

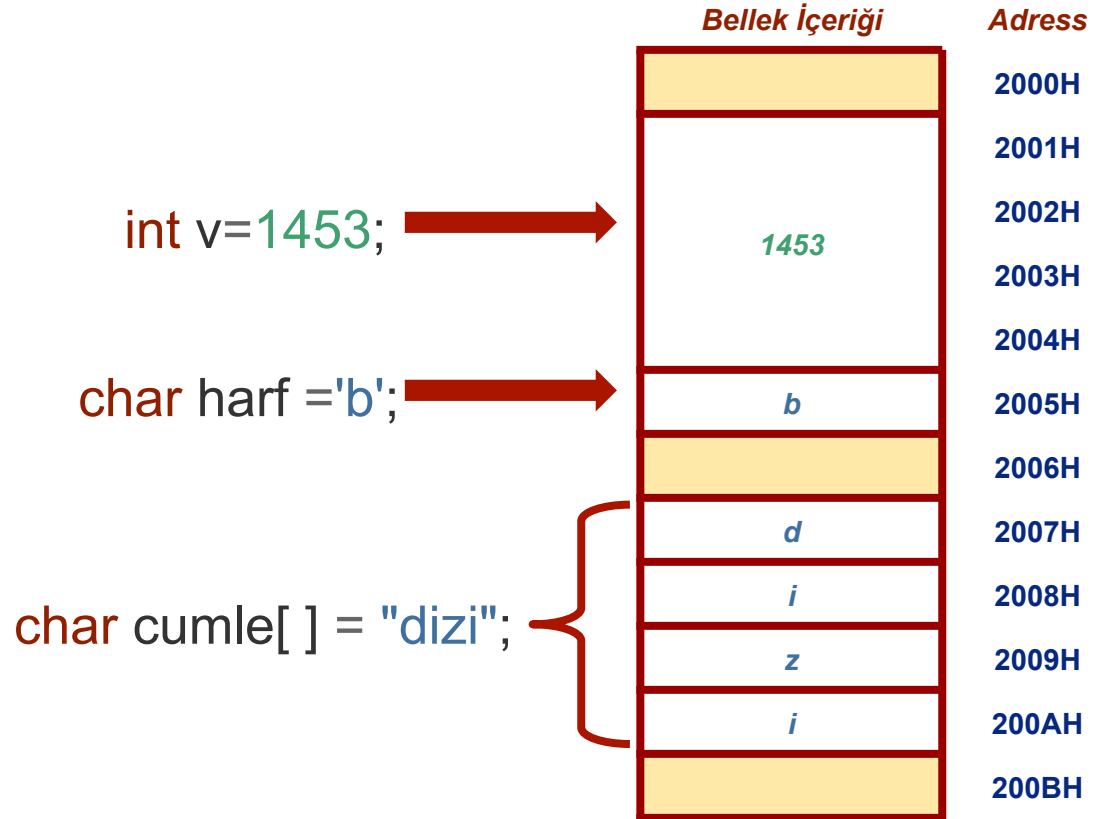
Struct



```
struct number {  
    int img;  
    float real;  
};
```

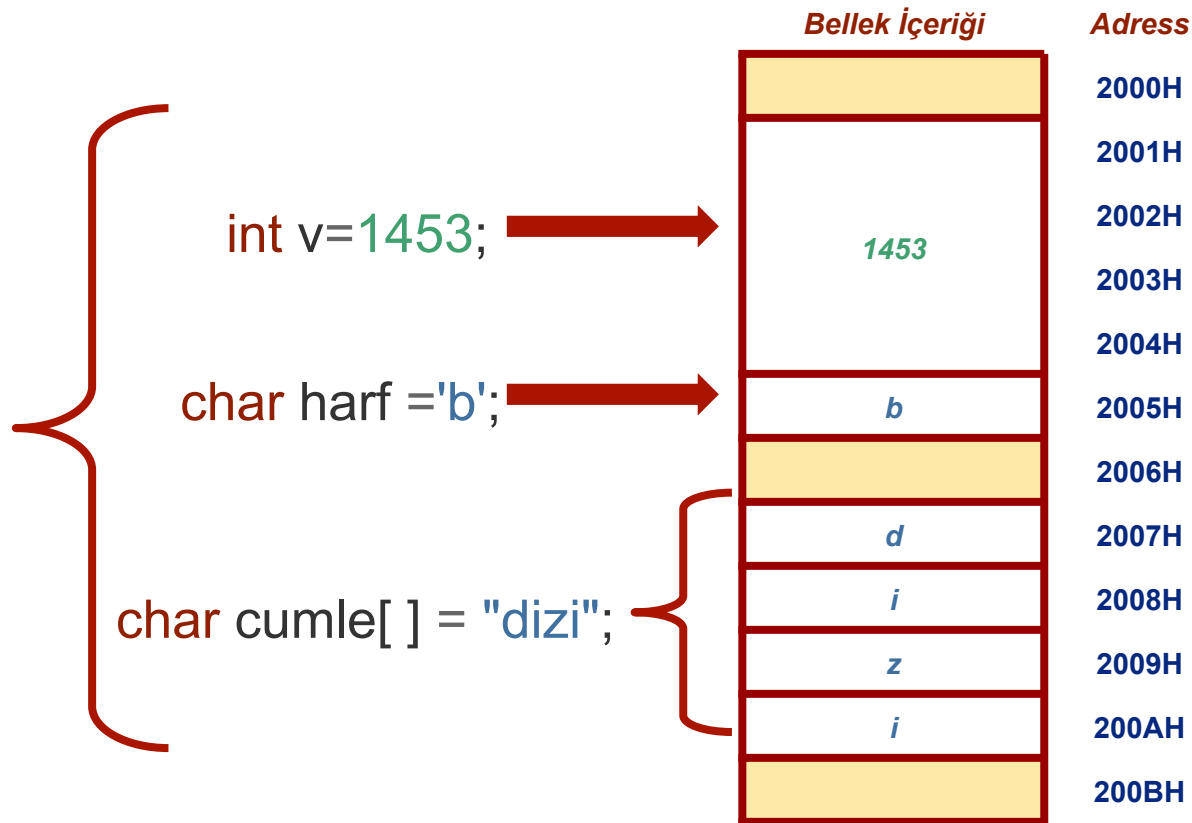
Suhap SAHIN
Onur GÖK

Hafıza Yerlesimleri



Hafıza Yerlesimleri

```
struct HafizaBirimleri {  
    int v=1453;  
    char harf ='b';  
    char cumle[ ] =  
    "dizi";  
};
```



Zaman islemleri

Saat

Dakika

Saniye



Zaman islemleri

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

    int saat1, saat2;
    int dakika1, dakika2;
    int saniye1, saniye2;

    printf("baslangic zamani girin(saat dakika saniye): ");
    scanf("%d %d %d", &saat1, &dakika1, &saniye1);

    printf("bitis zamani girin(saat dakika saniye): ");
    scanf("%d %d %d", &saat2, &dakika2, &saniye2);

    int saat_farki = saat2 - saat1;
    printf("saat farki : %d\n", saat_farki);
    return 0;
}
```

Zaman islemleri

```
#include <stdio.h>
#include <stdlib.h>
struct Zaman {
    int saat;
    int dakika;
    int saniye;
};

int main() {

    struct Zaman z1;

    // yapinin elemanlarina deger atama islemi
    z1.saat = 7;
    z1.dakika = 18;
    z1.saniye = 52;

    // degerlerin kullanimi
    printf("%d %d %d\n", z1.saat, z1.dakika, z1.saniye);
}
```

Zaman islemleri

```
#include <stdio.h>
#include <stdlib.h>
struct Zaman {
    int saat;
    int dakika;
    int saniye;
};

int main() {
    struct Zaman z1;
    struct Zaman z2;

    printf("baslangic zamani girin : ");
    scanf("%d %d %d", &z1.saat, &z1.dakika, &z1.saniye);

    printf("bitis zamani girin : ");
    scanf("%d %d %d", &z2.saat, &z2.dakika, &z2.saniye);

    int saat_farki = z2.saat - z1.saat;
    printf("saat farki : %d\n", saat_farki);
    return 0;
}
```

Baslangic Deger Atamasi

```
#include <stdio.h>
#include <stdlib.h>
struct Zaman {
    int saat;
    int dakika;
    int saniye;
};
int main()
{
    /// Zaman tipinde degisken tanimlama
    /// baslangic degeri atama
    struct Zaman z = { 14, 43, 3 };

    /// degerlerin kullanimi
    printf("z1.saat : %d\n", z.saat);
    printf("z1.dakika : %d\n", z.dakika);
    printf("z1.saniye : %d\n", z.saniye);
    printf("\n");

    return 0;
}
```


Baslangic Deger Atamasi

```
#include <stdio.h>
#include <stdlib.h>
struct Zaman {
    int saat;
    int dakika;
    int saniye;
};

int main(){

    /// degelerin tamamini atamadan tanimlanmasi
    struct Zaman z2 = { 15 };

    printf("z2.saat : %d\n", z2.saat);
    printf("z2.dakika : %d\n", z2.dakika);
    printf("z2.saniye : %d\n", z2.saniye);
    printf("\n");

    return 0;
}
```

Baslangic Deger Atamasi

```
#include <stdio.h>
#include <stdlib.h>
struct Zaman {
    int saat;
    int dakika;
    int saniye;
};

int main(){

    /// deger atamasi yapmadan tanimlanmasi
    struct Zaman z3;
    printf("z3.saat : %d\n", z3.saat);
    printf("z3.dakika : %d\n", z3.dakika);
    printf("z3.saniye : %d\n", z3.saniye);
    printf("\n");

    return 0;
}
```

Öğrencilere Ait Bilgiler

```
#include <stdio.h>
#include <stdlib.h>
struct Ogrenci {
    Ogrenci : ad soyad ogrNo kayitYili TCKimlik
    char ad[30];
    char soyad[30];
    int ogrenciNo;
    int kayitYili;
    char TCKimlik[12];
};

int main(){
    struct Ogrenci liste[100];
    strcpy(liste[0].ad, "deneme");
    adina "deneme" ata
    liste[0].kayitYili = 2015;
    yilina 2015 ata
    liste[0].ogrenciNo = 123;
    numarasina 123 ata
    printf("%s %d\n", liste[0].ad, liste[0].ogrenciNo);
    return 0;
}
```

// ilk ogrencinin

// ilk ogrencinin kayit

// ilk ogrencinin

// ilk ogrencinin adini ve numarasini yaz

Tarih Olusturma

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct Tarih {
    int gun;
    int ay;
    int yil;
};

struct Tarih Tarih_olustur(int g, int a, int y) {
    struct Tarih t;
    if (g > 31 || a > 12) {
        t.gun = 0;
        t.ay = 0;
        t.yil = 0;
    } else {
        t.gun = g;
        t.ay = a;
        t.yil = y;
    }
    return t;
}

int main(){
    struct Tarih t1;
    t1 = Tarih_olustur(10,15,2015);
    printf("%d %d %d\n", t1.gun, t1.ay, t1.yil);
    struct Tarih t5 = Tarih_olustur(3, 3, 2016);
    printf("%d %d %d\n", t5.gun, t5.ay, t5.yil);
    return 0;
}
```

Tarih Karsilastirma

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct Tarih {
    int gun;
    int ay;
    int yil;
};

struct Tarih Tarih_olustur(int g, int a, int y) {
    struct Tarih t;
    if (g > 31 || a > 12) {
        t.gun = 0;
        t.ay = 0;
        t.yil = 0;
    } else {
        t.gun = g;
        t.ay = a;
        t.yil = y;
    }
    return t;
}

int main(){
    struct Tarih t5 = Tarih_olustur(3, 3, 2016);
    struct Tarih t6 = Tarih_olustur(3, 3, 2016);
    // tum icerik tek tek karsilastirilabilir
    if (t5.yil == t6.yil && t5.ay == t6.ay && t5.gun == t6.gun)
        printf("t5 ve t6 esit\n");
    else
        printf("t5 ve t6 esit degil\n");
    return 0;
}
```

Tarih Karsilastirma

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct Tarih {
    int gun;
    int ay;
    int yil;
};

struct Tarih Tarih_olustur(int g, int a, int y) {
    struct Tarih t;
    if (g > 31 || a > 12) {
        t.gun = 0;
        t.ay = 0;
        t.yil = 0;
    } else {
        t.gun = g;
        t.ay = a;
        t.yil = y;
    }
    return t;
}

int main()
{
    struct Tarih t5 = Tarih_olustur(3, 3, 2016);
    struct Tarih t6 = Tarih_olustur(3, 3, 2016);
    if (memcmp(&t5, &t6, sizeof(struct Tarih)) == 0)
        printf("t5 ve t6 esit\n");
    else
        printf("t5 ve t6 esit degil\n");
    return 0;
}
```

Tarih Karsilastirma

```
#include <stdio.h>
struct Tarih {
    int gun;
    int ay;
    int yil;
};

struct Tarih Tarih_olustur(int g, int a, int y) {
    struct Tarih t;
    if (g > 31 || a > 12) {
        t.gun = 0;
        t.ay = 0;
        t.yil = 0;
    } else {
        t.gun = g;
        t.ay = a;
        t.yil = y;
    }
    return t;
}

int Tarih_karsilastir(struct Tarih t1, struct Tarih t2) {
    if (t1.yil > t2.yil)
        return 1;
    if (t1.yil == t2.yil && t1.ay > t2.ay)
        return 1;
    if (t1.yil == t2.yil && t1.ay == t2.ay && t1.gun > t2.gun)
        return 1;
    if (t1.yil == t2.yil && t1.ay == t2.ay && t1.gun == t2.gun)
        return 0;
    return -1;
}

int main()
{
    struct Tarih t5 = Tarih_olustur(3, 3, 2016);
    struct Tarih t6 = Tarih_olustur(3, 3, 2016);
    int sonuc = Tarih_karsilastir(t5, t6);
    if (sonuc == 1)
        printf("t5 > t6\n");
    else if (sonuc == 0)
        printf("t5 == t6\n");
    else
        printf("t5 < t6\n");
    return 0;
}
```

Tarih Karşılaştırma

```
int Tarih_karsilastir(struct Tarih t1, struct Tarih t2) {  
  
    int a = t1.yil * 365 + t1.ay * 30 + t1.gun;  
    int b = t2.yil * 365 + t2.ay * 30 + t2.gun;  
  
    if (a > b)  
        return 1;  
    else if (a == b)  
        return 0;  
    else  
        return -1;  
}
```


Bellek Alanı

```
#include <stdio.h>
#include <stdlib.h>
struct Tarih {
    int gun;
    int ay;
    int yil;
};
struct Ogrenci {
    char ad[30];
    char soyad[30];
    int ogrenciNo;
    int kayitYili;
    char TCKimlik[12];
    struct Tarih dogumTarihi;
    struct Tarih kayitTarihi;
};
int main(){
    printf("int : %d\n", sizeof(int));
    printf("Tarih : %d\n", sizeof(struct Tarih));
    printf("Ogrenci : %d\n", sizeof(struct Ogrenci));
    return 0;
}
```

Fonksiyon & Struct

```
#include <stdio.h>
#include <math.h>
struct Nokta {
    int x;
    int y;
};

double Nokta_uzaklik(struct Nokta n1, struct Nokta n2) {
    return sqrt(pow(n1.x-n2.x, 2) + pow(n1.y-n2.y, 2));
}

int main() {
    struct Nokta nokta1 = {1, 4};
    struct Nokta nokta2 = {1, 2};

    printf("nokta1 x,y : %d,%d\n", nokta1.x, nokta1.y);
    printf("nokta2 x,y : %d,%d\n\n", nokta2.x, nokta2.y);

    double uzaklik = Nokta_uzaklik(nokta1, nokta2);
    printf("nokta1 ve nokta2 arasindaki uzaklik: %f\n\n", uzaklik);

    return 0;
}
```

Fonksiyon & Struct

```
#include <stdio.h>
#include <math.h>
struct Nokta {
    int x;
    int y;
};

void Nokta_deger_ata_1(struct Nokta nokta, int x, int y) {
    nokta.x = x;
    nokta.y = y;
}

int main() {
    struct Nokta nokta1 = {1, 4};

    Nokta_deger_ata_1(nokta1, 100, 200);
    printf("deger_ata_1 fonksiyonu sonrasinda\n");
    printf("nokta1 x,y : %d,%d\n\n", nokta1.x, nokta1.y);

    return 0;
}
```

Fonksiyon & Struct

```
#include <stdio.h>
#include <math.h>
struct Nokta {
    int x;
    int y;
};
void Nokta_deger_ata_2(struct Nokta *nokta, int x, int y) {
    (*nokta).x = x;
    (*nokta).y = y;
}
double Nokta_uzaklik_2(struct Nokta *n1, struct Nokta *n2) {
    return sqrt(pow(n1->x - n2->x, 2) + pow(n1->y - n2->y, 2));
}
int main() {
    struct Nokta nokta1 = {1, 4};
    struct Nokta nokta2 = {1, 2};
    Nokta_deger_ata_2(&nokta1, 2, 1);
    printf("deger_ata_2 fonksiyonu sonrasinda\n");
    printf("nokta1 x,y : %d,%d\n", nokta1.x, nokta1.y);
    double uzaklik = Nokta_uzaklik_2(&nokta1, &nokta2);
    printf("nokta1 ve nokta2 arasindaki uzaklik: %f\n", uzaklik);
    return 0;
}
```

Struct Tipinde Dizi

```
#include <stdio.h>
#include <string.h>
struct Tarih {
    int gun;
    int ay;
    int yil;
};

struct Tarih Tarih_olustur(int gun, int ay, int yil) {
    struct Tarih t;
    if (gun > 31 || ay > 12) {
        t.gun = 0;
        t.gun = gun;
        t.ay = 0;
        t.ay = ay;
        t.yil = 0;
        t.yil = yil;
    }
    return t;
}

struct Ogrenci {
    char ad[21];
    char soyad[21];
    char no[11];
    char bolum[41];
    struct Tarih kayit_tarihi;
};

void Ogrenci_atama_yap(struct Ogrenci *ogr, const char adi[], const char soyadi[], const char numarası[], const char bolumu[], struct Tarih kayit_oldugu_tarih) {
    strcpy(ogr->ad, adi);
    strcpy(ogr->soyad, soyadi);
    strcpy(ogr->no, numarası);
    strcpy(ogr->bolum, bolumu);
    ogr->kayit_tarihi = kayit_oldugu_tarih;
}

int main() {
    struct Ogrenci ogr1;
    Ogrenci_atama_yap(&ogr1, "Abcd", "Soyad", "150202175", "Bilgisayar Muhendisligi", Tarih_olustur(01, 02, 2015));
    printf("adi: %s\n", ogr1.ad);
    printf("soyadi: %s\n", ogr1.soyad);
    printf("no: %s\n", ogr1.no);
    printf("bolum: %s\n", ogr1.bolum);
    printf("kayit yili: %d\n\n", ogr1.kayit_tarihi.yil);
    return 0;
}
```

Struct Tipinde Dizi

```
#include <stdio.h>
#include <string.h>
struct Tarih {
    int gun;
};

struct Tarih Tarih_olustur(int gun, int ay, int yil) {
    struct Tarih t;
    if (gun > 31 || ay > 12) {
        t.gun = 0;
    } else {
        t.gun = gun;
    }
    t.yil = 0;
    t.yil = yil;
}

struct Ogrenci {
    char ad[21];
    char soyad[21];
    char no[11];
    char bolum[41];
    struct Tarih kayit_tarihi;
};

void Ogrenci_atama_yap(struct Ogrenci *ogr, const char adi[], const char soyadi[], const char numarası[], const char bolumu[], struct Tarih kayit_oldugu_tarih) {
    strcpy(ogr->ad, adi);
    strcpy(ogr->soyad, soyadi);
    strcpy(ogr->no, numarası);
    strcpy(ogr->bolum, bolumu);
    ogr->kayit_tarihi = kayit_oldugu_tarih;
}

int main() {
    struct Ogrenci ogr_liste[20];
    Ogrenci_atama_yap(&ogr_liste[0], "A", "A", "150202177", "BilgisayarMuhendisligi", Tarih_olustur(1, 2, 2015));
    Ogrenci_atama_yap(&ogr_liste[1], "B", "B", "150202178", "BilgisayarMuhendisligi", Tarih_olustur(1, 2, 2015));
    Ogrenci_atama_yap(&ogr_liste[2], "C", "C", "150202179", "BilgisayarMuhendisligi", Tarih_olustur(2, 9, 2014));
    int kayitli_ogr_sayisi = 3;
    for (int i = 0; i < kayitli_ogr_sayisi; i++) {
        printf("adi soyad: %s %s\n", ogr_liste[i].ad, ogr_liste[i].soyad);
        printf("no: %s\n", ogr_liste[i].no);
        printf("bolum: %s\n", ogr_liste[i].bolum);
        printf("kayit tarihi: %02d.%02d.%d\n", ogr_liste[i].kayit_tarihi.gun, ogr_liste[i].kayit_tarihi.ay, ogr_liste[i].kayit_tarihi.yil);
    }
    return 0;
}
```

Sorular

