



YAPISAL PROGRAMLAMA 2024-2025 DÖNEM PROJESİ

Öğrenci Adı: MUHAMMED EMİR GÜL

Öğrenci Numarası: 23011051

Dersin Eğitmeni: HAFİZE İREM TÜRKMEN ÇİLİNGİR

Video Linki: <https://www.youtube.com/watch?v=W64UkKsrKaQ>

1- Proje Geliştirme Süreci:

Proje geliştirme sürecinde öncelikle structlar tanımlandı. Daha sonra tanımlanan structlara uygun fonksiyonlar yazıldı. Hazırlanan fonksiyonlar Linked Listte söylenen biçimde kaydedildi. Daha sonra işi basitleştirmek adına Fonksiyon Pointer uygulandı aynı parametre sayısı olanlara ve parametre düzeni aynı olanlara. Daha sonra gelen veriler CSV dosyasında kaydedildi. Program tekrar çalıştığında CSV dosyasından veriler okunup tekrardan programa geçirildi. Böylelikle tüm fonksiyonlar yapıldı.

2- Karşılaşılan Sorunlar:

Karşılaşılan sorunlardan bir tanesi ve en zorlayan dosyaya yazım ve dosyadan okuma işlemleri bunun için araştırma yapıldı ve dosya okuma o veriyi alma işlemleri edinilen bilgiler sonucunda yapıldı.

3- Ekran Çıktıları:

Hepsinin Detayları Videoadadır

ÖĞRENCİ EKLEME

```
void ogrenci_ekle(STUDENT **head) {
    STUDENT *s2 = malloc(sizeof(STUDENT));
    if (!s2) {
        printf("Bellek ayrılamadı!\n");
        return;
    }
    printf("Öğrencinin adını ve soyadını giriniz: ");
    scanf("%s %s", s2->isim, s2->soyisim);
    if (s2->isim[0] == '\0' || s2->soyisim[0] == '\0') {
        printf("Geçersiz isim veya soyisim!\n");
        free(s2);
        return;
    }
    s2->puan = 100;
    s2->next = NULL;
    s2->prev = NULL;
}
int max_id = 18011000;
STUDENT *tmp = *head;
while (tmp != NULL) {
    if (tmp->studentID > max_id) {
        max_id = tmp->studentID;
    }
    tmp = tmp->next;
}
s2->studentID = max_id + 1;
if (*head == NULL) {
    *head = s2;
} else {
    tmp = *head;
    STUDENT *prev = NULL;
    while (tmp != NULL && tmp->studentID < s2->studentID) {
        prev = tmp;
        tmp = tmp->next;
    }
    if (*head == NULL) {
        *head = s2;
    } else {
        s2->next = tmp;
        s2->prev = prev;
        prev->next = s2;
        if (tmp != NULL) {
            tmp->prev = s2;
        }
    }
}
ogrenci_dosyaya_yaz(*head);
printf("Öğrenci eklenildi: %s %s, ID: %d\n", s2->isim, s2->soyisim, s2->studentID);
```

Bu fonksiyon, bağlı listede yeni bir öğrenci düğümü oluşturup, kullanıcıdan öğrenci adı ve soyadı bilgilerini alarak doğrulama yapar. Öğrenciye otomatik olarak mevcut en yüksek ID'den bir sonraki ID atanır ve gerekli kontroller yapılır. Daha sonra listeye ID sırasına göre sıralı şekilde ekleme yapılır. Fonksiyon sonunda güncel öğrenci listesini dosyaya yazar ve eklenen öğrencinin bilgilerini ekrana bildirir.

The terminal window displays the following content:

- File contents ('Og'): A list of student records, each consisting of a student ID, first name, last name, and a score (100). The IDs range from 10202512 to 25012036.
- Program menu:
 - 9 - Kitap Sil
 - 10 - Kitap Guncelle
 - 11 - Kitap Bilgisi Goruntule
 - 12 - Raftaki Kitaplar Listele
 - 13 - Yazar Ekle
 - 14 - Yazar Sil
 - 15 - Yazar Guncelle
 - 16 - Yazar Listele
 - 17 - Kitap-Yazar Eslestir
 - 18 - Kitabın Yazarını Guncelle
 - 19 - Yazar Bilgisi Goruntule
 - 20 - Kitap Odunc/Iade Islemi
 - 21 - Zamanında Teslim Edilmeyen Kitaplar
 - 0 - Cikis
- User input:
 - Seçiminiz: 1
 - Öğrencinin adını ve soyadını giriniz: enes cetin
 - Öğrenci eklenildi: enes cetin, ID: 25012036
- Program footer:

----- KUTUPHANE OTOMASYON MENUSU -----
1 - Öğrenci Ekle
2 - Öğrenci Sil

ÖĞRENCİ SİLME

```
void ogrenci_sil(STUDENT **head) {
    int silinecekID;
    printf("Silinecek öğrencinin ID'sini giriniz: ");
    scanf("%d", &silinecekID);
    if (!validate_id(silinecekID)) {
        printf("Geçersiz öğrenci ID!\n");
        return;
    }

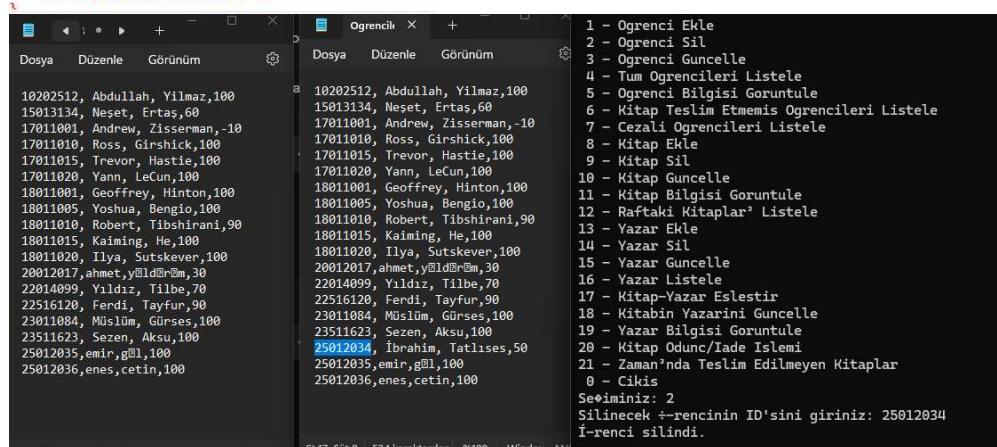
    STUDENT *tmp = *head;
    STUDENT *prev = NULL;
    int found = 0;

    while (tmp != NULL) {
        if (tmp->studentID == silinecekID) {
            found = 1;

            if (prev == NULL) {
                *head = tmp->next;
                if (*head != NULL) {
                    (*head)->prev = NULL;
                }
            } else {
                prev->next = tmp->next;
                if (tmp->next != NULL) {
                    tmp->next->prev = prev;
                }
            }
            free(tmp);
            tmp = NULL;
        } else {
            prev = tmp;
            tmp = tmp->next;
        }
    }

    if (!found) {
        printf("Öğrenci bulunamadı!\n");
        return;
    }

    ogrenci_dosyaya_yaz(*head);
    printf("Öğrenci silindi.\n");
}
```



```
Ogrenci: x + Dosya Düzenle Görünüm
Dosya Düzenle Görünüm
10202512, Abdullah, Yilmaz,100
15013134, Nese, Ertas,60
17011001, Andrew, Zisserman,-10
17011010, Ross, Girshick,100
17011015, Trevor, Hastie,100
17011020, Yann, LeCun,100
18011001, Geoffrey, Hinton,100
18011005, Yoshua, Bengio,100
18011010, Robert, Tibshirani,90
18011015, Kaiming, He,100
18011020, Ilya, Sutskever,100
20012017, ahmet,yildirim,30
22014099, Yildiz, Tilbe,70
22516120, Ferdi, Tayfur,90
23011084, Müsüm, Gürses,100
23511623, Sezen, Aksu,100
25012035, emir,göl,100
25012036, enes,cetin,100

1 - Ogrenci Ekle
2 - Ogrenci Sil
3 - Ogrenci Guncelle
4 - Tüm Öğrencileri Listele
5 - Öğrenci Bilgisi Görüntule
6 - Kitap Teslim Etmemis Öğrencileri Listele
7 - Cezalı Öğrencileri Listele
8 - Kitap Ekle
9 - Kitap Sil
10 - Kitap Guncelle
11 - Kitap Bilgisi Görüntule
12 - Raftaki Kitaplar Listele
13 - Yazar Ekle
14 - Yazar Sil
15 - Yazar Guncelle
16 - Yazar Listele
17 - Kitap-Yazar Eslestir
18 - Kitabın Yazarının Guncelle
19 - Yazar Bilgisi Görüntule
20 - Kitap Odunc/Iade Islemi
21 - Zamanında Teslim Edilmeyen Kitaplar
0 - Cikis
Seçiminiz: 2
Silinecek öğrencinin ID'sini giriniz: 25012034
Öğrenci silindi.
```

Bu fonksiyon, kullanıcidan silinmek istenen öğrencinin ID'sini alır ve geçerliliğini kontrol eder. Ardından bağlı listede bu ID'ye sahip öğrenciyi arar, bulduğunda listeden çıkarıp belleği serbest bırakır. Öğrenci bulunamazsa uyarı verir. Silme işlemi tamamlandıktan sonra güncel öğrenci listesini dosyaya kaydeder ve işlemin başarılı olduğunu ekrana bildirir.

ÖĞRENCİ GÜNCELLEME

The terminal window shows two parts of a program. On the left, a list of student records is displayed:

| ID | Ad | Soyad | Puan |
|----------|----------------|------------|------|
| 02020512 | Abdullah | Yilmaz | 100 |
| 15013134 | Neşet | Ertas | 60 |
| 17011001 | Andrew | Zisserman | -10 |
| 17011010 | Ross | Girshick | 100 |
| 17011015 | Trevor | Hastie | 100 |
| 17011020 | Yann | LeCun | 100 |
| 18011001 | Geoffrey | Hinton | 100 |
| 18011005 | Yoshua | Bengio | 100 |
| 18011010 | Robert | Tibshirani | 90 |
| 18011015 | Kaiming | He | 100 |
| 18011020 | Ilya | Sutskever | 100 |
| 20012017 | ahmet,yıldırım | 30 | |
| 22014099 | adanalı,ayhan | 85 | |
| 22511620 | Ferdi | Tayfur | 90 |
| 23011084 | Müslüm | Gürses | 100 |
| 23511623 | Sezen | Aksu | 100 |
| 25012035 | emir,gül | 100 | |
| 25012036 | enes,cetin | 100 | |

On the right, a menu for updating student information is shown:

- 14 - Yazar Sil
- 15 - Yazar Guncelle
- 16 - Yazar Listele
- 17 - Kitap-Yazar Eslestir
- 18 - Kitabin Yazarini Guncelle
- 19 - Yazar Bilgisi Goruntule
- 20 - Kitap Odunc/Iade Islemi
- 21 - Zaman'nda Teslim Edilmeyen Kitaplar
- 0 - Cikis

Seçiminiz: 3
Güncellenecek öğrenciin ID'sini girin: 22014099
Yeni ad ve soyad girin: adanalı
ayhan
Yeni puan girin: 85
İ-renci bilgisi güncellendi.

----- KUTUPHANE OTOMASYON MENUŞU -----
1 - Ogrenci Ekle
2 - Ogrenci Sil
3 - Ogrenci Guncelle

```
void ogrenci_guncelle(STUDENT *head) {
    int guncellenecekID;
    printf("Güncellenecek öğrencinin ID'sini girin: ");
    scanf("%d", &guncellenecekID);
    if (!validate_id(guncellenecekID)) {
        printf("Geçersiz öğrenci ID!\n");
        return;
    }

    STUDENT *tmp = head;
    int found = 0;

    while (tmp != NULL) {
        if (tmp->studentID == guncellenecekID) {
            printf("Yeni ad ve soyad girin: ");
            scanf("%s %s", tmp->isim, tmp->soyisim);
            if (!validate_string(tmp->isim) || !validate_string(tmp->soyisim)) {
                printf("Geçersiz yeni isim veya soyisim!\n");
                return;
            }
            printf("Yeni puan girin: ");
            scanf("%d", &tmp->puan);
            found = 1;
        }
        tmp = tmp->next;
    }

    if (!found) {
        printf("Öğrenci bulunamadı!\n");
        return;
    }

    ogrenci_dosyaya_yaz(head);
    printf("Öğrenci bilgisi güncellendi.\n");
}
```

Bu fonksiyon, kullanıcidan güncellenecek öğrencinin ID'sini alır ve listedeki öğrenciyi ID'ye göre arar. Öğrenci bulunduğuanda, kullanıcidan yeni ad, soyad ve puan bilgileri alınarak doğrulama yapılır ve ilgili öğrenci kaydı güncellenir. Eğer öğrenci bulunamazsa uygun uyarı verir. Güncelleme sonrası öğrenci listesi dosyaya kaydedilir ve işlem tamamlandığı ekrana bildirilir.

SEÇİLEN ÖĞRENCİ BİLGİSİ GÖRÜNTÜLEME

The terminal window shows a menu with options 11-21 and 0. The user selects option 11, which prompts for an ID and lists transactions for that student. The user then enters '22516120' and the system displays three transaction details. Below this, a function named 'ogrenci_bilgisi_goruntule' is shown in C code.

```
11 - Kitap Bilgisi Görüntüle
12 - Raftaki Kitaplar2 Listele
13 - Yazar Ekle
14 - Yazar Sil
15 - Yazar Guncelle
16 - Yazar Listele
17 - Kitap-Yazar Eslestir
18 - Kitabin Yazarini Guncelle
19 - Yazar Bilgisi Goruntule
20 - Kitap Odunc/Iade Islemi
21 - Zaman'nda Teslim Edilmeyen Kitaplar
0 - Cikis
Seçiminiz: 5
Bilgisi girdiğinizdeki öğrencinin ID'sini girin: 22516120
İ-renci: Ferdi Tayfur, ID: 22516120, Puan: 90
Kitap Hareketleri:
Etiket: 1478523691234_2, İşlem: İd'ne Alma, Tarih: 24.12.2022
Etiket: 6567502901954_1, İşlem: İd'ne Alma, Tarih: 25.01.2023
Etiket: 1478523691234_2, İşlem: Teslim, Tarih: 04.01.2023
----- KUTUPHANE OTOMASYON MENUŞU -----
1 - Ogrenci Ekle
2 - Ogrenci Sil
3 - Ogrenci Bul
4 - Ogrenci Listele
5 - Ogrenci Guncelle
6 - Ogrenci Sil
7 - Kitap Ekle
8 - Kitap Sil
9 - Kitap Guncelle
10 - Kitap Listele
11 - Kitap Bilgisi Görüntüle
12 - Raftaki Kitaplar2 Listele
13 - Yazar Ekle
14 - Yazar Sil
15 - Yazar Guncelle
16 - Yazar Listele
17 - Kitap-Yazar Eslestir
18 - Kitabin Yazarini Guncelle
19 - Yazar Bilgisi Goruntule
20 - Kitap Odunc/Iade Islemi
21 - Zaman'nda Teslim Edilmeyen Kitaplar
0 - Cikis
void ogrenci_bilgisi_goruntule(STUDENT *head, BORROW_LOG *logHead) {
    int ogrenciID;
    printf("Bilgisi görüntülenecek öğrencinin ID'sini girin: ");
    scanf("%d", &ogrenciID);
    if (!validate_id(ogrenciID)) {
        printf("Geçersiz öğrenci ID!\n");
        return;
    }

    STUDENT *tmp = head;
    int found = 0;

    while (tmp != NULL) {
        if (tmp->studentID == ogrenciID) {
            printf("Öğrenci: %s, ID: %d, Puan: %d\n", tmp->isim, tmp->soyisim, tmp->studentID, tmp->puan);
            printf("Kitap Hareketleri:\n");

            BORROW_LOG *log = logHead;
            while (log != NULL) {
                if (log->studentID == tmp->studentID) {
                    printf("Etiket: %s, İşlem: %s, Tarih: %s\n", log->etiket,
                           log->islemTuru == 0 ? "Ödünç Alma" : "Teslim", log->tarih);
                }
                log = log->next;
            }
            found = 1;
            tmp = NULL;
        } else {
            tmp = tmp->next;
        }
    }

    if (!found) {
        printf("Öğrenci bulunamadı!\n");
    }
}
```

Bu fonksiyon, kullanıcından bir öğrenci ID'si alarak liste içinde bu ID'ye sahip öğrenciyi arar. Öğrenci bulunduğuanda, öğrencinin adı, soyadı, ID'si ve puanı ekrana yazdırılır. Ayrıca, ödünç alma ve teslim işlemlerini içeren kayıtlar da öğrencinin ID'sine göre taranır ve ilgili kitap hareketleri işlem türü ve tarih bilgileriyle birlikte gösterilir. Eğer öğrenci bulunamazsa, kullanıcıya uygun bir uyarı mesajı iletilir. Fonksiyon, arama işlemi tamamlandıktan sonra gereksiz döngü devamını önlemek için akıllıca döngüyü sonlandırır.

KİTAP TESLİM ETMEMİŞ ÖĞRENCİLERİ GÖRÜNTÜLEME

```
void teslim_etmemis_ogrencileri_listele(STUDENT *head, BORROW_LOG *logHead) {
    printf("\n--- Teslim Etmemiş Öğrenciler ---\n");
    STUDENT *ogr = head;
    while (ogr != NULL) {
        int has_borrowed = 0;
        BORROW_LOG *log = logHead;
        while (log != NULL) {
            if (log->studentID == ogr->studentID && log->islemTuru == 0) {
                int returned = 0;
                BORROW_LOG *check = logHead;
                while (check != NULL && !returned) {
                    if (check->islemTuru == 1 && check->studentID == ogr->studentID &&
                        strcmp(check->etiket, log->etiket) == 0) {
                        returned = 1;
                    }
                    check = check->next;
                }
                if (!returned) {
                    has_borrowed = 1;
                }
            }
            log = log->next;
        }
        if (has_borrowed) {
            printf("%s %s, ID: %d\n", ogr->isim, ogr->soyisim, ogr->studentID);
        }
        ogr = ogr->next;
    }
}
```

Bu fonksiyon, kitap ödünç alma ve teslim işlemlerini içeren kayıtlar (BORROW_LOG) temelinde, teslim edilmemiş kitapları olan öğrencileri tespit eder ve listeler. Öğrenci listesinde dolaşırken her öğrenci için ödünç alınan kitap kayıtları taranır. Her ödünç alma kaydına karşılık, aynı kitabı teslim kaydı olup olmadığı kontrol edilir. Eğer teslim kaydı yoksa, öğrenci teslim etmemiş olarak işaretlenir. Fonksiyon, teslim etmemiş tüm öğrencilerin ad, soyad ve ID bilgilerini ekrana yazdırarak, kitapları zamanında iade etmeyenlerin takibini sağlar.

```
20 - Kitap Odunc/Iade Islemi
21 - Zaman'nda Teslim Edilmeyen Kitaplari
0 - Cikis
Seçiminiz: 6

--- Teslim Etmemis Öğrenciler ---
Abdullah Yilmaz, ID: 10202512
Ross Girshick, ID: 17011010
Trevor Hastie, ID: 17011015
Geoffrey Hinton, ID: 18011001
Robert Tibshirani, ID: 18011010
ahmet yıldırım, ID: 20012017
adanalı ayhan, ID: 22014099
Ferdi Tayfur, ID: 22516120
Müslüm Gürses, ID: 23011084
Sezen Aksu, ID: 23511623
```

CEZALI ÖĞRENCİLERİ GÖRÜNTÜLEME

```
void cezali_ogrencileri_listele(STUDENT *head) {
    printf("\n--- Cezali Öğrenciler ---\n");
    STUDENT *tmp = head;
    while (tmp != NULL) {
        if (tmp->puan < 0) {
            printf("%s %s, ID: %d, Puan: %d\n", tmp->isim, tmp->soyisim, tmp->studentID, tmp->puan);
        }
        tmp = tmp->next;
    }
}
```

Bu fonksiyon, bağlı listedeki tüm öğrencileri sırayla dolaşarak puanı sıfırın altında olan yani cezalı öğrencileri belirler. Cezalı öğrenci bulunan her düğüm için öğrencinin adı, soyadı, öğrenci numarası ve

mevcut puanı ekrana yazdırılır. Liste sonunda cezalı öğrenciler listesi tamamlanır. Böylece kullanıcı, sistemde ceza almış öğrencilerin detaylarını kolayca görebilir.

```
21 - Zannedi Noda Testim Editimeyen Kitaplar
0 - Cikis
Sehiriniz: 7

--- Cezalı İ-renciler ---
Andrew Zisserman, ID: 17011001, Puan: -10
```

TÜM ÖĞRENCİLERİ GÖRÜNTÜLEME

```
void ogrenci_listele_fp(void *head) {
    STUDENT *tmp = (STUDENT*)head;
    printf("\n--- ogrenci Listesi ---\n");
    if (tmp == NULL) {
        printf("ogrenci listesi bos!\n");
    } else {
        while (tmp != NULL) {
            printf("%d,%s,%s,%d\n", tmp->studentID, tmp->isim, tmp->soyisim, tmp->puan);
            tmp = tmp->next;
        }
    }
    printf("\n");}
```

```
--- ogrenci Listesi ---
10202512, Abdullah, Yilmaz,100
15013134, Nevet, Erta,60
17011001, Andrew, Zisserman,-10
17011010, Ross, Girshick,100
17011015, Trevor, Hastie,100
17011020, Yann, LeCun,90
18011001, Geoffrey, Hinton,100
18011005, Yoshua, Bengio,100
18011010, Robert, Tibshirani,90
18011015, Kaiming, He,100
18011020, Ilya, Sutskever,100
20012017,ahmet,yıldırım,30
22014099,adanalı,ayhan,85
22516120, Ferdi, Tayfur,90
23011084, M'sl'm, G'rses,100
23511623, Sezen, Aksu,100
25012035,emir,gül,100
25012036,enes,cetin,100
```

KİTAP ÖDÜNÇ ALMA

```

void kitap_odunc_islemleri(STUDENT *ogrenciHead, BOOK *kitapHead, BORROW_LOG **logHead, const char* logfile) {
    int studentID, islem;
    char etiket[30], tarih[12];
    printf("Öğrenci ID girin: ");
    scanf("%d", &studentID);
    if (!validate_id(studentID)) {
        printf("Geçersiz öğrenci ID!\n");
        return;
    }

    printf("Kitap etiketini girin (örn: 100000000001_1): ");
    scanf("%s", etiket);
    printf("İşlem türü (0=Ödünç alma, 1=teslim): ");
    scanf("%d", &islem);
    if (islem != 0 && islem != 1) {
        printf("Geçersiz işlem türü!\n");
        return;
    }

    printf("Tarih (gg.aa.yyyy): ");
    scanf("%s", tarih);

    STUDENT *ogr = ogrenciHead;
    int ogr_found = 0;
    while (ogr != NULL && !ogr_found) {
        if (ogr->studentID == studentID) {
            ogr_found = 1;
        }
        ogr = ogr->next;
    }

    if (!ogr_found) {
        printf("Öğrenci bulunamadı!\n");
        return;
    }

    if (islem == 0 && ogr->puan < 0) {
        printf("Öğrenci puanı negatif. Kitap verilemez!\n");
        return;
    }

    BORROW_LOG *yeni = malloc(sizeof(BORROW_LOG));
    if (!yeni) {
        printf("Bellek ayrılamadı!\n");
        return;
    }
    strcpy(yeni->etiket, etiket);
    yeni->studentID = studentID;
    yeni->islemTuru = islem;
    strcpy(yeni->tarih, tarih);
    yeni->next = *logHead;
    *logHead = yeni;

    FILE *fp = fopen(logFile, "a");
    if (!fp) {
        printf("Dosya açılamadı!\n");
        return;
    }
    fprintf(fp, "%s,%d,%d,%s\n", etiket, studentID, islem, tarih);
    fclose(fp);

    kitaplarla_dosyaya_yaz(kitapHead);
    printf("İşlem başarıyla kaydedildi.\n");
}

```

Bu fonksiyon, kütüphane sisteminde kitap ödünç alma veya iade işlemlerini yönetir. Öncelikle kullanıcıdan öğrenci ID'si, kitap etiketi, işlem türü (ödünç alma veya teslim) ve işlem tarihini alır ve doğrular. Öğrencinin sistemde kayıtlı olup olmadığını ve puan durumunu kontrol eder; negatif puanı olan öğrencilere ödünç kitap vermez. Ardından belirtlen kitap kopyasının mevcut durumunu kontrol ederek, ödünç verilebilir ya da teslim edilebilir olup olmadığını belirler. İşlem başarılıysa kitap kopyasının durumunu günceller. Teslim işlemi yapılrken, ödünç alma tarihinden itibaren 15 günü geçen gecikmelerde öğrenciye ceza puanı düşürür ve bu bilgiyi dosyaya kaydedeler. İşlem bilgileri hem hafızadaki log listesine hem de dosyaya yazılır. Son olarak, kitap listesini güncelliyip işlem sonucunu kullanıcıya bildirir.



KİTAP TESLİM ETME

```
KitapOdunc.cs * + - X
Dosya Düzenle Görünüm ☰
3219022972921_1,23511623,0,17.02.2023
1478523691234_2,17011015,0,12.02.2023
9780140449334_6,18011010,0,08.01.2023
1478523691234_2,22516120,1,04.01.2023
6567502901954_1,15013134,1,25.01.2023
9361229347876_1,23011084,0,21.12.2022
4054110263807_4,20012017,0,01.02.2023
9780140449335_3,17011015,0,10.01.2023
6567502901954_1,22516120,0,25.01.2023
9780140449335_2,23511623,0,20.01.2023
4001589630538_2,18011020,1,17.01.2023
1478523691234_2,22516120,0,24.12.2022
4054110263807_2,18011020,1,25.12.2022
7669545841623_4,10202512,0,13.12.2023
3219022972921_2,25012034,0,19.02.2023
4054110263807_5,18011010,1,20.02.2022
7669545841623_3,22014099,0,20.12.2022
1478523691234_3,18011001,0,18.02.2023
14 - Yazar Sil
15 - Yazar Guncelle
16 - Yazar Listele
17 - Kitap-Yazar Eslestir
18 - Kitabin Yazarini Guncelle
19 - Yazar Bilgisi Goruntule
20 - Kitap Odunc/Iade Islemi
21 - Zaman'nda Teslim Edilmeyen Kitaplar
0 - Cikis
Seçiminiz: 20
İ-renci ID girin: 17011015
Kitap etiketini girin (dn: 1000000000001_1): 1478523691234_2
!lem t'r (0=d'n alma, 1=teslim): 1
Tarih (gg.aa.yyyy): 27.05.2025
15 g'n' ge'eti, -10 puan cezası verildi.
!lem balar'yla kaydedildi.

----- KUTUPHANE OTOMASYON MENUSU -----
1 - Ogrenci Ekle
2 - Ogrenci Sil
3 - Ogrenci Guncelle
4 - Tum Ogrencileri Listele
5 - Ogrenci Bilgisi Goruntule
```

YAZAR İŞLEMLERİ

YAZAR EKLEME

```
Yazar *s1 = malloc(sizeof(YAZAR));
if (!s1) {
    printf("Bellek ayrılamadı!\n");
    return;
}

printf("Yazarın ismini ve soyismini giriniz: ");
scanf("%s %s", s1->isim, s1->soyisim);
if (!validate_string(s1->isim) || !validate_string(s1->soyisim)) {
    printf("Geçersiz isim veya soyisim!\n");
    free(s1);
    return;
}

s1->yazarID = (*count)++;
s1->next = NULL;

if (*head == NULL) {
    *head = s1;
} else {
    YAZAR *tmp = *head;
    YAZAR *prev = NULL;
    while (tmp != NULL && tmp->yazarID < s1->yazarID) {
        prev = tmp;
        tmp = tmp->next;
    }
    if (prev == NULL) {
        s1->next = *head;
        *head = s1;
    } else {
        prev->next = s1;
        s1->next = tmp;
    }
}

yazar_dosyaya_yaz(*head);
printf("Yazar eklendi: %s %s, ID: %d\n", s1->isim, s1->soyisim, s1->yazarID);
```

Bu fonksiyon, kullanıcidan

alınan isim ve soyisim bilgileri ile yeni bir yazar kaydı oluşturarak tek yönlü bağlı listeye ekler. Girişler geçerliyse yazara, global sayaç olan count değeri kullanılarak benzersiz bir yazarID atanır. Liste boşsa yazar doğrudan başa eklenir; değilse ID sırasına göre doğru konuma yerleştirilir. Böylece liste her zaman artan ID sırasını korur. Ekleneen yazar bilgileri ayrıca "Yazarlar.csv" dosyasına kaydedilir ve ekleme işlemi kullanıcıya bildirilir. Bu yapı sayesinde yazarlar bellekte ve dosyada düzenli bir şekilde saklanır.

YAZAR SILME

```

void yazar_sil(YAZAR **head) {
    int silinecekID;
    printf("Silinecek yazarın ID'sini giriniz: ");
    scanf("%d", &silinecekID);
    if (!validate_id(silinecekID)) {
        printf("Geçersiz yazar ID!\n");
        return;
    }

    YAZAR *tmp = *head;
    YAZAR *prev = NULL;
    int found = 0;

    while (tmp != NULL) {
        if (tmp->yazarID == silinecekID) {
            found = 1;
            break;
        }
        prev = tmp;
        tmp = tmp->next;
    }

    if (!found) {
        printf("Yazar bulunamadı!\n");
        return;
    }

    if (prev == NULL) {
        *head = tmp->next;
    } else {
        prev->next = tmp->next;
    }

    int i;
    int elemanSayisi;
    KitapYazar *liste = csv_oku("KitapYazar.csv", &elemanSayisi);
    if (liste) {
        for (i = 0; i < elemanSayisi; i++) {
            if (liste[i].yazarID == silinecekID) {
                liste[i].yazarID = -1;
            }
        }
        csv_yaz("KitapYazar.csv", liste, elemanSayisi);
        free(liste);
    }

    free(tmp);
    yazar_dosyaya_yaz(*head);
    printf("Yazar silindi.\n");
}

```

Bu fonksiyon, kullanıcidan silinecek yazarın ID'sini alarak doğrulama yaptıktan sonra, bağlı listede bu ID'ye sahip yazarı arar ve bulduğunda listeden çıkarır. Eğer silinen yazar listede başta ise doğrudan başlık güncellenir; değilse önceki düğüm ile bağlantılar yeniden düzenlenir. Ardından `KitapYazar.csv` dosyasındaki tüm kitap-yazar eşleştirmeleri taranarak, silinen yazara ait kayıtların yazar ID'si -1 ile işaretlenir. Bu işlem, silinen yazarın kitaplarla olan ilişkisinin kesilmesini sağlar. Son olarak hem güncel yazar listesi dosyaya yazılır hem de bellekten silme işlemi gerçekleştirilir.

YAZAR GÜNCELLEME

```
14 - Yazar Sil
15 - Yazar Guncelle
16 - Yazar Listele
17 - Kitap-Yazar Eslestir
18 - Kitabin Yazarini Guncelle
19 - Yazar Bilgisi Goruntule
20 - Kitap Odunc/Iade Islemi
21 - Zaman'nda Teslim Edilmeyen Kitaplar
0 - Cikis
Seçiminiz: 15
Güncellenecek yazarın ID'sini girin: 9
Yeni ismi ve soyismi girin: semih
bayraktar
Yazar bilgisi güncellendi.

----- KUTUPHANE OTOMASYON MENUSU -----
1 - Ogrenci Ekle
2 - Ogrenci Sil
3 - Ogrenci Guncelle
4 - Tum Ogrencileri Listele
5 - Ogrenci Bilgisi Goruntule
```

```
void yazar_guncelle(YAZAR *head) {
    int güncellenecekID;
    printf("Güncellenecek yazarın ID'sini girin: ");
    scanf("%d", &güncellenecekID);
    if (!validate_id(güncellenecekID)) {
        printf("Geçersiz yazar ID!\n");
        return;
    }

    YAZAR *tmp = head;
    int found = 0;

    while (tmp != NULL) {
        if (tmp->yazarID == güncellenecekID) {
            printf("Yeni ismi ve soyismi girin: ");
            scanf("%s %s", tmp->isim, tmp->soyisim);
            if (!validate_string(tmp->isim) || !validate_string(tmp->soyisim)) {
                printf("Geçersiz yeni isim veya soyisim!\n");
                return;
            }
            found = 1;
            break;
        }
        tmp = tmp->next;
    }

    if (!found) {
        printf("Yazar bulunamadı!\n");
        return;
    }

    yazar_dosyaya_yaz(head);
    printf("Yazar bilgisi güncellendi.\n");
}
```

Bu fonksiyon, bağlı listede yer alan yazarların kimlik bilgileri üzerinden bir güncelleme işlemi yapar. Kullanıcıdan güncellenecek yazarın ID'si alınarak geçerliliği kontrol edilir. Liste üzerinde ilgili ID'ye sahip yazar bulunana kadar ilerlenir. Eşleşen yazar bulunduğuunda kullanıcıdan yeni isim ve soyisim bilgileri alınır ve doğrulandıktan sonra mevcut verilerin üzerine yazılır. Güncelleme işlemi tamamlandığında, yazar listesinin tamamı dosyaya yeniden yazılarak kalıcılığı sağlanır. Eğer belirtilen ID'ye sahip yazar bulunamazsa kullanıcı bilgilendirilir.

TÜM YAZARLARI LİSTELEME

```
void yazar_listele_fp(void *head) {
    YAZAR *tmp = (YAZAR*)head;
    printf("\n--- Yazar Listesi ---\n");
    if (tmp == NULL) {
        printf("Yazar listesi bos!\n");
    } else {
        while (tmp != NULL) {
            printf("%d,%s,%s\n", tmp->yazarID, tmp->isim, tmp->soyisim);
            tmp = tmp->next;
        }
    }
    printf("\n");
```

```
16 - Yazar Listele
17 - Kitap-Yazar Eslestir
18 - Kitabin Yazarini Guncelle
19 - Yazar Bilgisi Goruntule
20 - Kitap Odunc/Iade Islemi
21 - Zaman'nda Teslim Edilmeyen Kitaplar
0 - Cikis
Seçiminiz: 16

--- Yazar Listesi ---
2, Stephen,Boyd
3, Batuhan,Tanriverdi
4, David,Johnson
5, Scott,Kirkpatrick
6, Lieven,Vandenbergh
7, Fabian,Pedregosa
8, Jorge,Nocedal
9,semih,bayraktar
11, Yaar,Kemal
12, O-uz,Atay
13, Reat Nuri,Gltekin
14, Ahmet Hamdi,Tanpnar
15, Stephen,Hawking
16, Carl,Sagan
```

KİTAP İŞLEMLERİ KİTAP EKLEME

```
void kitap_ekleme(BOOK **head) {
    getchar();
    BOOK *s3 = malloc(sizeof(BOOK));
    if (!s3) {
        printf("Bellek ayıramadı!\n");
        return;
    }

    printf("Kitabın adını giriniz: ");
    fgets(s3->kitapad, sizeof(s3->kitapad), stdin);
    s3->kitapad[strcspn(s3->kitapad, "\n")] = '\0';
    if (!validate_string(s3->kitapad)) {
        printf("Geçersiz kitap adı!\n");
        free(s3);
        return;
    }

    printf("Kitap adedini giriniz: ");
    scanf("%d", &s3->adet);
    if (s3->adet <= 0) {
        printf("Geçersiz adet!\n");
        free(s3);
        return;
    }

    s3->next = NULL;
    s3->copiesHead = NULL;

    long long max_isbn = 100000000000LL;
    BOOK *tmp = *head;
    while (tmp != NULL) {
        if (tmp->isbn > max_isbn) {
            max_isbn = tmp->isbn;
        }
        tmp = tmp->next;
    }
    s3->isbn = max_isbn + 1;
    int i;
    for (i = 1; i <= s3->adet; i++) {
        BOOK_COPY *newCopy = malloc(sizeof(BOOK_COPY));
        if (!newCopy) {
            printf("Bellek ayıramadı!\n");
        }
        newCopy->copy = s3->copiesHead;
        while (copy != NULL) {
            BOOK_COPY *temp = copy;
            copy = copy->next;
            free(temp);
        }
    }
}

void kitap_ekleme(BOOK **head) {
    if (*head == NULL) {
        *head = s3;
    } else {
        tmp = *head;
        BOOK *prev = NULL;
        while (tmp != NULL && (strcmp(tmp->kitapad, s3->kitapad) < 0 || (strcmp(tmp->kitapad, s3->kitapad) == 0 && tmp->isbn < s3->isbn))) {
            prev = tmp;
            tmp = tmp->next;
        }
        if (prev == NULL) {
            s3->next = *head;
            *head = s3;
        } else {
            prev->next = s3;
            s3->next = tmp;
        }
    }
    Kitaplar_dosyaya_yaz(*head);
    printf("Kitap eklendi: %s, ISBN: %lld\n", s3->kitapad, s3->isbn);
}

void kitap_cikar(BOOK **head) {
```

`kitap_ekleme` fonksiyonu, kullanıcıdan kitap adı ve adet bilgisi alarak yeni bir kitap nesnesi oluşturur ve bu kitabı bağlı listeye alfabetik sıraya göre ekler. Her kitap için benzersiz bir ISBN numarası üretilir (mevcut en büyük ISBN'den bir fazlası alınarak). Kitap adedine göre her bir kopya

İçin etiket bilgisi (ISBN_kopyaNo) oluşturulur ve kitap yapısının içinde bağlı liste şeklinde saklanır. Bellek yönetimi dikkatli yapılır; bellek ayrılamayan durumlarda daha önce ayrılmış kopyalar serbest bırakılır. Fonksiyon sonunda kitap listesi dosyaya kaydedilir ve eklenen kitabı bilgi ekrana yazdırılır.

```
Huzur,9702569648130,3
Algoritma Analizi,9780140449334,6
Yapı Verileri,9780140449335,3
sol ayağım,9780140449336,3
matematik ,9780140449337,4
matematik ,9780140449338,2
sosyal bilgiler,9780140449339,2
matematik,9780140449340,4
hayat bilgisi , 9780140449341,3
```

```
19 - Rafta Bulgisi Sıfırla
20 - Kitap Odunc/Iade Islemi
21 - Zaman'nda Teslim Edilmeyen Kitaplar
0 - Cikis
Seçiminiz: 8
Kitab'ın adı giriniz: hayat bilgisi
Kitap adedini giriniz: 3
Kitap eklendi: hayat bilgisi, ISBN: 9780140449341
----- KUTUPHANE OTOMASYON MENU'SU -----
```

KİTAP SILME

```
Fizik 1,1234567891012,4
Fizik 2,1234567891013,3
Finallerin Gazabı,1478523691234,3
Mesaj,2598714271168,2
C Programlama,3214569871012,4
Her Şeyin Teorisi,3219022972921,3
Yaprak Dökümü,4001589630538,7
hayat bilgisi,9780140449341,3
Çalışkuşu,4054110263807,6
Tutunamayanlar,5134504975481,3
Zamanın Kısa Tarihi,6567502901954,3
İnce Memed,7669545841623,4
Tehlikeli Oyunlar,8098255466027,2
Saatleri Ayarlama Enstitüsü,8134960522237,4
Soluk Mavi Nokta,9361229347876,3
Huzur,9702569648130,3
Algoritma Analizi,9780140449334,6
Yapı Verileri,9780140449335,3
matematik ,9780140449337,4
matematik ,9780140449338,2
sosyal bilgiler,9780140449339,2
matematik,9780140449340,4
```

```
12 - Raftaki Kitaplar' Listele
13 - Yazar Ekle
14 - Yazar Sil
15 - Yazar Guncelle
16 - Yazar Listele
17 - Kitap-Yazar Eslestir
18 - Kitabin Yazarini Guncelle
19 - Yazar Bilgisi Goruntule
20 - Kitap Odunc/Iade Islemi
21 - Zaman'nda Teslim Edilmeyen Kitaplar
0 - Cikis
Seçiminiz: 9
Silinecek kitab'ın adı giriniz: sol ayağım
Kitap silindi.
----- KUTUPHANE OTOMASYON MENU'SU -----
```

1 - Öğrenci Ekle

2 - Öğrenci Sil

3 - Öğrenci Guncelle

4 - Tüm Öğrencileri Listele

5 - Öğrenci Bilgisi Görüntüle

```
void Kitap_sil(BOOK **head)
{
    getch();
    char isim[50];
    printf("Silinecek kitabın adını giriniz: ");
    gets(isim, sizeof(isim));
    if (strcspn(isim, "\n") != '\0')
        if (!validate_string(isim))
            printf("Geçersiz kitap adı!\n");
        return;
}
```

```
BOOK *tmp = *head;
BOOK *prev = NULL;
int found = 0;

while (tmp != NULL && !found) {
    if (strcmp(tmp->kitapAdi, isim) == 0) {
        found = 1;
    } else {
        prev = tmp;
        tmp = tmp->next;
    }
}
```

```
if (!found) {
    printf("Kitap bulunamadı!\n");
    return;
}

if (prev == NULL) {
    *head = tmp->next;
} else {
    prev->next = tmp->next;
}
```

```
BOOK_COPY *copy = tmp->copiesHead;
while (copy != NULL) {
    BOOK_COPY *temp = copy;
    copy = copy->next;
    free(temp);
}
int i;

int elemanSayisi;
KitapYazar *liste = csv_oku("KitapYazar.csv", &elemanSayisi);
if (liste) {
    for (i = 0; i < elemanSayisi; i++) {
        if (liste[i].isbn == tmp->isbn) {
            liste[i].yazarID = -1;
        }
    }
    csv_yaz("KitapYazar.csv", liste, elemanSayisi);
    free(liste);
}

free(tmp);
kitapları_dosyaya_yaz(*head);
printf("Kitap silindi.\n");
}
```

`kitap_sil` fonksiyonu, kullanıcıdan silinmek istenen kitabı adını alarak, bağlı listedeki kitaplar arasında bu adı taşıyan kitabı arar. Kitap bulunursa, kitabı listeden çıkarılır, bellekte bu kitabı ait yapı ve içindeki tüm kopyalar (BOOK_COPY) serbest bırakılır. Ayrıca `KitapYazar.csv` dosyasındaki ilgili eşleşme yazar ID'si -1 yapılarak pasif hale getirilir. Son olarak kitabı listesi güncellenerek dosyaya yazılır ve kullanıcıya başarı mesajı verilir.

KİTAP GÜNCELLEME

```

5 - Ogrenci Bilgisi Goruntule
6 - Kitap Teslim Etmemis Ogrencileri Listele
7 - Cezali Ogrencileri Listele
8 - Kitap Ekle
9 - Kitap Sil
10 - Kitap Guncelle
11 - Kitap Bilgisi Goruntule
12 - Raftaki Kitaplar Listele
13 - Yazar Ekle
14 - Yazar Sil
15 - Yazar Guncelle
16 - Yazar Listele
17 - Kitap-Yazar Eslestir
18 - Kitabin Yazarini Guncelle
19 - Yazar Bilgisi Goruntule
20 - Kitap Odunc/Iade Islemi
21 - Zamaninda Teslim Edilmeyen Kitaplar
0 - Cikis
Se•iminiz: 10
Güncellenecek kitabın adı: huzursuz
Yeni adı girin: huzursuz
Yeni adet girin: 3
Kitap güncellendi.

----- KUTUPHANE OTOMASYON MENUSU -----
void kitap_guncelle(BOOK *head) {
    getchar();
    char aranan[50];
    printf("Güncellenecek kitabın adını giriniz: ");
    fgets(aranan, sizeof(aranan), stdin);
    aranan[strcspn(aranan, "\n")] = '\0';
    if (!validate_string(aranan)) {
        printf("Geçersiz kitabın adı!\n");
        return;
    }

    BOOK *tmp = head;
    int found = 0;

    while (tmp != NULL && !found) {
        if (strcmp(tmp->kitapad, aranan) == 0) {
            printf("Veni adı girin: ");
            fgets(tmp->kitapad, sizeof(tmp->kitapad), stdin);
            tmp->kitapad[strcspn(tmp->kitapad, "\n")] = '\0';
            if (!validate_string(tmp->kitapad)) {
                printf("Geçersiz yeni kitabın adı!\n");
                return;
            }
            printf("Veni adet girin: ");
            scanf("%d", &tmp->adet);
            if (tmp->adet <= 0) {
                printf("Geçersiz adet!\n");
                return;
            }
        }
    }

    BOOK_COPY *copy = tmp->copiesHead;
    while (copy != NULL) {
        BOOK_COPY *temp = copy;
        copy = copy->next;
        free(temp);
    }
    tmp->copiesHead = NULL;
    int i;
    for (i = 1; i <= tmp->adet; i++) {
        BOOK_COPY *newCopy = malloc(sizeof(BOOK_COPY));
        if (!newCopy) {
            printf("Bellek ayrılamadı!\n");
            return;
        }
        sprintf(newCopy->etiket, "%Id_%d", tmp->isbn, i);
        newCopy->odunc_durumu = 0;
        newCopy->next = NULL;

        if (tmp->copiesHead == NULL) {
            tmp->copiesHead = newCopy;
        } else {
            BOOK_COPY *copyTmp = tmp->copiesHead;
            while (copyTmp->next != NULL) {
                copyTmp = copyTmp->next;
            }
            copyTmp->next = newCopy;
        }
    }
    found = 1;
}

if (!found) {
    printf("Kitap Bulunamadi!");
}

```

`kitap_guncelle` fonksiyonu, kullanıcıdan güncellenecek kitabı adını alarak kitabı listesini tarar ve eşleşen kitabı bulur. Bulunan kitabı adı ve adet bilgisi kullanıcıdan alınarak güncellenir. Adet sayısının değişmesi durumunda, mevcut tüm kitabı kopyaları bellekte serbest bırakılır ve yeni adet sayısına göre

yeni kopya nesneleri oluşturularak kitapla ilişkilendirilir. Her kopyaya özgün bir etiket atanır. Güncellemeye işlemi tamamlandıktan sonra değişiklikler dosyaya yazılır. Fonksiyon, geçersiz girişler ve bellek ayırma hatalarına karşı kontroller içerir.

KİTAP BİLGİSİ GÖRÜNTÜLEME

```
0 - ÇIKIS
Seçiminiz: 11
Kitap adınızı giriniz: huzursuz
Kitap Adı: huzursuz
ISBN: 9702569648130
Adet: 3
Kopya Etiketi: 9702569648130_1 - Durum: Rafta
Kopya Etiketi: 9702569648130_2 - Durum: Rafta
Kopya Etiketi: 9702569648130_3 - Durum: Rafta
```

```
void kitap_bilgisi_goruntule(BOOK *head) {
    getchar();
    char aranan[50];
    printf("Kitap adınızı giriniz: ");
    fgets(aranan, sizeof(aranan), stdin);
    aranan[strcspn(aranan, "\n")] = '\0';
    if (!validate_string(aranan)) {
        printf("Geçersiz kitap adı!\n");
        return;
    }

    BOOK *tmp = head;
    int found = 0;

    while (tmp != NULL && !found) {
        if (strcmp(tmp->kitapAd, aranan) == 0) {
            printf("Kitap Adı: %s\n", tmp->kitapAd);
            printf("ISBN: %lld\n", tmp->isbn);
            printf("Adet: %d\n", tmp->adet);
            BOOK_COPY *copy = tmp->copiesHead;
            while (copy != NULL) {
                printf("Kopya Etiketi: %s - Durum: %s\n",
                       copy->etiket, copy->odunc_durumu == 0 ? "Rafta" : "Ödünçte");
                copy = copy->next;
            }
            found = 1;
        }
        tmp = tmp->next;
    }

    if (!found) {
        printf("Kitap bulunamadı!\n");
    }
}
```

kitap_bilgisi_goruntule fonksiyonu, kullanıcıdan bir kitap adı alarak bağlı listedeki kitaplar arasında arama yapar. Girilen adla eşleşen kitap bulunduğuanda, kitabı adı, ISBN numarası, adedi ve tüm kopyalarının etiketi ile ödünç durumu (rafta mı, ödünçte mi) ekrana yazdırılır. Kitap bulunamazsa kullanıcıya "Kitap bulunamadı!" mesajı verilir. Bu fonksiyon, kitap verilerinin görsel olarak kullanıcıya sunulmasını sağlar.

RAFTAKİ KİTAPLARI LİSTELEME

```
Seçiminiz: 12
---- Raftaki Kitaplar ---
Fizik 1 - 1234567891012_1
Fizik 1 - 1234567891012_2
Fizik 1 - 1234567891012_3
Fizik 1 - 1234567891012_4
Fizik 2 - 1234567891013_1
Fizik 2 - 1234567891013_2
Fizik 2 - 1234567891013_3
Finallerin Gazabı - 1478523691234_1
Mesaj - 2598714271168_1
Mesaj - 2598714271168_2
C Programlama - 3214569871012_1
C Programlama - 3214569871012_2
C Programlama - 3214569871012_3
C Programlama - 3214569871012_4
Her İeyin Teorisi - 3219022972921_3
Yaprak Döküm - 4001589630538_1
Yaprak Döküm - 4001589630538_3
Yaprak Döküm - 4001589630538_4
Yaprak Döküm - 4001589630538_5
Yaprak Döküm - 4001589630538_6
Yaprak Döküm - 4001589630538_7
hayat bilgisi - 9780140449341_1
hayat bilgisi - 9780140449341_2
hayat bilgisi - 9780140449341_3
Aalıkuşu - 4054110263807_1
Aalıkuşu - 4054110263807_2
```

```
void raftaki_kitaplari_listele(BOOK *head) {
    printf("\n--- Raftaki Kitaplar ---\n");
    BOOK *tmp = head;
    while (tmp != NULL) {
        BOOK_COPY *copy = tmp->copiesHead;
        while (copy != NULL) {
            if (copy->odunc_durumu == 0) {
                printf("%s - %s\n", tmp->kitapAd, copy->etiket);
            }
            copy = copy->next;
        }
        tmp = tmp->next;
    }
}
```

Raftaki kitaplari listele fonksiyonu, kütüphane sisteminde rafta bulunan kitaplari listelemek amacıyla kullanılır. Bu fonksiyon, tüm kitaplari ve her bir kitaba ait kopyaları dolaşarak, henüz ödünç verilmemiş (yani odunc_durumu == 0 olan) kitap kopyalarını ekrana yazdırır. Ekranda, kitap adı ile birlikte o kopyaya ait etiket bilgisi gösterilir. Böylece kullanıcılar rafta bulunan kitapların hangi kopyalarının erişilebilir olduğunu kolayca görebilir.

TESLİM EDİLMEYEN KİTAPLARI LİSTELEME

```
--- Zamanında Teslim Edilmeyen Kitaplar ---
Etiket: 1478523691234_2, İ-renci ID: 17011015, İd'num: 12.02.2023, Teslim: 27.05.2025
Etiket: 9780140449334_5, İ-renci ID: 23511623, İd'num: 25.12.2022, Teslim: 25.01.2023
```

```
void zamaninda_teslim_edilmeyenleri_listele(BORROW_LOG *logHead) {
    printf("\n--- Zamanında Teslim Edilmeyen Kitaplar ---\n");
    BORROW_LOG *log = logHead;
    while (log != NULL) {
        if (log->islemTuru == 1) {
            BORROW_LOG *borrow = logHead;
            while (borrow != NULL) {
                if (borrow->islemTuru == 0 && strcmp(borrow->etiket, log->etiket) == 0 &&
                    borrow->studentID == log->studentID) {
                    int days = compare_dates(borrow->tarih, log->tarih);
                    if (days > 15) {
                        printf("Etiket: %s, Öğrenci ID: %d, Ödünç: %s, Teslim: %s\n",
                            log->etiket, log->studentID, borrow->tarih, log->tarih);
                    }
                }
                borrow = borrow->next;
            }
            log = log->next;
        }
    }
}
```

zamaninda_teslim_edilmeyenleri_listele fonksiyonu, kitap ödünç ve iade kayıtlarını inceleyerek, belirli bir kitabı 15 günden daha uzun süreyle teslim etmeyen öğrencileri tespit eder. Fonksiyon, ilk olarak iade işlemlerini (islemTuru == 1) bulur, ardından aynı etiket ve öğrenci ID'sine sahip karşılık gelen ödünç alma kaydını arar. Eğer ödünç alma ile iade arasında geçen süre 15 günü aşıysa, bu gecikmeli işlem konsola yazdırılır. Böylece sistem, geç teslim edilen kitapların raporlanması sağlar.

KİTAP-YAZAR EŞLEŞTİRME

```
18
1301     yeni->tarih = tarih;
1302     strcpy(yeni->tarih, tarih);
1303     yeni->next = *logHead;
1304     *logHead = yeni;
1305
1306     FILE *fp = fopen(logFile, "a");
1307     if (!fp) {
1308         printf("Dosya açılamadı!\n");
1309         return;
1310     }
1311     for (int i = 0; i < 10; i++) {
1312         fprintf(fp, "%s,%d,%d,%s\n", etiket, std::to_string(i).c_str(), std::to_string(i).c_str(), std::string("etiket").c_str());
1313     }
1314 }
```

```
Seçiminiz: 17
Eşleştirilecek kitabın ISBN'sini girin: 9702569648130
Yazar ID'sini girin: 9
Eşleştirme tamamlandı.

----- KUTUPHANE OTOMASYON MENUSU -----
1 - Öğrenci Ekle
2 - Öğrenci Sil
3 - Öğrenci Güncelle
4 - Tüm Öğrencileri Listele
5 - Öğrenci Bilgisi Görüntüle
6 - Kitap Teslim Etmeyen Öğrencileri Listele
7 - CezaLİ Öğrencileri Listele
8 - Kitap Ekle
9 - Kitap Sil
```

```
void Kitap_Yazar_Eslestir(BOOK *bookHead, YAZAR *yazarHead) {
    long long isbn;
    int yazarID;
    printf("Eşleştirilecek kitabın ISBN'sini girin: ");
    scanf("%ld", &isbn);
    if (!Validate_Isbn(isbn)) {
        printf("Geçersiz ISBN!\n");
        return;
    }

    printf("Yazar ID'sini girin: ");
    scanf("%d", &yazarID);
    if (!Validate_Id(yazarID)) {
        printf("Geçersiz yazar ID!\n");
        return;
    }

    BOOK *book = bookHead;
    int book_found = 0;
    while (book != NULL && !book_found) {
        if (book->isbn == isbn) {
            book_found = 1;
        }
        book = book->next;
    }
    if (!book_found) {
        printf("Kitap bulunamadı!\n");
        return;
    }

    YAZAR *yazar = yazarHead;
    int yazar_found = 0;
    while (yazar != NULL && !yazar_found) {
        if (yazar->yazarID == yazarID) {
            yazar_found = 1;
        }
        yazar = yazar->next;
    }
    if (!yazar_found) {
        printf("Yazar bulunamadı!\n");
        return;
    }

    FILE *fp = fopen("KitapYazar.csv", "a");
    if (!fp) {
        printf("Dosya açılamadı!\n");
        return;
    }
    fprintf(fp, "%ld,%d\n", isbn, yazarID);
    fclose(fp);
    printf("Eşleştirme tamamlandı.\n");
}
```

kitap_yazar_eslestir fonksiyonu, bir kitap ile bir yazar arasında eşleştirme yapılmasını sağlar. Kullanıcıdan kitap ISBN numarası ve yazar ID'si alındıktan sonra, bu bilgiler doğrulanarak sistemde kayıtlı kitaplar ve yazarlar arasında arama yapılır. Eğer her iki kayıt da sisteme bulunursa, eşleştirme bilgisi "KitapYazar.csv" adlı dosyaya ISBN,YazarID formatında eklenir. Böylece bir kitabı hangi yazar tarafından yazıldığı sistemde kalıcı olarak kaydedilmiş olur. Fonksiyon, geçersiz giriş veya bulunamayan kayıtlar için kullanıcıyı bilgilendirir.

KİTABIN YAZARINI GÜNCELLEME

```
void kitabin_yazarini_guncelle(KitapYazar *liste, int eleman_sayisi) {
    long long isbn;
    int yeni_id;
    printf("ISBN girin: ");
    scanf("%lld", &isbn);
    if (!validate_isbn(isbn)) {
        printf("Geçersiz ISBN!\n");
        return;
    }

    printf("Yeni yazar ID girin: ");
    scanf("%d", &yeni_id);
    if (!validate_id(yeni_id)) {
        printf("Geçersiz yazar ID!\n");
        return;
    }

    int i;
    int found = 0;
    for (i = 0; i < eleman_sayisi; i++) {
        if (liste[i].isbn == isbn && liste[i].yazarID != -1) {
            liste[i].yazarID = yeni_id;
            found = 1;
        }
    }

    if (!found) {
        printf("Eşleştirme bulunamadı!\n");
        return;
    }

    csv_yaz("KitapYazar.csv", liste, eleman_sayisi);
    printf("Yazar güncellendi.\n");
}
```

```
14 - Yazar Sil
15 - Yazar Guncelle
16 - Yazar Listele
17 - Kitap-Yazar Eslestir
18 - Kitabın Yazarını Guncelle
19 - Yazar Bilgisi Goruntule
20 - Kitap Odunc/Iade Islemi
21 - Zaman'nda Teslim Edilmeyen Kitaplar
0 - Cikis
Seçiminiz: 18
ISBN girin: 4001589630538
Yeni yazar ID girin: 8
Yazar güncellendi.

----- KUTUPHANE OTOMASYON MENU'SU -----
1 - Ogrenci Ekle
2 - Ogrenci Sil
3 - Ogrenci Guncelle
4 - Tum Ogrencileri Listele
5 - Ogrenci Bilgisi Goruntule
6 - Kitap Teslim Etmemis Ogrencileri List
7 - Cezali Ogrencileri Listele
8 - Kitap Ekle
9 - Kitap Sil
10 - Kitap Guncelle
```

kitabin_yazarini_guncelle fonksiyonu, kullanıcıdan alınan ISBN numarasına sahip kitabın yazar bilgisini güncellemek amacıyla kullanılır. Fonksiyon, önce girilen ISBN ve yeni yazar ID'sinin geçerliliğini kontrol eder. Ardından, KitapYazar adlı ilişki listesini tarayarak bu ISBN'e ait kaydı bulur ve yazarID alanını yeni ID ile günceller. Eğer böyle bir eşleştirme bulunamazsa kullanıcıya bilgi verilir. Eşleştirme başarıyla güncellendikten sonra değişiklikler KitapYazar.csv dosyasına yazılıarak kalıcı hale getirilir. Bu işlem, kitap ile yazar arasındaki ilişkiyi düzenlemeye yarar.

PROJENİN KAZANDIRDIKLARI

Projeyi yaparken kazandığım en önemli kazanımlardan biri programlama sürecinde elde ettiğim akıcılık oldu. Öğrenci kayıtlarını yazıp, ardından dosya okuma ve yazma işlemlerini öğrenmemle birlikte, bu becerilerimi yazar, kitap ve yazar-kitap eşleşmelerinde uygularken işlerin hızlandığını fark ettim. Dosya işlemleri konusunda sağlam bir temel edindim ve veri yapıları arasında ilişki kurmayı deneyimledim. Ayrıca, verilerin dosya aracılığıyla saklanması konusunu kavradım. Proje sürecinde fonksiyon pointer kullanımı ve bunun amaçları hakkında da değerli bilgiler edindim. Tüm bu deneyimler programlama yetkinliğini artırdı ve daha verimli kod yazmamı sağladı.