

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<locale.h>
#include<string.h>
#include<time.h>

void dosya_yaz(struct seferler *);
void dosya_guncelle(char []);
void sefer_no_guncelle();
void sefer_ekle();
void basla();
void bilet_islem(int);
void bugun_tarih();
void dosya_oku(char []);
void gelir_hesapla();
void kaptan_degistir();
int karsilastir(char [],char []);
void sefer_listele();
void koltuk_listele(struct koltuk_listeleri *);
void sefer_sayisi();
void sefer_sil();
void ana_menu();
void sefer_menu();
void bilet_menu();
void bosluk();
void gecmis_oku();

/////////////////////////////////
struct koltuk_listeleri
{
    int koltuk_no;
    char yolcu_adi[30];
    float fiyat;
    char cinsiyet;
    int durum;
    struct koltuk_listeleri *onceki,*sonraki;
};

/////////////////////////////////
struct seferler
{
    char kaptan[30];
    char plaka[8];
    int sefer_no;
    char guzergah[25];
    char tarih[12];
    float bilet_fiyati;
    char saat[6];
    int kapasite;
    struct koltuk_listeleri *otobus;
    struct seferler *sonraki,*onceki;
};

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

struct seferler *son,*ilk;
int sefer_numarasi=1;
```

```
char uzanti[5]=".txt";
char bugun[12];
char guzergahlar[][20]={"kocaeli-istanbul", "kocaeli-ankara", "kocaeli-izmir",
"ankara-kocaeli", "istanbul-kocaeli", "izmir-kocaeli"};
FILE *log;

//////////////////////////////////////mainnnnnnnnnnnnnn
nnnnnn
main()
{
    setlocale(LC_ALL,"Turkish");

    bugun_tarih();
    basla();
    char secenek;

nokta0:
    system("cls");
    ana_menu();
    secenek=getchar();
    if(secenek=='1')
    {
        goto nokta1;
    }
    else if(secenek=='2')
    {
        goto nokta2;
    }
    else if(secenek=='K' || secenek=='k')
    {
        goto noktason;
    }
    else
    goto nokta0;

noktal1:
    system("cls");
    sefer_menu();
    secenek=getchar();
    if(secenek=='A' || secenek=='a')
    goto nokta0;
    else if(secenek=='1')
    {
        system("cls");
        sefer_ekle();
        printf("Ana Menü İçin ( A )          Üst Menü İçin( G )\n");
        Kapatmak İçin ( K )\n");
        nokta1x1:
        fflush(stdin);
        secenek=getchar();

        if(secenek=='a' || secenek=='A')
        {
            goto nokta0;
        }

        else if(secenek=='G' || secenek=='g')
        {
            goto nokta1;
        }
        else if(secenek=='K' || secenek=='k')
```

```
        {
            goto noktason;
        }
        else
        {
            bosluk();
            printf("Yanlış Seçim Tekrar Deneyin\n");
            goto nokta1x1;
        }

    }

    else if(secenek=='2')
    {
        system("cls");
        sefer_listele();
        printf("\n\n\n");
        sefer_sil();

        printf("Ana Menü İçin ( A )      Üst Menü İçin( G )
Kapatmak İçin ( K )\n");
        nokta1x2:

        fflush(stdin);
        secenek=getchar();

        if(secenek=='a' || secenek=='A')
        {
            goto nokta0;
        }

        else if(secenek=='G' || secenek=='g')
        {
            goto nokta1;
        }
        else if(secenek=='K' || secenek=='k')
        {
            goto noktason;
        }
        else
        {
            bosluk();
            printf("Yanlış Seçim Tekrar Deneyin\n");
            goto nokta1x2;
        }
    }

    else if(secenek=='3')
    {
        system("cls");
        sefer_listele();

        printf("\n\n\nAna Menü İçin ( A )      Üst Menü İçin( G
)      Kapatmak İçin ( K )\n");
        nokta1x3:

        fflush(stdin);
        secenek=getchar();

        if(secenek=='a' || secenek=='A')
        {
```

```
        goto nokta0;
    }

    else if(secenek=='G' || secenek=='g')
    {
        goto nokta1;
    }
    else if(secenek=='K' || secenek=='k')
    {
        goto noktason;
    }
    else
    {
        bosluk();
        printf("Yanlış Seçim Tekrar Deneyin\n");
        goto nokta1x3;
    }
}

else if(secenek=='4')
{
    system("cls");
    gecmis_oku();

    printf("\n\n\nAna Menü İçin ( A )      Üst Menü İçin( G
)      Kapatmak İçin ( K )\n");
    nokta1x4:

    fflush(stdin);
    secenek=getchar();

    if(secenek=='a' || secenek=='A')
    {
        goto nokta0;
    }

    else if(secenek=='G' || secenek=='g')
    {
        goto nokta1;
    }
    else if(secenek=='K' || secenek=='k')
    {
        goto noktason;
    }
    else
    {
        bosluk();
        printf("Yanlış Seçim Tekrar Deneyin\n");
        goto nokta1x4;
    }
}

else if(secenek=='5')
{
    system("cls");
    sefer_listele();
    printf("\n\n\n");
    gelir_hesapla();

    printf("\n\n\nAna Menü İçin ( A )      Üst Menü İçin( G
)      Kapatmak İçin ( K )\n");
```

```
nokta1x5:

fflush(stdin);
secenek=getchar();

if(secenek=='a' || secenek=='A')
{
goto nokta0;
}

else if(secenek=='G' || secenek=='g')
{
goto nokta1;
}
else if(secenek=='K' || secenek=='k')
{
goto noktason;
}
else
{
bosluk();
printf("Yanlış Seçim Tekrar Deneyin\n");
goto nokta1x5;
}

}

else if(secenek=='6')
{
system("cls");
sefer_sayisi();

printf("\n\nAna Menü İçin ( A )      Üst Menü İçin( G
)      Kapatmak İçin ( K )\n");
nokta1x6:

fflush(stdin);
secenek=getchar();

if(secenek=='a' || secenek=='A')
{
goto nokta0;
}

else if(secenek=='G' || secenek=='g')
{
goto nokta1;
}
else if(secenek=='K' || secenek=='k')
{
goto noktason;
}
else
{
bosluk();
printf("Yanlış Seçim Tekrar Deneyin\n");
goto nokta1x6;
}

}
```

```
else if(secenek=='7')
{
    system("cls");
    sefer_listele();
    printf("\n\n\n");
    kaptan_degistir();

    printf("\n\n\nAna Menü İçin ( A )      Üst Menü İçin( G
)      Kapatmak İçin ( K )\n");
    nokta1x7:

    fflush(stdin);
    secenek=getchar();

    if(secenek=='a' || secenek=='A')
    {
        goto nokta0;
    }

    else if(secenek=='G' || secenek=='g')
    {
        goto nokta1;
    }
    else if(secenek=='K' || secenek=='k')
    {
        goto noktason;
    }
    else
    {
        bosluk();
        printf("Yanlış Seçim Tekrar Deneyin\n");
        goto nokta1x7;
    }
}
else
{
    printf("Yanlış Giriş Yaptınız !!!");
    goto nokta1;
}

nokta2:
system("cls");
bilet_menu();
secenek=getchar();

if(secenek=='A' || secenek=='a')
{
    goto nokta0;
}

else if(secenek=='1')
{
    system("cls");
    sefer_listele();
    printf("\n\n\n");
    bilet_islem(1);

    printf("\n\n\nAna Menü İçin ( A )      Üst Menü
İçin( G )      Kapatmak İçin ( K )\n");
    nokta2x1:
```

```
fflush(stdin);
secenek=getchar();

if(secenek=='a' || secenek=='A')
{
    goto nokta0;
}

else if(secenek=='G' || secenek=='g')
{
    goto nokta2;
}
else if(secenek=='K' || secenek=='k')
{
    goto noktason;
}
else
{
    bosluk();
    printf("Yanlış Seçim Tekrar Deneyin\n");
    goto nokta2x1;
}

}

else if(secenek=='2')
{

    system("cls");
    sefer_listele();
    printf("\n\n\n");
    bilet_islem(0);

    printf("\n\n\nAna Menü İçin ( A )      Üst Menü
İçin( G )      Kapatmak İçin ( K )\n");
    nokta2x2:

    fflush(stdin);
    secenek=getchar();

    if(secenek=='a' || secenek=='A')
    {
        goto nokta0;
    }

    else if(secenek=='G' || secenek=='g')
    {
        goto nokta2;
    }
    else if(secenek=='K' || secenek=='k')
    {
        goto noktason;
    }
    else
    {
        bosluk();
        printf("Yanlış Seçim Tekrar Deneyin\n");
        goto nokta2x2;
    }
}
```



```
printf("xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n\n");
bosluk();
printf(" ( K )      Kapat\n\n");
bosluk();
printf(" ( 1 )      Sefer İşlemleri\n\n");
bosluk();
printf(" ( 2 )      Bilet İşlemleri\n\n");
bosluk();
printf("xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n\n\n\n");

}
////////////////////////////////////
void sefer_menu()
{
    system("color 1F");
    printf("\n\n\n");
    bosluk();
    printf("                SEFER İŞLEMLERİ                \n");
    bosluk();
    printf("xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n\n");
    bosluk();
    printf(" ( A )      Ana Menü\n\n");
    bosluk();
    printf(" ( 1 )      Sefer Ekle\n\n");
    bosluk();
    printf(" ( 2 )      Sefer Sil\n\n");
    bosluk();
    printf(" ( 3 )      Sefer Listele\n\n");
    bosluk();
    printf(" ( 4 )      Geçmiş Seferleri Listele\n\n");
    bosluk();
    printf(" ( 5 )      Sefer Geliri Hesapla\n\n");
    bosluk();
    printf(" ( 6 )      Tüm Sefer Sayısı\n\n");
    bosluk();
    printf(" ( 7 )      Kaptan Değiştir\n\n");
    bosluk();
    printf("xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n\n\n\n");

}
////////////////////////////////////
void bilet_menu()
{
    system("color 1F");
    printf("\n\n\n");
    bosluk();
    printf("                BİLET İŞLEMLERİ                \n");
    bosluk();
    printf("xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n\n");
    bosluk();
    printf(" ( A )      Ana Menü\n\n");
    bosluk();
    printf(" ( 1 )      Bilet Satış\n\n");
    bosluk();
    printf(" ( 2 )      Rezervasyon\n\n");
    bosluk();
    printf(" ( 3 )      Bilet İptali\n\n");
    bosluk();
    printf("xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n\n\n\n");

}
```

```
////////////////////////////////////
void bosluk()
{
    int i;
    for(i=1;i<=20;i++)
    {
        printf(" ");
    }
}

////////////////////////////////////
void sefer_sil()
{
    if((log=fopen("log.txt","a"))==NULL)
    {
        printf("Log Dosyası Açılamadı");
        exit(0);
    }

    printf("Silinecek Sefer Numarasi = ");
    int numara;
    scanf("%d",&numara);

    struct seferler *sil;
    struct koltuk_listeleri *koltuk;
    sil=ilk;

    while(sil)
    {
        if(sil->sefer_no==numara)
        {
            koltuk=sil->otobus;

            while(koltuk)
            {
                if(koltuk->durum==2)
                {
                    printf("Bu Sefere Bilet Satışı
Yapılmıştır Silinemez !!!\n\n");
                    goto son;
                }

                koltuk=koltuk->sonraki;
            }

            if(sil==ilk)
            {
                ilk=sil->sonraki;
                ilk->onceki=NULL;
            }
            else
            {
                sil->onceki->sonraki=sil->sonraki;
            }
        }
    }
}
```

```
        if (sil!=son)
        {
            sil->sonraki->onceki=sil->onceki;
        }

        else
        {
            son=sil->onceki;
        }

    }
    fprintf(log,"%d.Sefer Silindi \n",sil->sefer_no);
    dosya_guncelle(sil->tarih);
    break;
}

    sil=sil->sonraki;
}

fclose(log);

son;;
}
```

```
////////////////////////////////////
void sefer_listele()
{
    if((log=fopen("log.txt","a+"))==NULL)
    {
        printf("Log Dosyası Açılamadı");
        exit(0);
    }
    struct seferler *listele;

    listele=ilk;
    printf("S_No Guzergah          Tarih          Saat  Kaptan          Plaka
Kapasite Fiyat\n\n");
    while(listele)
    {
        if(karsilastir(bugun,listele->tarih)==0)
        {
            continue;
        }

        printf("%-4d ",listele->sefer_no);
        printf("%-18s ",listele->guzergah);
        printf("%-10s ",listele->tarih);
        printf("%-5s ",listele->saat);
        printf("%-10s ",listele->kaptan);
        printf("%-7s ",listele->plaka);
        printf("%-8d ",listele->kapasite);
        printf("%.2f ",listele->bilet_fiyati);

        printf("\n");
        listele=listele->sonraki;
    }
}
```

```
    }
    fprintf(log, "Seferler Listelendi\n");
    fclose(log);
}

////////////////////////////////////
void koltuk_listele(struct koltuk_listeleri *kol_listele)
{

    printf("\n\n\n");
    printf("No      Durum      Yolcu Adı              Cinsiyet      Fiyat\n");

    while(kol_listele)
    {
        printf("%-4d ", kol_listele->koltuk_no);
        if(kol_listele->durum==0)
            printf("%-7s ", "Boş");
        else if(kol_listele->durum==1)
            printf("%-7s ", "Rezerv");
        else if(kol_listele->durum==2)
            printf("%-7s ", "Dolu");

        printf("%-20s ", kol_listele->yolcu_adi);
        printf("%-10c ", kol_listele->cinsiyet);
        printf("%.2f\n", kol_listele->fiyat);

        kol_listele=kol_listele->sonraki;
    }
}

////////////////////////////////////
void bilet_islem(int secenek)
{
    if((log=fopen("log.txt", "a+"))==NULL)
    {
        printf("Log Dosyası Açılamadı");
        exit(0);
    }

    int sefer_numarasi;
    int koltuk_numarasi;
    char bos[30]="X";

    printf("Sefer No Seçiniz = ");
    scanf("%d", &sefer_numarasi);

    struct seferler *sefer=ilk;
    struct koltuk_listeleri *koltuk;

    geri:
```

```
while (sefer)
{
    if (sefer->sefer_no==sefer_numarasi)
    {
        tekrar:
        koltuk=sefer->otobus;
        koltuk_listele(koltuk);

        printf("\n\nKoltuk Numarası Seçiniz = ");
        scanf("%d",&koltuk_numarasi);

        while (koltuk)
        {
            if (koltuk->koltuk_no==koltuk_numarasi)
            {
                switch (secenek)
                {
                    case 0:
                        if (koltuk->durum==1 ||
koltuk->durum==2)
                        {
                            printf("Bu Koltuk Dolu
Rezervasyon Yapılamaz ");
                            goto tekrar;
                        }
                        printf("Yolcu Adı = ");
                        scanf("%s",&koltuk->
>yolcu_adi);

                        fflush(stdin);
                        tekrar2:
                        printf("Cinsiyet = ");
                        scanf("%c",&koltuk->
>cinsiyet);

                        koltuk->durum=1;
                        dosya_guncelle(sefer->
>tarih);

                        fprintf(log,"%d Nolu
Seferin %d Nolu Koltuğu Rezerv Edildi\n",sefer->sefer_no,koltuk->koltuk_no);
                        break;

                    case 1:
                        if (koltuk->durum==2)
                        {
                            printf("Bu Koltuk
Satılmış Satış Yapılamaz ");
                            goto tekrar;
                        }
                        printf("Yolcu Adı = ");
                        scanf("%s",&koltuk->
>yolcu_adi);

                        fflush(stdin);
                        printf("Cinsiyet = ");
                        scanf("%c",&koltuk->
>cinsiyet);

                        koltuk->durum=2;
                        dosya_guncelle(sefer->
>tarih);
```

```
fprintf(log, "%d Nolu
Seferin %d Nolu Koltuğu Satıldı\n", sefer->sefer_no, koltuk->koltuk_no);
break;

case 2:

koltuk->cinsiyet='X';
koltuk->durum=0;
strcpy(koltuk->

dosya_guncelle(sefer->

fprintf(log, "%d Nolu
Seferin %d Nolu Koltuğu iptal Edildi\n", sefer->sefer_no, koltuk->koltuk_no);
break;

default:
printf("Gecersiz Secim
!!!");

goto geri;
break;

}

break;

}

koltuk=koltuk->sonraki;

}

break;

}

sefer=sefer->sonraki;

}

fclose(log);

}

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

void dosya_yaz(struct seferler *yaz)
{
FILE *dosya, *dosya2;

char isim[20];
isim[0]=NULL;
strcat(isim, yaz->tarih);
strcat(isim, uzanti);

if((dosya=fopen(isim, "r"))==NULL)
{
if((dosya2=fopen("data.txt", "a"))==NULL)
```

```
        {
            printf("data dosyası açılmadı");
            exit(0);
        }
        fprintf(dosya2, "\n%s", yaz->tarih);
    }

    if((dosya=fopen(isim, "a"))==NULL)
    {

        printf("dosya acilamadi  !!!");
        exit(0);

    }

    struct koltuk_listeleri *kol_yaz;

    fprintf(dosya, "Sefer No      Güzergah      Tarih
Saat      Bilet Fiyatı      Kapasite      Kaptan      Plaka
\n\n");

    fprintf(dosya, "%-10d %-20s %-16s %-15s %-17f %-15d %-20s %-
15s\n\n", yaz->sefer_no, yaz->guzergah, yaz->tarih, yaz->saat, yaz->bilet_fiyati, yaz
>kapasite, yaz->kaptan, yaz->plaka);
    kol_yaz=yaz->otobus;

    fprintf(dosya, "\tKoltuk No   Durum   Yolcu Adı      Cinsiyet
Bilet Fiyatı\n");

    while(kol_yaz)
    {

        fprintf(dosya, "\t%-10d %-7d %-18s %-10c %f\n", kol_yaz-
>koltuk_no, kol_yaz->durum, kol_yaz->yolcu_adi, kol_yaz->cinsiyet, kol_yaz->fiyat);

        kol_yaz=kol_yaz->sonraki;

    }

    fprintf(dosya, "\n-----
-----\n");

    fclose(dosya);
    fclose(dosya2);

}

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

int karsilastir(char tarih1[], char tarih2[])
{
    struct tarihler
    {
```

```
        int gun,ay,yil;

    }t1,t2;

    char gun_1[3];
    char ay_1[3];
    char yil_1[5];
    char gun_2[3];
    char ay_2[3];
    char yil_2[5];

    gun_1[0]=tarih1[0];
    gun_1[1]=tarih1[1];
    gun_1[2]=NULL;
    t1.gun=atoi(gun_1);

    ay_1[0]=tarih1[3];
    ay_1[1]=tarih1[4];
    ay_1[2]=NULL;
    t1.ay=atoi(ay_1);

    yil_1[0]=tarih1[6];
    yil_1[1]=tarih1[7];
    yil_1[2]=tarih1[8];
    yil_1[3]=tarih1[9];
    yil_1[4]=NULL;
    t1.yil=atoi(yil_1);

    gun_2[0]=tarih2[0];
    gun_2[1]=tarih2[1];
    gun_2[2]=NULL;
    t2.gun=atoi(gun_2);

    ay_2[0]=tarih2[3];
    ay_2[1]=tarih2[4];
    ay_2[2]=NULL;
    t2.ay=atoi(ay_2);

    yil_2[0]=tarih2[6];
    yil_2[1]=tarih2[7];
    yil_2[2]=tarih2[8];
    yil_2[3]=tarih2[9];
    yil_2[4]=NULL;
    t2.yil=atoi(yil_2);

    if(t1.yil>t2.yil)
    {
        return 0;
    }
    else if(t1.yil<t2.yil)
    {
        return 2;
    }
    else
    {
        if(t1.ay>t2.ay)
        {
            return 0;
        }
    }
}
```



```
        else if (t1.ay<t2.ay)
        {
            return 2;
        }

        else
        {

            if (t1.gun>t2.gun)
            {
                return 0;
            }
            else if (t1.gun<t2.gun)
            {
                return 2;
            }
            else
            return 1;

        }

    }

}

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

void basla()
{
    bas:
    FILE *dosya;
    if ((dosya=fopen("data.txt","r"))==NULL)
    {
        printf("data.txt dosyasi bulunamadi !!!\n");

        FILE *dosya2=fopen("data.txt","w");
        fprintf(dosya,"%-10d\n",sefer_numarasi);
        fclose(dosya2);
        goto bas;

    }

    fscanf(dosya,"%d\n",&sefer_numarasi);
    char okunan[20];

    while (!feof(dosya))
    {
        fscanf(dosya,"%s\n",&okunan);

        if ((karsilastir(bugun,okunan)==1) || (karsilastir(bugun,
okunan)==2))
        {
            dosya_oku(okunan);
        }

    }

}
```

```
        sefer_no_guncelle();
        fclose(dosya);

}

////////////////////////////////////
void gecmis_oku()
{

    struct seferler *gecici_ilk=ilk,*gecici_son=son;
    ilk=NULL;
    son=NULL;

    FILE *dosya;
    if((dosya=fopen("data.txt","r"))==NULL)
    {
        printf("data.txt dosyasi bulunamadi !!!");
        exit(0);
    }

    fscanf(dosya,"%d\n",&sefer_numarasi);

    char okunan[20];

    while(!feof(dosya))
    {
        fscanf(dosya,"%s\n",&okunan);

        if((karsilastir(bugun,okunan)==0))
        {
            dosya_oku(okunan);

        }

    }

    sefer_listele();

    ilk=gecici_ilk;
    son=gecici_son;

    fclose(dosya);

}

////////////////////////////////////
void sefer_no_guncelle()
{

    FILE *dosya;
    if((dosya=fopen("data.txt","r+"))==NULL)
    {
        printf("data.txt dosyasi bulunamadi !!!");
        exit(0);
    }

    fprintf(dosya,"%-10d",sefer_numarasi);

    fclose(dosya);

}
```

```
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void gelir_hesapla()
{
    if((log=fopen("log.txt","a"))==NULL)
    {
        printf("Log Dosyası Açılamadı");
        exit(0);
    }

    printf("Hesaplamak istediginiz sefer no = ");
    int numara;
    scanf("%d",&numara);

    struct seferler *hesapla=ilk;
    while(hesapla)
    {
        if(numara==hesapla->sefer_no)
        {
            struct koltuk_listeleri *liste=hesapla->otobus;
            float toplam=0;
            while(liste)
            {
                if(liste->durum==2)
                {
                    toplam+=liste->fiyat;
                }

                liste=liste->sonraki;
            }

            printf("Seferin Toplam Geliri = %f\n\n\n",toplam);
            fprintf(log,"%d Nolu Seferin Geliri Hesaplandı\n",
hesapla->sefer_no);
            break;
        }

        hesapla=hesapla->sonraki;
    }
    fclose(log);
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void dosya_guncelle(char tarih[])
{
    char ad[20];
    ad[0]=NULL;
}
```

```
    strcat(ad,tarih);
    strcat(ad,uzanti);

    FILE *dosya;

    if((dosya=fopen(ad,"w"))==NULL)
    {
        printf("\ndosya okuma hatasi !!!");
        exit(0);
    }

    struct seferler *sefer;
    sefer=ilk;
    while(sefer)
    {

        if(strcmp(tarih,sefer->tarih)==0)
        {
            dosya_yaz(sefer);

        }

        sefer=sefer->sonraki;
    }

    fclose(dosya);
}

////////////////////////////////////
void sefer_ekle()
{
    if((log=fopen("log.txt","a"))==NULL)
    {
        printf("Log Dosyası Açılamadı");
        exit(0);
    }

    struct seferler *yer;
    yer=(seferler *) malloc(sizeof(struct seferler));
    int uyum=0;

    yer->sefer_no=sefer_numarasi++;
    tekrar:
    bosluk();printf("Guzergah = ");
    scanf("%s",&yer->guzergah);
    for(int i=0;i<=5;i++)
    {
        if(strcmp(yer->guzergah,guzergahlar[i])==0)
        {
            uyum=1;
        }
    }

    if(uyum==0)
    {
        printf("\n\n");
        bosluk();
        printf("Böyle Bir Güzergah Olamaz Tekrar Girin!!!\n\n");
        goto tekrar;
    }
}
```

```
        }

        fflush(stdin);

        bosluk();printf("Tarih = ");
        scanf("%s",&yer->tarih);

        fflush(stdin);

        bosluk();printf("Saat = ");
        scanf("%s",&yer->saat);

        fflush(stdin);

        bosluk();printf("Kaptan = ");
        scanf("%s",&yer->kaptan);

        fflush(stdin);

        bosluk();printf("Plaka = ");
        scanf("%s",&yer->plaka);

        fflush(stdin);

        bosluk();printf("Kapasite = ");
        scanf("%d",&yer->kapasite);

        fflush(stdin);

        bosluk();printf("Bilet Fiyati = ");
        scanf("%f",&yer->bilet_fiyati);

    ///

    struct koltuk_listeleri *yer2,*kol_ilk,*kol_son;

    for(int i=1;i<=yer->kapasite;i++)
    {
        yer2=(koltuk_listeleri *) malloc(sizeof(koltuk_listeleri));

        yer2->koltuk_no=i;
        yer2->durum=0;
        yer2->fiyat=yer->bilet_fiyati;
        yer2->cinsiyet='X';
        yer2->yolcu_adi[0]='X';

        if(i==1)
        {
            kol_ilk=yer2;
            kol_son=yer2;
            kol_ilk->sonraki=NULL;
            kol_ilk->onceki=NULL;
            yer->otobus=yer2;

        }
        else
        {
            kol_son->sonraki=yer2;
            yer2->onceki=kol_son;
        }
    }
}
```

```
        kol_son=yer2;
        kol_son->sonraki=NULL;

    }

}

///  

if(ilk==NULL)
{
    ilk=yer;
    son=yer;
    ilk->onceki=NULL;
    ilk->sonraki=NULL;
}

else
{
    son->sonraki=yer;
    yer->onceki=son;
    son=yer;
    son->sonraki=NULL;
}

fprintf(log,"%d Nolu Sefer Eklendi\n",yer->sefer_no);
fclose(log);
sefer_no_guncelle();
dosya_yaz(yer);
printf("\n\n");
bosluk();printf("SEFER BAŞARIYLA EKLENDİ !!!\n\n");
}

//////////////////////////////////////
void kaptan_degistir()
{
    if((log=fopen("log.txt","a"))==NULL)
    {
        printf("Log Dosyası Açılamadı");
        exit(0);
    }

    printf("Kaptanını Değiştirmek İstedığınız Sefer No = ");
    int numara;
    scanf("%d",&numara);

    struct seferler *degistir=ilk;
    while(degistir)
    {
        if(numara==degistir->sefer_no)
        {
```

```
        printf("Yeni kaptan adini giriniz = ");
        scanf("%s",&degistir->kaptan);
        dosya_guncelle(degistir->tarih);
        fprintf(log,"%d Nolu Seferin Kaptanı Değiştirildi\n",
degistir->sefer_no);
        break;
    }

    degistir=degistir->sonraki;
}

printf("Kaptan Başarıyla DEğiştirildi !!!\n\n\n");

fclose(log);
}

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

void bugun_tarih()
{
    time_t zaman;
    struct tm *zaman_bilgi;
    time(&zaman);
    strftime(bugun,80,"%d.%m.%Y",localtime(&zaman));
}

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

void dosya_oku(char isim[])
{
    char ad[20];

    ad[0]=NULL;
    strcat(ad,isim);
    strcat(ad,uzanti);

    FILE *dosya;
    if((dosya=fopen(ad,"r"))==NULL)
    {
        printf("dosya acilamadi 5 !!!");
        exit(0);
    }

    struct seferler *oku,*oku_ilk=NULL,*oku_son=NULL;

    struct koltuk_listeleri *kol_oku,*kol_ilk,*kol_son;

    while(!feof(dosya))
    {

        oku=(seferler *) malloc(sizeof(struct seferler));
```

```
fseek(dosya, sizeof(char)*135, 1);
fscanf(dosya, "\n\n");

fscanf(dosya, "%d %s %s %s %f %d %s %s\n\n", &oku-
>sefer_no, &oku->guzergah, &oku->tarih, &oku->saat, &oku->bilet_fiyati, &oku
>kapasite, &oku->kaptan, &oku->plaka);

fscanf(dosya, "\t");
fseek(dosya, sizeof(char)*61, 1);
fscanf(dosya, "\n");

for(int i=1; i<=oku->kapasite; i++)
{

    kol_oku=(koltuk_listeleri *)
malloc(sizeof(struct koltuk_listeleri));

    fscanf(dosya, "\t%d %d %s %c %f\n", &kol_oku-
>koltuk_no, &kol_oku->durum, &kol_oku->yolcu_adi, &kol_oku->cinsiyet, &kol_oku
>fiyat);

    if(i==1)
    {
        kol_ilk=kol_oku;
        kol_son=kol_oku;
        kol_ilk->sonraki=NULL;
        kol_ilk->onceki=NULL;
        oku->otobus=kol_oku;

    }

    else
    {

        kol_son->sonraki=kol_oku;
        kol_oku->onceki=kol_son;
        kol_son=kol_oku;
        kol_son->sonraki=NULL;

    }

}

if(oku_ilk==NULL)
{
    oku_ilk=oku;
    oku_son=oku;
    oku_ilk->onceki=NULL;
    oku_ilk->sonraki=NULL;
}

else
{
    oku_son->sonraki=oku;
    oku->onceki=oku_son;
    oku_son=oku;
}
```



```
        oku_son->sonraki=NULL;

    }

    fscanf(dosya, "\n");
    fseek(dosya, sizeof(char)*135, 1);
    fscanf(dosya, "\n");

}

if(ilk==NULL)
{
    ilk=oku_ilk;
    son=oku_son;
}

else
{
    son->sonraki=oku_ilk;
    oku_ilk->onceki=son;
    son=oku_son;
}

fclose(dosya);

}

////////////////////////////////////
void sefer_sayisi()
{
    if((log=fopen("log.txt", "a+"))==NULL)
    {
        printf("Log Dosyası Açılamadı");
        exit(0);
    }

    struct seferler *say=ilk;
    int sayi=0;
    while(say)
    {
        sayi++;

        say=say->sonraki;
    }

    printf("Toplam Sefer Sayisi = %d", sayi);

    fprintf(log, "Toplam Sefer Sayisi %d Olarak Gösterildi\n", sayi);
    fclose(log);
}
```