

## Ders 39

Şaşırtıcı örnekler diziyi ekrana yazdıralım

```
int main(void)
{
    int a[10][20];

    randomize();
    for (int i = 0; i < 10; ++i) {
        for (int k = 0; k < 20; ++k) {
            printf("%d ", a[i][k] = rand() % 10);
        }
        printf("\n");
    }
}
```

klasik yazım yöntemi

Bu en basit olan yol

```
printf("\n\n\n");

for (int i = 0; i < 10; ++i) {
    print_array(a[i], 20);
    printf("\n");
}
```

Burada şaşırtıcı olan print array fonksiyonuna argüman göndererek yazmak oldu ancak buraki olay

A[i] yaptığımız zaman bu 20 elemanlı bir dizinin adresi oluyor.

Print array dizi adresi ve boyutu ile çağırılıyordu zaten bu yüzden sorunsuz tür uyumu sorunu olmadan ekrana yazı yazdık.

```
int size = 20;
int(*p)[20];

while (size--) {
    print_array(*p++, 20);
}
```

Diğer ekrana yazma yöntemi fakat burada bir hata var bu hata ne olabilir.

Hata aslında print\_aary yada \*p de değil necati bizi kandırdı.

Hata size 10 kere dönmesi gerekirken 20 yazmış

```
int size = 10;
int(*p)[20] = a;

while (size--) {
    print_array(*p++, 20);
}
```

Buradada ayı int \* [20] türüne gönderdik

Ancak direference edince bu \*p 10 elemanlı dizinin 20 elemanlı elemanı oldu

Yani 20 elemanlı içteki dizinin anderi oldu.

Bu adres türü int \* değil mi bu yüzden yine print\_array ile yazdırabildik.

```
int(*p)[20] = a;

for (int i = 0; i < 10; ++i) {
    for (int k = 0; k < 20; ++k) {
        printf("%d ", (*p)[k]);
    }
    printf("\n");
}
```

Aynı şeyi yaptık aslında normal bir dizide ar[k] dizinin elemanına ulaşıyor ar adresi

E bizimde ulaştığımız dizinin adresi \*p

\*p[k] ile yapmamıyıp (\*p)[k] yapma nedenimiz operatör önceliği

Bu sayededede çok boyutlu dizinin elemanlarına erişebiliyoruz.

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

Örnek a[10][20] ye rasgele değ er atayan fonksiyon yazalım.

```
void set_array_20(int(*p)[20], size_t size)
{
    for (size_t i = 0; i < size; ++i) {
        for (size_t k = 0; k < 20; ++k) {
            p[i][k] = (size_t)rand() % 10;
        }
    }
}

//void set_array_20(int(*p)[20], size_t size)
//{
//    while (size--) {
//        for (int i = 0; i < 20; ++i)
//            (*p)[i] = rand() % 10;
//        ++p;
//    }
//}
//

int main(void)
{
    int a[10][20];
    int b[4][20];

    randomize();

    set_array_20(p:a, size:10);
    set_array_20(p:b, size:4);
    //set_array_20(&a[0], 10);
}
```

Aynı fonksiyon üzerinden elemanları ekrana yazdıran fonksiyon

```

    }
}

void print_array_20(const int(*p)[20], size_t size)
{
    for (size_t i = 0; i < size; ++i) {
        for (size_t k = 0; k < 20; ++k) {
            printf(_Format: "%d", p[i][k]);
        }
    }
}

```

Tek yaptığımız printf eklemek hersey aynı

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

Örnek makro ile hileli olarak bütün matrisler için çalışabilecek fonksiyon yazalım

```

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

#define gmsf(sz) void set_matrix_##sz(int (*p)[sz], size_t size) \
{ \
    for (size_t i = 0; i < size; ++i) { \
        for (int k = 0; k < sz; ++k) { \
            p[i][k] = rand() % 10; \
        } \
    } \
}

gmsf(10)
gmsf(20)
gmsf(16)
gmsf(8)

int main()
{
    int a[20][10];
    int b[18][20];
    int c[4][16];
    int d[28][8];

    randomize();

    set_matrix_10(a, 20);
    set_matrix_20(b, 18);
    set_matrix_16(c, 4);
    set_matrix_8(d, 28);
}

```

Aslında bu kodda 4 farklı fonksiyon tanımladık.

##size yazarak size yerine farklı bir isim gelmesini sağladık ve tür ismindede aynısını yaptık.

Aslında derleyici burada 4 farklı fonksiyon görüyor.

Benim ön işlemci programım benim için farklı fonksiyonlar oluşturuyor fonksiyonel makrolar sayesinde

\ karakteride alt satırdan yazmamızı sağlar.

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

Örnek dizi bellekte sıralı olduğu için diziyi sıralı biçimde rasgele sayı atıyıp yazdıran fonksiyonu yazalım

```

{
    for (size_t i = 0; i < row * col; ++i)
        p[i] = rand() % 10;
}

//row 10 col 20
void print_matrix(const int* p, size_t row, size_t col)
{
    for (size_t i = 0; i < row; ++i) {
        for (size_t k = 0; k < col; ++k) {
            printf("%d", p[i * col + k]);
        }
        printf("\n");
    }
    pdline();
}

int main()
{
    int a[20][10];
    int b[18][20];
    int c[4][16];
    int d[28][8];

    randomize();

    set_matrix_random((int *)a, 20, 10);
    print_matrix((int *)a, 20, 10);

    set_matrix_random((int*)b, 18, 20);
    print_matrix((int*)b, 18, 20);

    set_matrix_random((int*)c, 4, 16);
    print_matrix((int*)c, 4, 16);
}

```

En üsteki kodda diziye ilk değer verirken adresini (int\*)a yaptık buda ilk elamanının ilk elemanı adresi demek yani int \* türünden bu yüzden fonksiyonda birkez direference edilebilir.

dizinin satır ve sütununu çarptık ve buradan toplam eleman sayısını elde ettik her tur bir arttırıp elemanları atadık

yazdırma fonksiyonuda bir nevi aynı

buradada 2 for kullandık nedeni 1 sonraki elemanı olan diziye alt satıra yazması.

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

## Çok boyutlu char diziler

```
char names[10][20] = { "erhan", "mustafa", "tahsin" };

e
r
h
a
n
\0
\0
\0
\0
\0
\0
\0
\0
\0
\0

m
u
s
t
a
f
a
\0
\0
```

İlk değer verme sentaxı bu şekilde verilmeyenler null karakter ve bellkete yine sıralı biçimde bulunur.

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

Soru 1 hata var mı

```

int main()
{
    char p[][20] = {
        "yeliz", "tansel", "gunay", "sumeyye", "nihal", "kerem", "ayla", "berk", "sezer", "samet",
        "cihat", "saniye", "naz", "ata", "cansu", "yasemin", "durmus", "refik", "rumeysa", "kelami",
        "atil", "taci", "akin", "berivan", "semsit", "hilmi", "sefer", "nasrullah", "tuncer", "polat",
        "kaya", "yelda", "asim", "saadet", "esra", "can", "umit", "turhan", "fazilet", "kayahan",
        "perihan", "pinat", "tayfun", "papatya", "turgut", "keriman", "emine", "niyazi", "sadiye", "lale",
        "rupen", "aynur", "kerim", "esin", "edip", "sidre", "melisa", "fuat", "kunter", "lamia",
        "hilal", "melek", "ceylan", "nalan", "jade", "kazim", "dost", "alican", "engin", "canan",
        "yurdagul", "nagehan", "zeliha", "tekin", "caner", "nazife", "ferhunde", "yurdanur", "zahide", "tanju",
        "celik", "pakize", "sadi", "izzet", "aslican", "osman", "abdi", "nisan", "abdulezziz", "gulsen",
        "adem", "fugen", "erdem", "sabriye", "tarik", "fahri", "sadettin", "polathan", "aslihan", "atalay",
        "feramuz", "dilek", "demir", "askin", "alparslan", "ceyda", "okan", "baran", "sami", "hulki",
        "cemal", "refika", "muruvvet", "beste", "sadri", "cihan", "cemre", "furkan", "seyhan", "busra",
        "hulusi", "petek", "ziya", "adnan", "kamile", "ercument", "mehmet", "zekai", "durriye", "sadegul",
        "cebrail", "yusuf", "sevim", "cemil", "emrecan", "ahmet", "kayhan", "pelinsu", "julide", "gurbuz",
        "ismail", "hakan", "nucacit", "emre", "selin", "nusret", "cahit", "sezen", "zahit", "egemen",
        "murat", "azmi", "ayse", "bekir", "ceyhan", "aylin", "tevfik", "diana", "nuriye", "suphi",
        "muhsin", "zarife", "dilber", "nevin", "tayyip", "binnaz", "helin", "melike", "yasin", "necmi",
        "emirhan", "nuri", "aziz", "sezai", "atif", "arda", "zubeyde", "kamil", "hasan", "ceyhan",
        "ciler", "cemile", "efecan", "necati", "aydan", "tayyar", "leyla", "onat", "cengiz", "bulent",
        "burhan", "sinem", "irmak", "hikmet", "mahir", "erhat", "cahide", "tansu", "huseyin",
        "bilge", "taner", "nahit", "aydan", "orkun", "ses", "iffet", "necmiye", "burak",
        "feraye", "galip", "cezmi", "ugur", "nurullah", "kenan", "suheyla", "murathan", "malik", "tugay",
        "ali", "cuneyt", "suleyman", "korhan", "utku", "nihat", "tijen", "olcay", "sevilay", "yasas",
        "zerrin", "hulya", "beril", "sade", "yilmaz", "ege", "derin", "bora", "tufan", "esen",
        "sarp", "eylul", "yurdakul", "muslum", "gizem", "agah", "atakan", "metin", "abdullah", "bilgin",
        "ediz", "gulsah", "nedim", "nazif", "belgin", "cesim", "ece", "cem", "dogan", "azize",
        "hakki", "tarcan", "derya", "kurthan", "mukerrem", "halime", "candan", "handan", "naci", "anil",
        "demet", "gazi", "efe", "nurdan", "tugra", "aytac", "kasim", "alev", "bennur", "gulden",
        "yunus", "billur", "tarkan", "recep", "fikret", "sevda", "melih", "necmettin", "hande", "teoman",
        "gursel", "aykut", "selenay", "ufuk", "soner", "muzaffer", "temel", "teslime", "enes", "gul",
        "naciye", "haldun", "yesim", "eda", "handesu", "binnur", "birhan", "yavuz", "figen", "nazli",
        "cetin", "devrim", "haluk", "devlet", "deniz", "mustafa", "sadullah", "celal", "pelin", "mert",
        "afacan", "poyraz", "menekse", "yalcin", "kezman", "garo", "tunc", "sefa", "fadime", "su",
        "tonguc", "kaan", "cumhur", "ilayda", "aysel", "tan", };
}

```

Hiç hata yok.

Soru2 boyutu ekrana yazdırabilir miyim saymadan evet.

Asize(p) ile yapabilirim.

```

int main()
{
    char p[][20] = { ... }

    printf("size p = %zu\n", asize(p));
}

```



Soru 3 her bir ismi ekrana yazan döngüyü nasıl yazarsınız.

```
int main()
{
    char p[][20] = { ... }

    for (size_t i = 0; i < asize(p); ++i) {
        printf("%s ", p[i]);
    }
}
```

Soru 4 budda tanımsız davranış varmı

```
int main()
{
    char p[][20] = { ... }

    for (size_t i = 0; i < asize(p); ++i) {
        p[i][0] = toupper(p[i][0]);
    }

    for (size_t i = 0; i < asize(p); ++i) {
        printf("%s ", p[i]);
    }
}
```

Microsoft Visual Studio Debug Console

Yeliz Tansel Gunay Sumeyye Nihal Kerem Ayla Berk Sezer Samet Cihat Saniye Naz Ata Cansu Yasem  
aci Akin Berivan Semsit Hilmi Sefer Nasrullah Tuncer Polat Kaya Yelda Asim Saadet Esra Can Um  
at Tayfun Papatya Turgut Keriman Emine Niyazi Sadiye Lale Rupen Aynur Kerim Esin Edip Sidre M  
eylan Nalan Jade Kazim Dost Alican Engin Canan Yurdagul Nagehan Zeliha Tekin Caner Nazife Fer

Hayır isimlerin il harfini büyük harf yaptım. Toupper ile

Soru 5 yazının ilk karakterini yazdırmak

```

int main()
{
    char p[][20] = { ... }

    for (size_t i = 0; i < asize(p); ++i) {
        putchar(p[i][0]);
    }

    printf("\n\n");

    for (size_t i = 0; i < asize(p); ++i) {
        putchar(*p[i]);
    }
}

```

İki kodda aynı işi yapıyor bakıldığında

`p[i]` ilk eleman adresi `*p[i]` adresin gösterdiği karakter oldu.

Ancak dizi `const` olsaydı değişiklik yapamazdık

Sadece okuma amacıyla olurduk.

Soru 6 fonksiyon ile ekrana yazdıralım.

```

void print_names(const char(*p)[20], size_t size)
{
    for (size_t i = 0; i < size; ++i) {
        printf("%s ", p[i]);
    }
    printf("\n");
}

int main()
{
    char p[][20] = { ... }

    print_names(p, asize(p));
}

```

Bende bu şekilde yaptım `p[i]` türü decay oluyor her atırma dizinin bir sonraki 20 elemanlı dizisinin adresi oluyor.

Soru 7 dizinin elemanları alfabetik sıra ile yazacak.

```
void print_names(const char(*p)[20], size_t size)
{
    for (size_t i = 0; i < size; ++i) {
        printf("%s ", p[i]);
    }
    printf("\n");
}

int main()
{
    char p[][20] = {
        "yeliz", "tansel", "gunay", "sumeyye", "nihal", "kerem", "ayla", "berk", "sezer", "samet",
        "cihat", "saniye", "naz", "ata", "cansu", "yasemin", "durmus", "refik", "rumeysa", "kelami",
    };

    //code
    print_names(p, asize(p));
}
```

Code yazan yere öyle bir kod CYAZACAĞIZ Kİ DİZİNİN ELEMANLARI ALFABETİK SIRAYLA YAZILACAK.

```
void print_names(const char(*p)[20], size_t size)
{
    for (size_t i = 0; i < size; ++i) {
        printf("%s ", p[i]);
    }
    printf("\n");
}

int main()
{
    char p[][20] = {
        "yeliz", "tansel", "gunay", "sumeyye", "nihal", "kerem", "ayla", "berk", "sezer", "samet",
        "cihat", "saniye", "naz", "ata", "cansu", "yasemin", "durmus", "refik", "rumeysa", "kelami",
    };

    for (size_t i = 0; i < asize(p) - 1; ++i) {
        for (size_t k = 0; k < asize(p) - 1 - i; ++k) {
            if (strcmp(p[k], p[k + 1]) > 0) {
                char temp[20];
                strcpy(temp, p[k]);
                strcpy(p[k], p[k + 1]);
                strcpy(p[k + 1], temp);
            }
        }
    }

    print_names(p, asize(p));
}
```

Bende hoca gibi yaptım buuble sort ve strcmp aynı ama ben swap işlemni generic swap ile yaptım.

Ancak buda iyi bir yöntem

Kopyalamada yazım kaybolmasın diye önce temp dizi tanımla

Temp diziye ilk diziyi ata

İlk diziye ikinci diziyi ata

2. diziye temp i ata srt cpy ile

```
void swap_str(char *p1, char *p2)
{
    static char temp[20];
    strcpy(_Destination:temp, _Source:p1);
    strcpy(_Destination:p1, _Source:p2);
    strcpy(_Destination:p2, _Source:temp);
}

int main()
{
    char p[][20] = {
        "yeliz", "tansel", "gunay", "sumeyye", "nihal", "kerem", "ayla", "berk", "sezer", "samet",
        "cihat", "saniye", "naz", "ata", "cansu", "yasemin", "durmus", "refik", "rumeysa", "kelami",
    };

    for (size_t i = 0; i < asize(p) - 1; ++i) {
        for (size_t k = 0; k < asize(p) - 1 - i; ++k) {
            if (strcmp(_Str1:p[k], _Str2:p[k + 1]) > 0) {
                swap_str(p1:p[k], p2:p[k + 1]);
            }
        }
    }
}
```

Burdada kendi swap fonksiyonumuzu yazdık.

Soru 8 ben öyle bir şey istiyorum ki benim char dizim değişmesin ama ben bu dizinin sıralı halini kullanabiliyim.

```
#include "utility.h"
```

```
int main()
{
    char names[][20] = { ... };
    char* p[asize(names)];

    for (size_t i = 0; i < asize(p); ++i) {
        p[i] = names[i];
    }

    for (size_t i = 0; i < asize(p) - 1; ++i) {
        for (size_t k = 0; k < asize(p) - 1 - i; ++k) {
            if (strcmp(p[k], p[k + 1]) > 0) {
                char* ptemp = p[k];
                p[k] = p[k + 1];
                p[k + 1] = ptemp;
            }
        }
    }

    for (size_t i = 0; i < asize(names); ++i) {
        printf("%s ", names[i]);
    }
    (void) getchar();
    system("cls");
    for (size_t i = 0; i < asize(p); ++i) {
        printf("%s ", p[i]);
    }
}
```

Eğer bu dizinin elemanlarının adresini bir pointer dizisine atarsam ve bu pointer dizinin gösterdiği elemanların adresini değiştirsem multi diziye dokunmadan sıralı haline ulaşabilirim.

Hocanın verdiği ödev strcmp yi qsort ile çağırmak.

```
int scmp(const void*vp1, const void*vp2)
{
    return strcmp(*(char**)vp1, *(char**)vp2);
}
```

