

# “ HAVALİMANI-İŞLEMLERİ “ PROJE RAPORU

1-)

```
1  #include <stdio.h>
2  #include <stdlib.h> //malloc() ve free() kodları ile bellek işlemleri
3  #include <string.h> //strcpy gibi string özelliklerine erişim
4  #include <locale.h> //setlocale kodu ile türkçe karakterlere izin
5
6  //1
7  //Düğüm tanımlaması , içinde "kişi,sonraki,şehir" değişkenlerini tutuyor.
8  struct dugum{
9      char *kisi;
10     struct dugum *sonraki;
11     char *sehir;
12 };
13
14 struct dugum *ilk=NULL; //Düğümün ilk i null dur
15 struct dugum *son=NULL; //Düğümün son u null dur
16
```

- Düğümü **struct** yardımıyla tanımlıyoruz içinde ise string tipinde **kişi**, düğüm tipinde **sonraki**, string tipinde **şehir** değişkenlerini bulundurur.
- Düğüm tipinde **ilk** tanımlıyoruz bu ilk; ilk elemanı tutucak ilk başta ilkin değerini **NULL** olarak yani boş olarak atıyoruz.
- Düğüm tipinde son tanımlıyoruz bu **son**; son elemanı tutucak ilk başta sonun değerini **NULL** olarak tanımlıyoruz.

2-)

```
17 //2
18 /*Onaybekle() metodu çoğu işlemde işlem sonrasında menu ekranına bağlanmak için
19 kullanıcının bir tuşa basmasını bekler.*/
20 int onayBekle(){
21     printf("İşlemlerinize devam etmek için bir tuşa basınız.\n");
22     getchar();
23     getchar();
24
25     return 1;
26 }
```

- OnayBekle() fonksiyonu çoğu işlemin sonrasında çağrılarak işlem sonrasında kullanıcıya devam etmek için bir tuşa basmasını söyler, **getchar()** opsiyonu ile bu işlem gerçekleştirilir, garantiye almak için 2 kere kullanılmıştır.

(1)

3-)

```
28 //3
29 /*EkranıTemizle() metodu ,parametre olarak tam sayı tipinde onayDurumunu tutar,
30 onayDurumu 1 ise onaybekle metodu çalıştırılır ve bazı durumlar gerçekleştirilir,
31 bu metod ekranı temizlemek için kullanılır çoğu işlem bitiminde kullanılır.*/
32 void ekranıTemizle(int onayDurumu){
33     if(onayDurumu==1){
34         onayBekle();
35     }
36     system("cls");
37 }
```

- Bu metod her işlem sonrasında çağrılarak ekranı temizleme işlemini gerçekleştirir. EkranıTemizle() metodu parametre olarak **onayDurumu** 'nu alır (açıklaması kod görüntüsünde var.).
- If bloğu onay durumu 1 e eşit olması durumunda çalışır ve **onayBekle()** metodunu çalıştırır.
- Sonrasında **system("cls")** komutu ile ekranı temizleme işlemi gerçekleştirilmiş olunur.

4-)

```
39 //4
40 /*UçuşEkle metodu,uçuşSayisi parametresiyle kullanılır uçuş sayısını güncellemek için,
41 kuyruğa kişi ve kişinin gideceği şehir ekler
42 */
43 int ucusEkle(int *ucusSayisi){
44     char yolcu[50];
45     char sehir[20];
46     ekraniTemizle(0);
47     printf("\n\tSeçiminiz *>>>> 1 ");
48     printf("\n\n:-: Yolcu ismi: ");
49     scanf("%s", yolcu);
50     yolcu[0]=toupper(yolcu[0]);
51
52     printf(":-: %s yolcusu hangi şehire gitmek istiyor? : ",yolcu);
53     scanf("%s", sehir);
54     sehir[0]=toupper(sehir[0]);
55
56     struct dugum *yeniDugum=malloc(sizeof(struct dugum)); //Yeni düğüm için bellekte malloc ile yer açar
57     yeniDugum->kisi = malloc(strlen(yolcu));
58     yeniDugum->sehir= malloc(strlen(sehir));
59     strcpy(yeniDugum->kisi , yolcu);
60     strcpy(yeniDugum->sehir , sehir);
61     yeniDugum->sonraki=NULL;
62
63
64     if(ilk==NULL && son==NULL){ //Eğer uçuş listesi boş ve ilk bileti alan kişiye
65
66         ilk = yeniDugum;
67         son = yeniDugum;
68         ilk->sehir = malloc(strlen(sehir));
69         strcpy(ilk->sehir, sehir);
70     }else{
71         son->sonraki = yeniDugum;
72         son = yeniDugum;
73         son->sehir = malloc(strlen(sehir));
74         strcpy(son->sehir, sehir);
75     }
76     printf("\n*****\n");
77     printf("\n\t>Yeni uçuş eklendi: \n");
78     printf("\n:-: Yolcu ismi: %s ",yolcu);
79     printf("\n:-: Rota: İstanbul --> %s \n",sehir);
80     printf("\n*****\n\n");
81     (*ucusSayisi)++;
82     return *ucusSayisi;
83 }
```

- UçuşEkle() fonksiyonu **ucusSayisi** parametresi alır,ve değerini günceller(parametre ayrıntısı kod kısmında var.).
- Bu fonksiyon uçuşa yolcu ve yolcunun gideceği şehir bilgisini eklemeye yarar.
- String tipinden **yolcu** ve **sehir** değişkenleri vardır.
- **EkraniTemizle()** metodunu çağırır sebebi ise menüyü ekrandan silmektir.
- **Scanf** ile yolcu ve şehir değişkenlerini alır ancak **toupper** kullanarak ilk harflerini büyük harfe çevirir.
- YeniDüğüm **malloc** ile düğümün eleman sayısı kadar yer açar.
- YeniDüğümün kişisi malloc ve **strlen**(Stringlerde karakter sayısını hesaplar) ile yolcunun karakteri kadar yer açar aynısını şehir değişkeni içinde yapar.
- **Strcpy** ile YeniDüğümün kişisine yolcunun değerini kopyalar , benzerini şehir içinde yapar.
- **YeniDüğümün sonraki** NULL olarak atar.
- İlk bloğu ilk eleman yoksa ve son elemanda yoksa yani uçuş listesi boşsa çalışır;
- İçinde ise ilk e ve son a yeniDüğümü atamasını yapar,son elemanın şehrine şehir değişkenini atar.

(3)

5-)

```
85 //5
86 /*UçuşListesi() metodu, dolayliKullanım parametresi ile kullanılır,
87 uçuş listesini yani yolcular ve gidecekleri şehirleri ekrana yazdırır.*/
88 void ucuşListesi(int dolayliKullanım){
89
90     if(dolayliKullanım==0){
91         ekraniTemizle(0);
92         printf("\n\tSeçiminiz *>>>> 2 ");
93     }
94
95     struct dugum *gecici=ilk;
96     int donguyeGirildi=0;
97     int sayac=1;
98     if(donguyeGirildi==0 && gecici==NULL){
99         printf("\n*****\n");
100         printf("\n >BOŞ YOLCU LİSTESİ< !! LİSTELENME GERÇEKLEŞTİRİLEMEDİ !!\n");
101         printf("\n*****\n\n");
102     }
103     else{
104         printf("\n\n***** UÇUŞ LİSTESİ *****\n\n");
105
106         while(gecici!=NULL){
107             donguyeGirildi=1;
108             printf("\t %d> %s --> %s\n",sayac, gecici->kisi, gecici->sehir);
109
110             gecici = gecici->sonraki;
111             sayac++;
112         }
113         printf("\n*****\n\n");
114     }
115 }
116 }
```

- UçuşListesi() metodu **dolaylıKullanım** parametresine sahiptir.
- İf bloğuna dolaylıKullanım değeri 0 ise girer ve ekranı temizler,ekrana seçimin 2 olduğunu yazdırır çünkü 2 e tuşlandığı zaman gelen metoddur.
- Duğum tipinden **gecici** değişkeni tanımlar ve ilk değere eşitler **sayac** değişkeni ile kaçınıcı eleman olduğu ekrana yazdırılırken kullanılır.
- 2.İf bloğu içine eğer **donguyeGirildi** değeri 0 ise ve gecici yani ilk değer boş ise girer ve ekrana boş yolcu listesi olduğunu yazdırır.
- Else bloğuna girdiğinde ekrana uçuş listesi yazdırır ;While bloğunda gecici değeri null a eşit olmayana kadar kalır ve içinde ise donguyeGirildi değeri 1 olarak atanır,ekrana kaçınıcı eleman olduğu ,yolcu ismi,şehiri yazar sırayla tüm elemanlar için bunu yapar **geciciyi gecicinin sonrakine eşitlemek** buna olanak sağlar,sayaçı arttırır .

## 6-)

```
118 //6
119 /*UcuşSil() metodu ,ucuşSayısı parametresi ile gelir ve bu sayıyı günceller,
120 görevi uçuştan yolcuyu ve gideceği şehri silmektir.*/
121 int ucuşSil(int *ucuşSayisi) {
122     ekraniTemizle(0);
123     printf("\n\tSeçiminiz *>>>> 6 ");
124     int sayac=1;
125     if (ilk == NULL) { //Herhangi bir uçuş yoksa liste boşsa
126         printf("\n*****\n");
127         printf("\n\t*> YOLCU LİSTESİ BOŞTUR\n");
128         printf("\n*****\n\n");
129     } else { //Liste boş değilse
130         int silinecekUcuş=0;
131         printf("\n\t*> Silmek istediğiniz uçuş hangisidir? ");
132         ucuşListesi(1);
133         scanf("%d",&silinecekUcuş);
134
135         while(silinecekUcuş<=0){
136             printf("\n*****\n");
137             printf("\n\t*> !! SEÇİMİ YANLIŞ YAPTINIZ !!\n");
138             printf("YENİ SEÇİM: ");
139             scanf("%d",&silinecekUcuş);
140             printf("\n*****\n");
141             system("cls");
142         }
143
144         struct dugum *onceki = NULL;
145         struct dugum *simdiki = ilk;
146
147         while (simdiki != NULL && sayac < silinecekUcuş) {
148             onceki = simdiki;
149             simdiki = simdiki->sonraki;
150             sayac++;
151         }
152
153         if (simdiki == NULL ) {
154             printf("\n*****\n");
155             printf("\nUçuş bulunmamaktadır.\n");
156             printf("\n*****\n\n");
157         } else {
158             if (onceki == NULL) { // İlk elemanı sileceksek
159                 ilk = simdiki->sonraki;
160             }
161             else if(son == simdiki) { // Son elemanı siliceksek
162                 son = onceki;
```

- UcuşSil fonksiyonu **ucuşSayisi** parametresi alır.Uçuş sildiğinde değerini azaltır.
- Sayac ı 1 olarak atar
- İf bloğuna eğer ilk eleman yoksa girer ve boş liste olduğunu yazar.
- Else bloğuna liste boş değilse girer **silinecekUcuş** adlı değişken tanımlar ve 0 atar,silinmek istenilen ucusu sorar ve duğum tipinden onceki tanımlar ve null atar,duğum tipinden şimdiki tanımlar ve ilk atar While bloğuna şimdiki null değilse ve sayac silinecek uçuştan küçükse girer içinde ise oncekini şimdiki olarak atar,şimdikini şimdikinin sonrakisi olarak atar ve sayacı 1 arttırır.
  - İf bloğu içine şimdiki null ise girer ve uçuş bulunmuyor diye yazdırır.

(5)

DEVAMI DİĞER SAYFADA



## 6.1 -)

```
160 }
161 else if(son == simdiki) { // Son elemanı siliceksek
162     son = onceki;
163     son->sonraki=NULL;
164 }
165 else {
166     onceki->sonraki = simdiki->sonraki;
167 }
168
169 printf("\n*****\n");
170 printf("\n%d . yolcu olan %s kişisi uçuş listesinden çıkarıldı.\n",sayac, simdiki->kisi);
171 printf("\n*****\n\n");
172 free(simdiki->kisi);
173 free(simdiki->sehir);
174 free(simdiki);
175
176 (*ucusSayisi)--;
177 if(ilk==NULL){
178     son=NULL;
179 }
180 }
181 }
182
183 return *ucusSayisi;
184 }
```

- Else girince içinde if bloğu vardır ona eğer **onceki** null ise yani ilk elemanı siliceksek girer ve **ilk** i **şimdiki** **sonrakine** atar.
- Else if e ise son şimdikiye eşitse girer yani son elemanı siliceksek içinde ise **son** a **öncekini** atar,sonun sonrakisine null atar.
- Else bloğuna bu durumlar dışında girer yani ortadan bir eleman silinecekse girer ,içinde ise **öncekinin sonrakisine , şimdiki** **sonrakisini** atar.
- Ekranı çıkarılan elemanın numarasını ve ismini çıkarıldı olarak yazar.
- **Free** komutları ile şimdiki
- Uçuşsayısını azaltır bir yolcu silindiği için
- If bloğuna ilk null ise girer yani kişi silindikten sonra eleman kalmadıysa,içinde ise son elemana da null atar.
- Return ile uçuşSayısını döndürür.

7-)

```
186 //7
187 /*UçuşSayısıGöster() metodu , uçuşSayısı parametresi ile gelir,
188 aktif uçuş sayısını göstermek için kullanılır.*/
189 void uçuşSayısıGöster(int uçuşSayısı){
190     ekraniTemizle(0);
191     printf("\n\tSeçiminiz *>>>> 5 ");
192     printf("\n*****\n");
193     printf("\n\t*>Aktif uçuş sayısı: %d \n",uçuşSayısı);
194     printf("\n*****\n\n");
195 }
```

- *UçuşSayısıGöster metodu **uçuşSayısı** parametresi alır*
- *Aldığı uçuşSayısı parametresi ile aktif uçuş sayısını ekrana yazdırır ve kaç yolcu olduğunu kullanıcı öğrenmiş olur.*

8-)

```
197 //8
198 /*YolcuListesi() metodu , dolaylıKullanım parametresi ile gelir,
199 sadece yolcu isimlerini ekrana yazdırmak istendiğinde kullanılır.*/
200 void yolcuListesi(int dolaylıKullanım){
201     ekraniTemizle(0);
202     if(dolaylıKullanım==0){
203         printf("\n\tSeçiminiz *>>>> 3 ");
204     }
205
206     struct dugum *gecici=ilk;
207
208     int sayac=1;
209     if(gecici!=NULL){
210         printf("\n\n***** UÇAKTAKİ YOLCU LİSTESİ *****\n\n");
211         while(gecici!=NULL){
212             printf("\t %d> %s\n",sayac, gecici->kisi);
213             gecici = gecici->sonraki;
214             sayac++;
215         }
216         printf("\n*****\n\n");
217     }
218
219     else{
220         printf("\n*****\n");
221         printf("\n >BOŞ YOLCU LİSTESİ< !! LİSTELENME GERÇEKLEŞTİRİLEMEDİ !!\n");
222         printf("\n*****\n\n");
223     }
224 }
225 }
```

- *YolcuListesi metodu **dolaylıKullanım** parametresi alır*
- *Düğüm tipinden **gecici** tanımlar ve değer **ilk** atar.*
- *İf bloğuna gecici null a eşit değilse girer yani ilk eleman boş değilse ,içinde ise While vardır while ı gecici null a eşit olmadığı sürece çalıştırır içinde ise yolcu ismi ve yolcunun şehrini yazdırır sonrasında gecici ye gecicinin sonrakisini atar sayacı arttırır.*
- *Else bloğuna diğer durumlar için girer boş yolcu listesi olduğunu yazdırır.*

(7)

## 9-)

```
227 //9
228 /*ŞehirListesi() metodu , gidilecek şehirler listelenmek istediğinde kullanılır.*/
229 void şehirListesi(){
230     ekranıTemizle(0);
231     printf("\n\tSeçiminiz *>>>> 4 ");
232     struct dugum *gecici=ilk;
233
234     int sayac=1;
235     if(gecici!=NULL){
236
237         printf("\n\n***** GİDİLECEK ŞEHİRLERİN LİSTESİ *****\n\n");
238         while(gecici!=NULL){
239             printf("\t %d> İstanbul --> %s\n",sayac, gecici->şehir);
240             gecici = gecici->sonraki;
241             sayac++;
242         }
243         printf("\n*****\n\n");
244     }
245     else{
246
247         printf("\n*****\n\n");
248         printf("\n\n >BOŞ Şehir LİSTESİ< !! LİSTELENME GERÇEKLEŞTİRİLEMEDİ !!\n\n");
249         printf("\n*****\n\n");
250     }
251 }
252
253 }
```

- ŞehirListesi metodu ,gidilecek şehirleri listelemeye yarar.
- EkranıTemizle () metodu ile ekranı temizler.
- Struct dugum tipinden **geçici** pointer tanımlanır ve değer olarak ilk i alır.
- If bloğuna gecici null a eşit değilse girer yani boş elemanı göstermediği sürece ,içinde ise şehiri while ile yazdırır sonra sonraki elemanlar varsa yazdırmak için geçicinin değerini günceller geciciyi gecicinin sonrakisi olarak atar. Sayacı 1 arttırır.
- Else bloğuna ise if e girmezse girer yani gecici boş elemanı gösteriyorsa .İçinde ise boş şehir olduğunu ve listeleme yapılamadığını yazdırır.

## 10-)

```
246 //10
247 /*Menu() metodu , havalimanı menüsünü ekrana yazdırır.*/
248 void menu(){
249     printf("\n\n***** HAVALİMANI MENÜSÜNE HOŞGELDİNİZ *****\n\n");
250     printf("\t-> Uçuş eklemek için '1' 'e basınız.\n");
251     printf("\t-> Uçuşları listelemek için '2' 'ye basınız.\n");
252     printf("\t-> Tüm yolcu isimlerini listelemek için '3' 'e basınız.\n");
253     printf("\t-> Gidilecek tüm şehirleri listelemek için '4' 'e basınız.\n");
254     printf("\t-> Şu andaki toplam uçuş sayısını görmek için '5' 'ye basınız.\n");
255     printf("\t-> Uçuş silmek için '6' 'ya basınız.\n");
256     printf("\t-> Çıkış yapmak için '7' 'ye basınız.\n");
257     printf("\n\n***** ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ *****\n\n");
258 }
259 }
```

- Menu metodu menüyü içinde bulundurur kullanıcın çoğu zaman gördüğü arayüzdür.
- Hangi işlemlerin olduğunu görmüş,öğrenmiş olur,1,2,3,4,5,6,7 tuşlarına basarak işlemleri gerçekleştirebilir.



## Main-)

```
270 //MAIN
271 /*Main fonksiyonu kodun çalıştırıldığı fonksiyondur tüm işlemler burda gerçekleşir.
272 İçinde setlocale kodu ile türkçe karakterlere izin vermiş oluyoruz.*/
273 int main(){
274     setlocale(LC_ALL, "Turkish"); //Türkçe karakterlere izin vermiş oluyoruz.
275     char secim; //
276     int durum=1;
277     int ucusSayisi=0;
278
279     do{
280         menu();
281
282         printf("\n\tSeçiminiz *>>>> ");
283         scanf(" %c",&secim);
284         char sehir[30]="";
285
286
287         switch(secim){
288
289             case '1': //Uçuş eklemeyi gerçekleştir.
290                 ucusEkle(&ucusSayisi);
291                 ekraniTemizle(1);
292                 break;
293
294             case '2': //Uçuşları listelemeyi gerçekleştir .
295                 ucusListesi(0);
296                 ekraniTemizle(1);
297                 break;
298
299             case '3': //Uçuştaki yolcu listesini gösterir .
300                 yolcuListesi(0);
301                 ekraniTemizle(1);
302                 break;
303
304             case '4': //Gidilecek şehirlerin listesini gösterir .
305                 sehirListesi(0);
306                 ekraniTemizle(1);
307                 break;
308
309             case '5': //Kaç tane uçuş olduğunu gösterir.
310                 ucusSayisiGoster(ucusSayisi);
311                 ekraniTemizle(1);
312                 break;
```

- Main metodu kodun çalıştırıldığı ve satır satır işlendiği yerdir.
- SetLocale kodundan yararlanarak türkçe karakterlere izin verilmiş olunur.
- Durum değişkeni atanır ve başlangıç değeri 1 olarak verilir.(bu değişken while koşulundadır ve çıkış seçeneği seçildiğinde durum değişkeni 0 olarak atanır.).
- UcusSayisi değişkeni tanımlanır ve başlangıç değeri olarak 0 atanır.(mevcut uçuş sayısını güncellerken ve gösterirken kullanılır birkaç metoda parametre olarak gönderilmiştir.).
- Do while in içine girildiğinde menu metodunu çağırıp ekrana seçenekleri sunar .30 karakter tutabilen sehir değişkeni tanımlanır.
- Switch case içinde seçilen seçime göre yönlendirme yapılır.

## Main.1-)

```
313
314     case '6':        //Uçuşu silmeyi gerçekleştir.
315         ucusSil(&ucusSayisi);
316         ekraniTemizle(1);
317         break;
318
319     case '7':        //Programı sonlandırır.
320         ekraniTemizle(0);
321         sleep(1);
322         printf("\n\n\n\n\n\n\t\t*****\n");
323         sleep(1,5);
324         printf("\n\t\t.../.../.../.../.../.../ Program kapatılıyor .../.../.../.../.../.../");
325         sleep(1,8);
326         printf("\n\t\t*****\n\n");
327         durum=0;
328         break;
329
330     default:         //Geçersiz seçim yapıldığı için doğru seçim yapmaya zorluyor.
331         ekraniTemizle(0);
332         printf("!!!Lütfen Geçerli Bir seçim Yapınız!!!\n");
333         break;
334 }
335 }while(durum==1);
336
337 return 0;
338
```

1. *ucusEkle()* metodu sayesinde **uçuş ekleme** işlemleri gerçekleştirilir.
2. *ucusListesi()* metodu sayesinde **uçuşları listelemeyi** gerçekleştirir.
3. *yocuListesi()* metodu sayesinde **yolcu isimlerini listelemeyi** gerçekleştirir.
4. *sehirListesi()* metodu sayesinde **gidilecek şehirleri listelemeyi** gerçekleştirir.
5. *ucusSayisiGoster()* metodu sayesinde **mevcut uçuş sayısını** gösterir.
6. *ucusSil()* metodu sayesinde istenilen **uçuşları silmeyi** gerçekleştirir.
7. **Programı sonlandırır.**

Default. 1' den 7' e kadar bir seçim dışında bir seçim yapılırsa geçerli seçim yapmasını söyler ve döngünün başına geçer.

**KOD KISMI BİTTİ**

## Menü Ekranı

```
***** HAVALİMANI MENÜSÜNE HOŞGELDİNİZ *****

-> Uçuş eklemek için '1' 'e basınız.
-> Uçuşları listelemek için '2' 'ye basınız.
-> Tüm yolcu isimlerini listelemek için '3' 'e basınız.
-> Gidilecek tüm şehirleri listelemek için '4' 'e basınız.
-> Şu andaki toplam uçuş sayısını görmek için '5' 'ye basınız.
-> Uçuş silmek için '6' 'ya basınız.
-> Çıkış yapmak için '7' 'ye basınız.

***** ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ *****

Seçiminiz *>>>>
```

## Uçuş Ekleme Ekranı

```
Seçiminiz *>>>> 1

:-: Yolcu ismi: ahmet
:-: Ahmet yolcusu hangi şehire gitmek istiyor? : ankara

*****

*>Yeni uçuş eklendi:

:-: Yolcu ismi: Ahmet
:-: Rota: İstanbul --> Ankara

*****

İşlemlerinize devam etmek için bir tuşa basınız.
|
```

## Uçuş Listeleme Ekranı

```
Seçiminiz *>>>> 2

***** UÇUŞ LİSTESİ *****

1> Ahmet --> Ankara
2> Berkant --> Tokat
3> Omer --> Kayseri
4> Ilkay --> Samsun

*****

İşlemlerinize devam etmek için bir tuşa basınız.
```

## Yolcu Listeleme Ekranı

```
Seçiminiz *>>>> 3

***** UÇAKTAKİ YOLCU LİSTESİ *****

1> Ahmet
2> Berkant
3> Omer
4> Ilkay

*****

İşlemlerinize devam etmek için bir tuşa basınız.
```

## Gidilecek Şehirleri Listeleme Ekranı

```
Seçiminiz *>>>> 4

***** GİDİLECEK ŞEHİRLERİN LİSTESİ *****

1> İstanbul --> Ankara
2> İstanbul --> Tokat
3> İstanbul --> Kayseri
4> İstanbul --> Samsun

*****

İşlemlerinize devam etmek için bir tuşa basınız.
```

## Mevcut Uçuş Sayısını Gösterme Ekranı

```
Seçiminiz *>>>> 5

*****

*>Aktif uçuş sayısı: 4

*****

İşlemlerinize devam etmek için bir tuşa basınız.
```



## Uçuş Silme Ekranı

```
Seçiminiz *>>>> 6
*> Silmek istediğiniz uçuş hangisidir?

***** UÇUŞ LİSTESİ *****

1> Ahmet --> Ankara
2> Berkant --> Tokat
3> Omer --> Kayseri
4> Ilkay --> Samsun

*****

2

*****
2 . yolcu olan Berkant kişisi uçuş listesinden çıkarıldı.

*****

İşlemlerinize devam etmek için bir tuşa basınız.
```

## Uçuş Silindikten Sonra

## Mevcut Uçuş Sayısını Gösterme Ekranı

```
Seçiminiz *>>>> 5
*****

*>Aktif uçuş sayısı: 3

*****

İşlemlerinize devam etmek için bir tuşa basınız.
```

## Uçuş Silindikten Sonra

## Uçuş Listeleme Ekranı

```
Seçiminiz *>>>> 2

***** UÇUŞ LİSTESİ *****

1> Ahmet --> Ankara
2> Omer --> Kayseri
3> Ilkay --> Samsun

*****

İşlemlerinize devam etmek için bir tuşa basınız.
|
```

## Programın Kapatılma Ekranı

```

*****/*****/*****
.../.../.../.../.../.../ Program kapatılıyor .../.../.../.../.../.../
*****/*****/*****

-----
Process exited after 329.9 seconds with return value 0
Press any key to continue . . .
```