayac);

		return 0;
}
```

strtok

```c
#include <stdio.h>
#include <string.h>
int main() {

        char cumle[200] = "Bu ornek;noktalama isaretlerini ve bosluklari kullanarak cumleyi
                parcalar. Bu cumle,test icin yazilmistir.";

        const char *ayrac = " .,;"; // bosluk, nokta ve virgul, noktali virgul...

        char *kelime = strtok(cumle, ayrac);
        while (kelime != NULL) {
                printf("%s\n", kelime);

                kelime = strtok(NULL, ayrac);
        }

        return 0;
}
```

```
Bu
ornek
noktalama
isaretlerini
ve
bosluklari
kullanarak
cumleyi
parcalar
Bu
cumle
test
icin
yazilmistir
```

Dizilerde islem yapmak

memcpy(a, b, size):

b'yi a'ya kopyalar. size dizinin bellekteki boyutu.

memset :

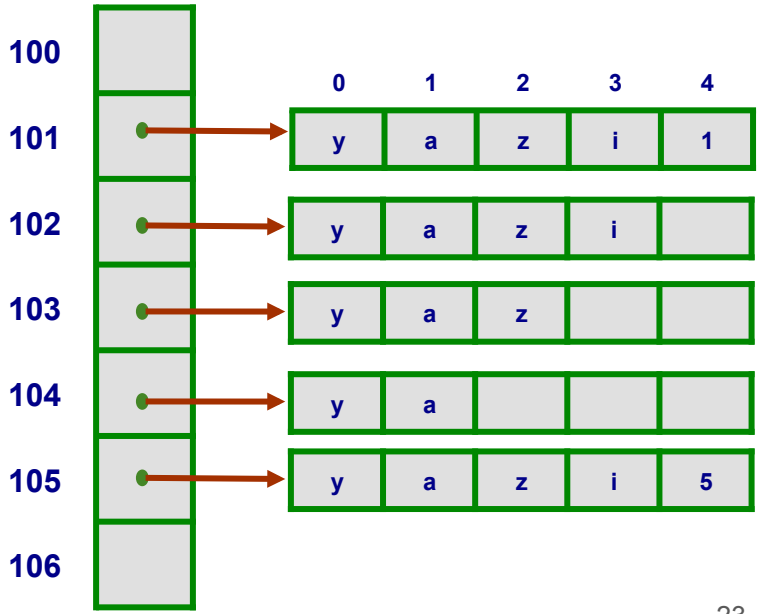
diziye char boyutunda sabit bir deger atar.

Dizilerde islem yapmak

```
#include <stdio.h>
#include <string.h>
int main() {
        int i;
        /************* string dizisi ****************/
        // 5 tane 20 karakterlik string dizisi
        char string_dizisi[5][20];

        for (i = 0 ; i < 5 ; i++) {
                printf("%d. kelimeyi girin: ",i+1);
                scanf("%s", string_dizisi[i]);
        }
        printf("girilen kelimeler:\n");
        for (i = 0 ; i < 5 ; i++)
                printf("%s\n", string_dizisi[i]);
        printf("\n");

        return 0;
}
```



memcmp

```c
#include <stdio.h>
#include <string.h>
int main() {
        /************* dizi karsilastirma ***************/
        int dizi_1[5] = {1,2,3,4,5};
        int dizi_2[5] = {1,2,3,4,5};

        if (dizi_1 == dizi_2) //HATA!!! bu sekilde kullanilamaz
                printf("esit\n");
        else
                printf("esit degil\n");

        int sonuc = memcmp(dizi_1, dizi_2, sizeof(int)*5);
        if (sonuc == 0)
                printf("esit\n");
        else
                printf("esit degil\n");
        printf("\n");

        return 0;
}
```

memcpy

```c
#include <stdio.h>
#include <string.h>
int main() {

		/************* dizi kopyalama ****************/
		int dizi_1[5] = {1,2,3,4,5};
		int dizi_3[9] = {0};

		// dizi_1'yi dizi_3'e kopyalar
		memcpy(dizi_3, dizi_1, sizeof(int)*5);

		// dizi_3'u ekrana yaz
		for (int i = 0 ; i < 9 ; i++)
				printf("%d ", dizi_3[i]);
		printf("\n\n");

		return 0;
}
```

memset

```c
#include <stdio.h>
#include <string.h>
int main() {

    /************* dizi sifirlama ****************/
                int dizi_1[5] = {1,2,3,4,5};

                // dizi_3'u sifirlar
                memset(dizi_1, 0, sizeof(int)*5);

                // dizi_3'u ekrana yaz
                for (int i = 0 ; i < 5 ; i++)
                                printf("%d ", dizi_1[i]);
                printf("\n\n");

                return 0;
}
```

memcpy

```c
#include <stdio.h>
#include <string.h>
int main() {
		int dizi_1[5] = {1,2,3,4,5};
		int dizi_3[9] = {0};

		/************* diziye ortadan kopyalama ****************/
		// dizi_3'un 3. elemanindan itibaren dizi_1'i atamak istiyoruz
		// dizi_3'un [2,7] arasina dizi_1'in [0,4] arasini atamak
		memcpy(dizi_3+2, dizi_1, sizeof(int)*5);

		// dizi_3'u ekrana yaz
		for (int i = 0 ; i < 9 ; i++)
				printf("%d ", dizi_3[i]);
		printf("\n\n");

		return 0;
}
```

Sorular