

## Ders 33

### Örnek stricmp

```
#define      SIZE      100

int mystricmp(const char* p1, const char* p2)
{
    while (toupper(*p1) == toupper(*p2)) {
        if (*p1 == '\0')
            return 0;
        ++p1, ++p2;
    }

    return toupper(*p1) - toupper(*p2);
}
```

Ödev ve mülakat sayısı

Ödev bu soru

```

pdev
I
#include <stdio.h>
#include <string.h>
#include "nutility.h"

#define SIZE 100

// boolean int
// raw loop yok
// strlen / strcmp

int ends_with(const char* p1, const char* p2);

int main(void)
{
    char s1[SIZE];
    char s2[SIZE];

    printf("birinci yaziyi girin: ");
    sgets(s1);
    printf("ikinci yaziyi girin: ");
    sgets(s2);

    if (ends_with(s1, s2)) {
        printf("evet var\n");
    }
    else {
        printf("hayir yok!\n");
    }
}

```

Raw loop yok strlen ve str cmp ile yaz.

Ödevi yaptım

```

#define _CRT_SECURE_NO_WARNING

#include <stdio.h>
#include "nutility.h"
#include <stdlib.h>
#include <string.h>

//tersten yazdırma

#define SIZE 5

int ends_with(const char* p1, const char *p2)
{
    size_t temp1 =strlen(p1);

```

```

        size_t temp2 = strlen(p2);
        return !(strcmp(p1 + (temp1 - temp2), p2));
    }

int main()
{
    char s1[100];
    char s2[100];

    printf("birinci yaziyi giriniz  ");
    sgets(s1);

    printf("\nikinci yaziyi giriniz  ");
    sgets(s2);

    if (ends_with(s1, s2))
        printf("evet yazi var\n");
    else
        printf("hayir yazi yok\n");
}

```

Örnek strpbrk bir grup karakteri var mı yok mu diye kontrol ediyor.

```

#include <stdio.h>
#include <string.h>
#include "nutility.h"

#define SIZE 100

int main(void)
{
    char str[SIZE];
    char letters[SIZE] = "aeoui";

    printf("bir yazi girin: ");
    sgets(str);

    char* p = strpbrk(str, letters);
    if (!p) {
        printf("bulunamadi\n");
    }
    else {
        printf("bulundu... yazinin %d indisli elemani\n", p - str);
    }
}

```

Microsoft Visual Studio Debug Console

```

bir yazi girin: kral ciplak!
bulundu... yazinin 2 indisli elemani

```

İlk bulunduğu yerin adresini yazıyor.

Rakamları bulalım.

```
int main(void)
{
    char str[SIZE];
    char digits[SIZE] = "0123456789";

    printf("bir yazi girin: ");
    sgets(str);

    char* p = strpbrk(str, digits);
    if (!p) {
        printf("bulunamadi\n");
    }
    else {
        printf("bulundu... yazinin %d indisli elemani\n", p - str);
    }
}
```

```
Microsoft Visual Studio Debug Console
bir yazi girin: ertan 28 suluagac
bulundu... yazinin 6 indisli elemani
```

Örnek kendi strpbrk kodumuzu yazalım bu belli karakterleri yazıda arıyor.

```
char* mystrpbrk(const char* p1, const char* p2)
{
    while (*p1) {
        if (strchr(p2, *p1) != NULL) {
            return (char *)p1;
        }
        ++p1;
    }

    return NULL;
}
```

////////////////////////////////////

STRING LİTERALLERİ

```

#include <stdio.h>
#include <string.h>
#include "nutility.h"

int main(void)
{
    putchar(*"ertan");
    putchar("volkan"[3]);
}

```

include "nutility.h"

Microsoft Visual Studio Debug Console

```

ek
D:\KURSLAR\NISAN_2022_C\Release\NISAN_2022_C.exe (process 28152) exited with code 0.
Press any key to close this window . . .

```

E VE K YAZDI ÇÜNKÜ "" ARASINDAKİ DİZİYE DÖNÜŞÜR VE ARADA KALAN YAZI DİZİNİN İLK ADRESİ OLUR.

```

// odev
#include <stdio.h>
#include <string.h>
#include "nutility.h"

int main(void)
{
    putchar(*"ertan");
    putchar("volkan"[3]);
    putchar('\n');
    printf("%p\n", "eray");
    printf("uzunluk = %zu\n", strlen("ilayda"));
}

```

SONDADA ADRES YAZDIK VE HATTA STRLEN İLE BOYUTU BULDUK DRES OLDU.

## TANIMSIZ DAVRANIŞ ÖRNEĞİ

STRING LİTERALİNİ DEĞİŞTİRMEK TANIMSIZ DAVRANIŞTIR.

```
int main(void)
{
    char* ptr = "selim";

    *ptr = 'k';    //u.b
    ptr[2] = 'r';
}
```

Char \*ptr ="selim" yazmanın çok sakıncası var.

Örnek asla yapılmaması lazım const char\* ı char \* a yoluluyoruz.

Kesinlikle const char \* olmalı.

```
void do_xyz_str(char* p);

int main(void)
{
    do_xyz_str("eray goku");
}
```

Örnek statik adres

```
char* weekday_str(int day_no)
{
    switch (day_no) {
        case 1: return "pazartesi";
        case 2: return "sali";
        case 3: return "carsamba";
        case 4: return "persembe";
        case 5: return "cuma";
        case 6: return "cumartesi";
        case 7: return "pazar";
    }

    return NULL;
}

int main(void)
{
    int day;

    printf("haftanin kacinci gunu: ");
    scanf("%d", &day);

    puts(weekday_str(day));
}
```

Statik ömürlü diye adres döndürmesi çok normal.

Soru 2 şimdide aynı adresmi döner

```
void func(void)
{
    char* p = "selami";
    printf("%p\n", p);
}

int main(void)
{
    for (int i = 0; i < 10; ++i) {
        func();
    }
}
```

Ben hayır çünkü otomatik öürlü dedim ama yanlış

P kendi adresini değil statik ömürlü selami adresini tutuyor her seferinde aynı adres olacak.

Önemli örnek

Doğru mu yazar yanlış mı



//unspecified behavior

```
int main(void)
{
    char* p1 = "murat";
    char* p2 = "murat";

    if (p1 == p2)
        printf("dogru\n");
    else
        printf("yanlis\n");
}
```

Unspecified behavior yani derleyiciye bağı

Önemli yukardakinin kuralı

Dikkat,

özdeş string literalleri  
derleyiciler tarafından

a) tek bir dizi olarak

b) farklı diziler olarak

Soru 4 bu bilinçli yazılmıştır.

```
//strcmp
//

int main(void)
{
    char str[100];
    //code

    if (str == "murat") {
        I
    }
}
```

Burada always false bu iki dizi adresi bilinçli yazılamaz.

```
int main(void)
{
    char s1[100];
    char s2[100];
    //code

    if (s1 == s2) { //always false
    }
}
```

Bu karşılaştırmada aynı aslında dizilerin içeriğini karşılaştırmak yapmak istemiş ama yanlış yapmış

Bunu strcmp ile yapmalıydı.

```

int main(void)
{
    char str[100];
    //code

    if (!strcmp(str, "ilayda"))

}

```

Asıl içerik karşılaştırması budur.

Soru 5 bilinçlimi yazılmış.

```

//

int main(void)
{
    char* p = "ilayda";

    //no code

    if (p == "ilayda") {

}

```

1. Kuralda dediğimiz gibi 2 string literalı aynıda olabilir farklıda olabilir bu derleyiciye bağlı biheviör ilk bakışta yanıltmak için soruluyor.
2. Aynıımı diye sakın adres karşılaştırmayın bu sıkıntı yaratır aynı adrestemi bilinmez.

```

if (p1 == p2) //bu iki adres eşit mi

if (!strcmp(p1, p2))

```

Bu ikisi başka şeyler.

```
/////////////////////////////////////////!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!/!1111111111111111//////////////////////////////////////////
```

## EN SIK SORULAN MÜLAKAT SORUSU

İKİSİ ARASINDAKİ FARK NEDİR.

```
int main(void)
{
    char s[] = "alim";
    char* p = "alim";
    //
}
```

AÇIKLAMA DEFTERDE.

### ÖRNEK ""

```
void func(const char* p)
{
}

int main(void)
{
    func("");
}
```

Yazı sonuna bişey yazmak istemiyorsa boş şekilde kullanılırım.

## Örnekler quiz

## 1 Ekranı ne yazar

```
int main(void)
{
    printf("%zu\n", strlen(""));
    printf("%zu\n", sizeof(""));
}
```

İlki 0 yazar yazı olmadığı için boyutu 0

İkincisi 1 yazar çünkü 1 elemanlı char dizinin byte alanı 1 kaplar.

## 2 Sentax hatası var mı

```
int main(void)
{
    int a[sizeof ""];
}
```

Hata yok sizeof "" boş dizinin boyutu sabittir.

## 3

```
int main(void)
{
    char *p = "bugun gunlerden
    persembe degi mi"
}
```

Bu şekilde alt satıra geçilemez. \ karakteri lazım

```
int main(void)
{
    const char* p = "bugun gunlerden \
                    persembe";
}
```

```
#include "nutility.h"
```

Microsoft Visual Studio Debug Console

```
bugun gunlerden                persembe

D:\KURSLAR\NISAN_2022_C\Release\NISAN_2022_C.exe (process 10676) exited with code 0.
Press any key to close this window . . .
```

Ancak bunda boşluklarda alınacak ve sıkıntı olacak satırı boşluksuz yazmalısın

Farklı bir yöntem daha var

```
int main(void)
{
    const char *p = "ahmet"
                  "mehmet"
                  "ilayda"
                  "metin";

    puts(p);
}
```

Microsoft Visual Studio Debug Console

```
ahmetmehmetilaydametin
```

Arada bir başka karakter yoksa derleyici buları birleştirir.

```
int main(void)
{
    const char* p = "ahmetmehmetilaydametin";
}
```

İki yazıda aynıdır derleyici tek bir char dizi yapar.birleştirir.

Bunun faydası nedir

```
int main(void)
{
    printf("[1] kayit ekle\n"
           "[2] kayit sil\n"
           "[3] kayit ara\n"
           "[4] kayit yenile\n"
           "[5] menuden cik\n"
           );
}
```

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

Microsoft Visual Studio Debug Console

```
[1] kayit ekle
[2] kayit sil
[3] kayit ara
[4] kayit yenile
[5] menuden cik
```

D:\KURSLAR\NISAN\_2022\_C\Release\NISAN\_2022\_C.exe (process 20608) exited with code 0.  
Press any key to close this window . . .

Ekrana istediğimiz ekrandaki yazıyı koddada görebiliyoruz.

Örnek ekrana ne yazar

```
int main(void)
{
    const char* p = "\\x42URS\\x41\\xA";

    printf("%zu\n", strlen(p));
}
```

Yazı aslında BURSA\n dir

Ekrana 6 yazar ordaki sayıların heisi bir karakter

Hex olarak yazmak

Ancak düzgün yazarsak olurdu.

```
int main(void)
{
    const char* p = "\x42ABAdede";
}
```

Bu geçerli karakter sabiti değil hex olarak yazmayı sürdürdük .

Octal ile baba yazalım.

```
int main(void)
{
    printf("\102102\101101");
}
```

Octal olarak \ ile 3 sayı yazılabilir gerisi değişiyor

Ekrana B102A101 YAZAR.

Pointer diziler

```
int *a[10];
int ival = 5;
int* ptr = &ival;
int ar[3] = { 1, 2, 3 };
```

```
a[0] = &ival;
a[3] = foo();
a[4] = ptr;
a[7] = NULL;
a[8] = ar;
```



Yukardaki atamaları yapabilirim

```
int x, y, z, t;

int main(void)
{
    int* a[10] = { &x, &y, &z, &t };
}
```

Soru dizinin son elemanının değeri

```
int x, y, z;

int main(void)
{
    int* a[4] = { &x, &y, &z };
}
```

Bence null pointer

Soru 2 kime atandı 99

```
int x, y, z;

int main(void)
{
    int a[] = { 10, 20, 30, 40 };
    int* p[4] = {&x, a, &y, &z };

    p[1][2] = 99;
}
```

P[1] = a dizi adresi demek

A[2] 3. Elemanı 99 yaptık.

```

int x, y, z;

int main(void)
{
    int a[] = { 10, 20, 30, 40 };
    int* p[4] = {&x, a, &y, &z };

    p[1][2]++;

    print_array(a, 4);

```

3. Eleman 1 artar burada sentax hatası yok aslında a[2]++ demek oda

\*(a+2)++ demek

Soru 3 zor bu kod derlenince neyapar.

```

int x, y, z;

int main(void)
{
    int a[] = { 10, 20, 30, 40 };
    int b[] = { -10, -20, -30, -40 };
    int c[] = { 1, 2, 3, 4 };
    int* p[] = { a, b, c };

    ++p[1];
    ++*p[1];

```

Bence b nin 2. Elemanı -19 olur.

Cevabım doğru

P[1] = b nin adresi

++b artık dizinin 2. Elemanı

\*p[1] = \*b

++\*b -19 oldu.

Soru en çok kullanılan örnek

```
int main(void)
{
    const char* pmons[] = {
        "Ocak",
        "Subat",
        "Mart",
        "Nisan",
        "Mayis",
        "Haziran",
        "Temmuz",
        "Agustos",
        "Eylul",
        "Ekim",
        "Kasim",
        "Aralik"
    };

    for (int i = 0; i < 12; ++i) {
        puts(pmons[i]);
    }
}
```

Sırayla ayları yazar.

```
int main(void)
{
    const char* pmons[] = {
        "Ocak",
        "Subat",
        "Mart",
        "Nisan",
        "Mayis",
        "Haziran",
        "Temmuz",
        "Agustos",
        "Eylul",
        "Ekim",
        "Kasim",
        "Aralik"
    };

    for (int i = 0; i < 12; ++i) {
        printf("%s\n", pmons[i]);
    }
}
```

Sırayla ayparı basar

%p olsa ne olurdu.

```
int main(void)
{
    const char* pmons[] = {
        "Ocak",
        "Subat",
        "Mart",
        "Nisan",
        "Mayis",
        "Haziran",
        "Temmuz",
        "Agustos",
        "Eylul",
        "Ekim",
        "Kasim",
        "Aralik"
    };

    for (int i = 0; i < 12; ++i) {
        printf("%p\n", pmons[i]);
    }
```

Yazıların tutulduğu adresleri yazardı.

%zu yazsaydık ne olurdu.

```
int main(void)
{
    const char* pmons[] = {
        "Ocak",
        "Subat",
        "Mart",
        "Nisan",
        "Mayis",
        "Haziran",
        "Temmuz",
        "Agustos",
        "Eylul",
        "Ekim",
        "Kasim",
        "Aralik"
    };

    for (int i = 0; i < 12; ++i) {
        printf("%s %zu\n", pmons[i], strlen(pmons[i]));
    }
```

Önce ayın ismi sonra uzunluğu yazılabilirdi.

Ne yaptık anlat bakalım aşağıda.

```
int main(void)
{
    const char* pmons[] = {
        "",
        "Ocak",
        "Subat",
        "Mart",
        "Nisan",
        "Mayis",
        "Haziran",
        "Temmuz",
        "Agustos",
        "Eylul",
        "Ekim",
        "Kasim",
        "Aralik"
    };

    int mon_idx;

    printf("yilin kacinci ayi : ");
    scanf("%d", &mon_idx);

    printf("yilin %d. ayi %s\n", mon_idx, pmons[mon_idx]);
}
```