

Ders 36

Örnek memset

Dizinin elemanlarını sıfırlamak.

Bu işi standart c fonksiyonu yapabilir döngü değimi kullanma

Ben bunu ödevde yapmışım.

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

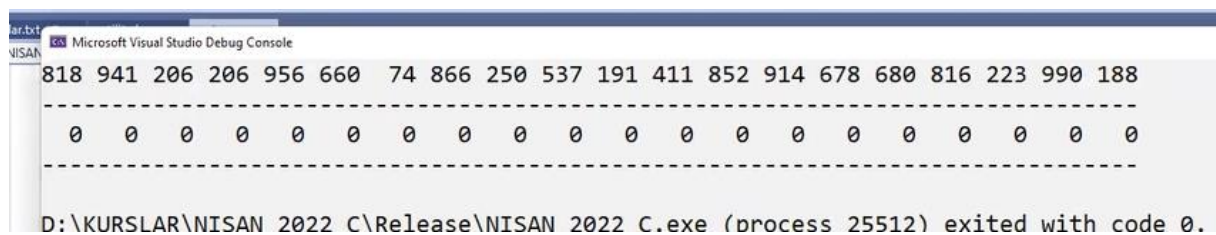
#define SIZE 20

int main(void)
{
    int a[SIZE];

    randomize();
    set_array_random(a, SIZE);
    print_array(a, SIZE);

    memset(a, 0, sizeof a);
    //memset(a, 0, SIZE * sizeof(int));

    print_array(a, SIZE);
}
```



```
ar.b.t
NISAN
Microsoft Visual Studio Debug Console
818 941 206 206 956 660 74 866 250 537 191 411 852 914 678 680 816 223 990 188
-----
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-----
D:\KURSLAR\NISAN 2022 C\Release\NISAN 2022 C.exe (process 25512) exited with code 0.
```

Boyutunu sizeof a ile çağırdık size ile eğil.

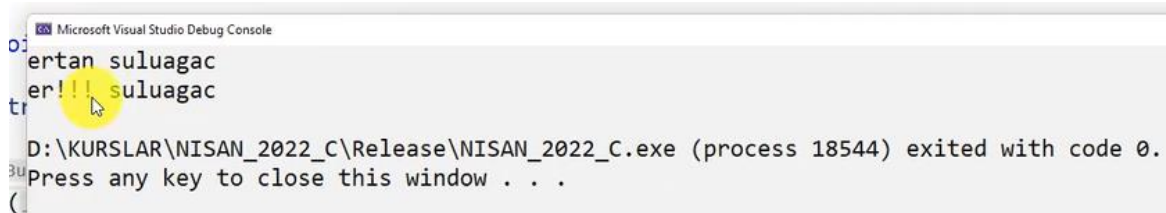
Örnek char dizi

Yazının belli yerlerine ??? yazmak.

```
#define SIZE 20

int main(void)
{
    char str[] = "ertan suluagac";

    puts(_Buffer: str);
    memset(_Dst: str + 2, _Val: '!', _Size: 3);
    puts(_Buffer: str);
}
```



```
Microsoft Visual Studio Debug Console
ertan suluagac
er!!! suluagac
D:\KURSLAR\NISAN_2022_C\Release\NISAN_2022_C.exe (process 18544) exited with code 0.
Press any key to close this window . . .
```

Farklı örnek arayıp bulma veya yazdırma

```
#define SIZE 20

int main(void)
{
    char str[] = "ertan suluagac";
    char* p = strstr(str, "tan");

    if (p) {
        memset(p, '?', 3);
        puts(p);
    }
}
```

Yazıyı arayıp bularakta yapabildik.

Örnek string literal

Kesinlikle kendin bi daha yap

```
printf("eski dosya ismi: ");
scanf("%s", old_file_name);

strcpy(new_file_name, old_file_name);

//eger dosyanın uzantısı yok ise
//dosyanın uzantısı .txt olacak
//dosyanın uzantısı jpeg ise dosya uzantısı png olarak degistirilecek
//eray.jpeg
//dosyanın uzantısı .xls ise dosyanın uzantısını silin
// hamdi.xls ==> hamdi

char* p = strrchr(new_file_name, '.');
if (!p) {
    strcat(new_file_name, ".txt");
}
else if (!strcmp(p, ".jpeg")) {
    strcpy(p, ".png");
}
else if (!strcmp(p, ".xls")) {
    *p = '\0';
    p[0] = '\0';
    strcpy(p, "");
}

printf("(%s) ==> (%s)\n", old_file_name, new_file_name);
}
```

```
printf("(%s) ==> (%s)\n", old_file_name, new_file_name);
puts(p + 1);
}
```

Buradada null dan sonarki xls i tekrar yazdırdık tamamen silinmedi.

Soru burada ne yapılıyor

String literali sayesinde kodda ne yapıldığını anlıyoruz.

Örek memset kodu yazılım.

```
#define SIZE 20

void* mymemset(void* vp, int val, size_t n)
{
    char* p = (char *)vp;

    while (n--) {
        *p++ = (char)val;
    }

    return vp;
}

int main(void)
```

Char *p ile her byte gönderdiğimiz değeri n kadar atadık bu kadar.

Soru 1 memset ekrana ne yazar.

```
int main(void)
{
    int x;

    memset(&x, 0, sizeof(x));
    printf("x = %d\n", x); //1
    memset(&x, 255, sizeof(x));
    printf("x = %d\n", x); //2
    memset(&x, 1, sizeof(x));
    printf("x = %d\n", x); //3
}
```

Burada olay her byte aynı değeri yazdırmak bu yüzden bytelar aşağıdaki gibi oluyor.

```
main.c

0000 0000
0000 0000
0000 0000
0000 0000

1111 1111
1111 1111
1111 1111
1111 1111

0000 0001 0000 0001 0000 0001 0000 0001
```

```
Microsoft Visual Studio Debug Console

X = 0
X = -1
X = 16843009

D:\KURSLAR\NISAN_2022_C\Release\NISAN_2022_C.exe (process 31552) exited
Press any key to close this window . . .
```

1. Bütün byteleri 1
2. Bütün byteleri 1 olan kod yani en büyük değer ama hepsi 1 -1 kuralı işaretli olduğu için
3. Sonuncuda son byte değeri 1 olacak.

Memcpy kendi fonsiyonumuz ile yaoalım.

```
#include "nutility.h"

#define SIZE 20

void* mymemcpy(void* vptest, const void* vpsource, size_t n)
{
    char* pdest = (char*)vptest;
    const char* psource = (const char*)vpsource;

    while (n--)
        *pdest++ = *psource++;

    return vptest;
}

int main(void)
```

Memcpy ile bir diziye başka bir diziye kopyalıyalım

Bu işlemi yapan fonksiyon var bu yüzden kendiniz kod yazmayın ödevde yaptığım gibi yapma memcpy ile yap

```
(Global Scope)

{
    char* pdest = (char*)vptest;
    const char* psource = (const char*)vpsource;

    while (n--)
        *pdest++ = *psource++;

    return vptest;
}

int main(void)
{
    int a[SIZE];
    int b[SIZE];

    randomize(a, SIZE);
    set_array_random(a, SIZE);
    print_array(a, SIZE);

    memcpy(b, a, sizeof a);
    print_array(b, SIZE);
}
```

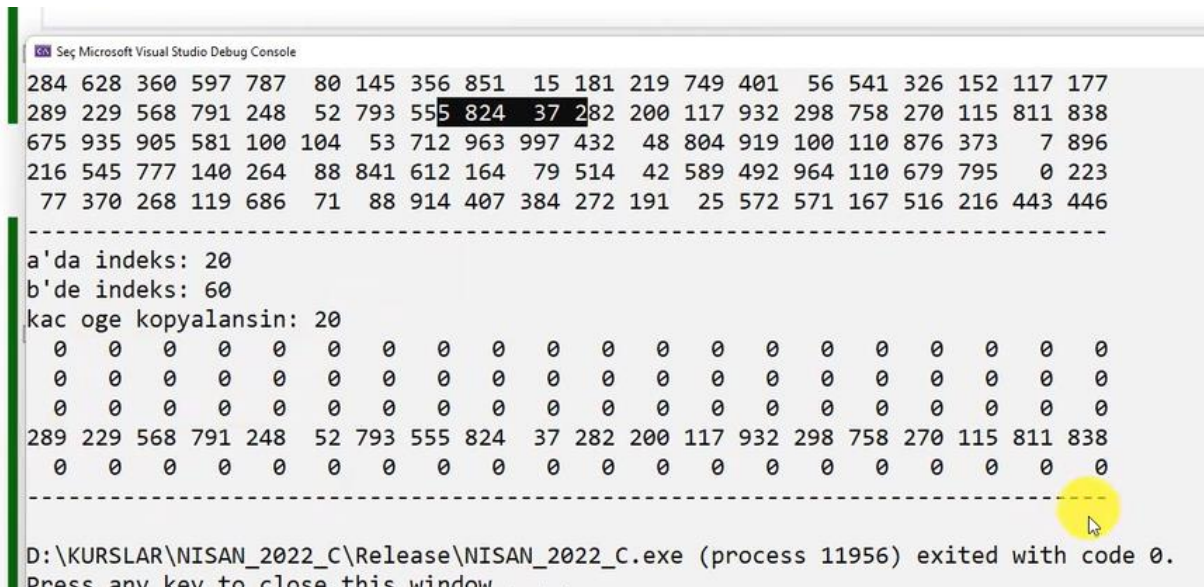
İstedğimiz indexten istediğimiz kadar değer kopyalıyalım.

```
randomize(a, SIZE);
set_array_random(a, SIZE);
print_array(a, SIZE);

int idx_a, idx_b;
size_t n;

printf("a'da indeks: ");
scanf("%d", &idx_a);
printf("b'de indeks: ");
scanf("%d", &idx_b);
printf("kac oge kopyalansin: ");
scanf("%zu", &n);

memcpy(b + idx_b, a + idx_a, n * sizeof(int));
print_array(b, SIZE);
```



```
Seç Microsoft Visual Studio Debug Console
284 628 360 597 787 80 145 356 851 15 181 219 749 401 56 541 326 152 117 177
289 229 568 791 248 52 793 555 824 37 282 200 117 932 298 758 270 115 811 838
675 935 905 581 100 104 53 712 963 997 432 48 804 919 100 110 876 373 7 896
216 545 777 140 264 88 841 612 164 79 514 42 589 492 964 110 679 795 0 223
77 370 268 119 686 71 88 914 407 384 272 191 25 572 571 167 516 216 443 446
-----
a'da indeks: 20
b'de indeks: 60
kac oge kopyalansin: 20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
289 229 568 791 248 52 793 555 824 37 282 200 117 932 298 758 270 115 811 838
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-----
D:\KURSLAR\NISAN_2022_C\Release\NISAN_2022_C.exe (process 11956) exited with code 0.
Press any key to close this window.
```

İndexleri ve kopyalanacak veri miktarını istedik ve yaptık.

Yazılar ile işlem

İlayda toprak yerine ilkurt toprak yazdıralım memcpy ile

```
int main(void)
{
    char s1[] = "ilayda toprak";
    char s2[] = "kenan kurt";

    memcpy(s1 + 2, s2 + 6, 4);

}
```

İlkurt toprak yazar

Strcpy ile yazsak ne olurdu.

```
int main(void)
{
    char s1[] = "ilayda toprak";
    char s2[] = "kenan kurt";

    strcpy(s1 + 2, s2 + 6);

    puts(s1);
}
```

Burada da ilkurt olurdu ancak sonuna null karakter koyardı ve toprak okunamazdı

Ekranda ilkurt yazardı.

Örnek ilayda toprak adını strcpy ile kendini kopyalasın!!!!!!!!!!!!!!!!!!!!111

```
int main(void)
{
    char s[] = "ilailayda toprak";

    strcpy(s + 3, s, 6);
```

Aynı bellek bloğunda olduğundan tanımsız davranış.

Memmove ile örnek bir önceki örneğin aksine tanımsız davranış değildir.

```
int main(void)
{
    char s[] = "ilayda toprak";

    memmove(s + 3, s, 6);

    puts(s);
```

Yukardaki örnek den farkı tanımsız davranış olmadı.

Kendi memmove fonksiyonumuz. Dakika 1.20.22

Ödev sorusu anlatımı.

```

char *str_prepend(char* pdest, const char* psource);

int main(void)
{
    char str[SIZE];
    char s[SIZE];

    printf("basina ekleme yapılacak yazi: "); //kurt
    scanf("%s", str);
    printf("eklenecek yazi: "); //kenan
    scanf("%s", s);
    str_prepend(str, s);

    printf("(s)\n", str); //(kenankurt)
}

```

Ödevin test kodudur

Benim ilk aklıma gelen memmove fonksiyonunu kullanarak yazmak yazı boyutu kadar memmove a kaydırma yaparım sonrada istediğim yazıyı yazarım.

Örnek memmove

```

#define SIZE 100

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

    puts(str);
    strcpy(str + 3, str);
    puts(str); //sevsevcan
}

```

Bu örneği strcpy ile yapamam ama memmove ile yapabilirim.

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

#define SIZE 100

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

    puts(str);
    memmove(str + 3, str, strlen(str) + 1);
    puts(str); //sevsevcan
}
```

Burdaki gibi

Memchr fonksiyonu bellek bloklarında byte değerlerini arar.

Neden kullanılır strchr ile null kontrolü yapmak zorunda kalmadan memchr kullanırız

Sadece yazı değil bütün değerler için kullanılır.

Örnek memchr ile a bulmak

```
int main(void)
{
    char str[SIZE];

    printf("bir yazi girin: ");
    sgets(str);
    char* p = (char *)memchr(str, 'a', 5);
    if (!p) {
        printf("bulunamadi\n");
    }
    else {
        printf("bulundu yazinin %d indeksli karakteri\n", p - str);
    }
}
```

Yazıda ilk bes karakterinde arama yaptık.

Strchr sonuna kadar bakardı.

Örnek dizide sayı değeri aramak

Char kullanma nedenimiz char 1 byte olması ama zorunlu değil.

```
#define SIZE 100

int main(void)
{
    unsigned char buffer[SIZE] = { 0 };

    //
    randomize();
    buffer[rand() % SIZE] = 1;

    unsigned char* p = memchr(buffer, 1, SIZE);
    if (!p) {
        printf("bulundu indeks %d\n", p - buffer);
    }
    else {
        printf("bulunamadi\n");
    }
}
```

Bu örnekte memset ile rasgele atama yapılan dizide rasgele bir indisine 1 değeri atayıp atanan indisi memchr fonksiyonu ile bulmaya çalışıyoruz.

Örnek memcpy

```

#include <string.h>
#include <stdlib.h>

#include "nutility.h"

#define SIZE 10

int main(void)
{
    int a[SIZE];
    int b[SIZE];

    randomize();
    set_array_random(a, SIZE);
    memcpy(b, a, sizeof a);

    if (!memcmp(a, b, sizeof a))
        printf("evet esit\n");
    else
        printf("hayir esit degil\n");
}

```

Dizi elemanını kopyalayıp bu işlemi yaptığım için hepsi eşit oldu.

Elemanlardan birini değiştirirsek artık değer 0 dönmez ve eşit değil yazısı çıkar.

İlginç örnek

```

int main(void)
{
    int x = -1;
    int y = 1;

    if (memcmp(&y, &x, sizeof(int)) > 0) {
        printf("%d > %d\n", y, x);
    }
}

```

Burada doğru kısma girmeden neden

Çünkü byte byte bakılıyor

0000 0001	1111 1111
0000 0000	1111 1111
0000 0000	1111 1111
0000 0000	1111 1111

1 değeri sağdaki -1 soldaki

Memcpy için -1 daha büyük görünüyor o yüzden büyük küçük sorgulaması yapılması sağlıklı değil.

Kendi memcpy kodumuzu yazalım.

```
int mymemcpy(const void* vp1, const void* vp2, size_t n)
{
    const unsigned char* p1 = vp1;
    const unsigned char* p2 = vp2;

    while (n-- > 0) {
        if (*p1 != *p2) {
            return *p1 > *p2 ? 1 : -1;
        }
        ++p1, ++p2;
    }

    return 0;
}
```

Greverse fonksiyonu yazalım dizileri ters çeviren generek kod

```
// bir diziyi reverse edecek bir fonksiyonu (generic olarak) tanımlayabilir miyiz?
```

```
void* greverse(void* vpa, size_t size, size_t sz)
{
    char* p = (char*)vpa;
    for (size_t i = 0; i < size / 2; ++i) {
        //p[i] p[size - 1 - i]
        gswap(p + i * sz, p + (size - 1 - i) * sz, sz);
    }

    return vpa;
}
```

```
int main(void)
{
    int a[SIZE];

    randomize();
    set_array_random(a, SIZE);
    print_array(a, SIZE);
    greverse(a, SIZE, sizeof(int));
    print_array(a, SIZE);

    double b[] = { 1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7 };
    greverse(b, asize(b), sizeof(*b));

    for (size_t i = 0; i < asize(b); ++i)
        printf("%f\n", b[i]);
}
```

Başta yazdığımız gswap fonksiyonu ile her turda dizi elemanlarını değiştirdik ancak pointer aritmetiğinden yaralanmamız lazım ama char olan pointer değeri int değer kadar arttırmam gerekiyor bu yüzden her seferinde 1 arttırmak istediğimizde sz yani int byte sayısı çarpı i yapıyoruz.

2. yazım yöntemi

```
// soru
// bir diziyi reverse edecek bir fonksiyonu (generic olarak) tanımlayabilir miyiz?

void* greverse(void* vpa, size_t size, size_t sz)
{
    char* pf = (char*)vpa;
    char* pe = pf + (size - 1) * sz;

    while (pf < pe) {
        gswap(pf, pe, sz);
        pf += sz;
        pe -= sz;
    }
}
```

Daha klas bir yazım yöntemi bu direkt ilk ve son eleman adresini kullandık.

Örnek dizide generik arama fonsiyonu.

```
7
3 #define      SIZE      20
3
3 void* g_search(const void* vpa, size_t size, size_t sz, const void* vpkey)
1 {
2     const char* p = vpa;
3
4     while (size--) {
5         if (!memcmp(p, vpkey, sz))
6             return (char *)p;
7         p += sz;
8     }
9
10    return NULL;
11
12 }
```

```
int main(void)
{
    int a[SIZE];

    randomize();
    set_array_random(a, SIZE);
    print_array(a, SIZE);

    int sval;

    printf("aranacak degeri giriniz: ");
    scanf("%d", &sval);

    int* p = (int *)g_search(a, SIZE, sizeof(*a), &sval);
    if (!p) {
        printf("bulunamadi\n");
    }
    else {
        printf("bulundu...dizinin %d indisli elemani\n", p - a);
    }
}
```

```
nt main(void)
Microsoft Visual Studio Debug Console
408 838 959 487 777 894 766 379 900 789 726 464 993 530 809 701 92 722 476 712
-----
aranacak degeri giriniz: 476
bulundu...dizinin 18 indisli elemani

D:\KURSLAR\NISAN_2022_C\Release\NISAN_2022_C.exe (process 33280) exited with code 0.
Press any key to close this window . . .
```

Memcpy kullanarak zaten bellek blokları kontrol ediliyordu. Bizde fonksiyonda bundan yaralandık.

Dizinin adresini char değere atadım her turda memcpy ile sz yani boyutu bilgisi kadar eşitlik soruladı

Eğer eşitse p adresi döner

Ancak eşit değilse null pointer döner