**Algoritma ve Programlama II**

Proje 2  
Havayolu Bilet Satış Sistemi

05100000003 – Abdulbaki Furkan TANRIVERDİ  
05110000064 – Onur Cem ŞENEL

**Teslim Tarihi:** ...03.2012

ANALİZ

Bizden bir havayolu şirketinin günlük uçuşlarının ve bu uçuşlara ilişkin yapılan bilet satışlarının takip edilmesini sağlayacak bir uygulama geliştirilmesi istenmektedir. Bu amaçla uçuşlara ilişkin aşağıdaki bilgilere, hem uçuş numaraları hem kalkış yerleri aracılığıyla hızlı erişilebilmesi istenmektedir.

•Uçuş numarası

•Kalkış yeri

•Varış yeri

•Kalkış zamanı

•Koltuk kapasitesi

•Başlangıç bilet fiyatı

Uçuş bilgilerine uçuş numarası aracılığıyla erişmek için, 9000 elemanlı bir işaretçi dizisi kullanmalıyız. Çünkü uçuş numaraları 1000-9999 arasında(9000 farklı uçuş) arasında olmalı. Uçuş bilgilerine kalkış yeri aracılığıyla erişmek için ise kalkış yeri aynı harf ile başlayan uçuşların bilgilerinin kalkış yerlerine göre alfabetik sırada (kalkış yeri aynı olan uçuşlar, kalkış zamanlarına göre artan sırada) tutuldukları liste-başı düğümüne sahip dairesel tek-bağlı listelerden(çakışma listeleri) oluşan bir hash tablosu kullanıcaz. Böylece istediğimiz uçuşlara hem uçuş numarasından hem de kalkış yerinden hızlı erişme imkanı sağlamış olacağız.

Ayrıca her uçuş için yapılan bilet satışlarına ilişkin aşağıdaki bilgilere de, yolcuların TC kimlik numaraları aracılığıyla hızlı erişilmesi istenmektedir:

•Yolcunun TC kimli numarası

•Biletin fiyatı

Hem uçuş için hem de bilet için struct tanımlamak işimizi kolaylaştıracaktır. Uçuşlar için liste-başı düğümüne sahip dairsel bağlı liste, satılan biletler için TC kimlik numaralarına göre artan sırada tek-baplı liste kullanacağız.

Bilet fiyatı hesaplaması için aşağıdaki standartlar kullanılacaktır:

|  |  |
| --- | --- |
| **Boş Koltuk Oranı (BKO) (%)** | **Bilet fiyatı (TL)** |
| 100,00≥BKO>90,00 | Başlangıç Bilet Fiyatı (BBF) |
| 90,00≥BKO >80,00 | BBF\*1,1 |
| … | … |
| 10,00 ≥BKO >0,00 | BBF\*1,9 |

Bilet fiyatının hesaplanması için “if-else” merdiveni yardımcı olabilir tabii ki ama bu uzun ve karmaşık olacaktır. Bu hesaplama için uygun formül bulunmalıdır.

Programı bize bir menü sunmakta. Menüde bize uygun olanı seçerek istediğimiz kontrolleri yapabilmeliyiz.

TASARIM

Programımız farklı türde ve çok sayıda işlemler gerektirdiği için olabildiğince fonsiyonlara bölmeye çalıştık. Çünkü fonksiyon sayısı azaldıkça “main” fonksiyon daha kalabalık olacak ve programın okunabilirliği ve anlaşılabilirliği azalacaktır.

Tasarımdaki önemli noktalar ise şunlar:

•Hash fonksiyonları

•Uçuşların kalkış yerlerine göre sıralamak

•Dairesel bağlı listede, tek bağlı liste kullanmak

•Uçuş ve bilet iptali

Tüm istenilenleri aşağıdaki fonksiyonlar yardımıyla yapmaya çalıştık.

Programda Kullanılan Yapılar

**•struct** **bilet{**

**char** TC[12];

**float** fiyat;

**struct** bilet \*sonraki;

**};**

Varolan uçuşlar hakkında satılan biletleri tutmak için oluşturduğumuz ve içinde TC kimlik numarası ve bilet fiyatını tutan struct.

**•struct ucus{**

**int** ucusNo;

**char** kalkisYeri[MAX];

**char** varisYeri[MAX];

**int** zaman;

**int** kapasite;

**int** bosKoltuk;

**float** fiyat;

**struct ucus \***kalkisYerineGoreSonraki;

**struct bilet \***sonraki;

**};**

Her uçuş yukarıda sıralanan verileri tutmaktadır. “**struct ucus \***kalkisYerineGoreSonraki” ile bir sonraki uçuş düğümünün adresi tutulmakta, “**struct bilet \***sonraki” ise görüldüğü gibi **struct bilet** tipinde. Bunun nedeni, biletlerin yapısı farklı. Fakat adresi için uçuş düğümünde bilet tipinde bir işaretçi yeri ayırıyoruz.

Programda Kullanılan Fonksiyonlar

**•void beklet(long int);**

Bu fonksiyon sadece programın çıktılarını yavaşlatmak amacıyla kullanılmıştır. Bu sayede programın çıktıları ve hata mesajları kolayca okunabilir hale gelmiştir.

**•void baslikYazdir();**

Her ekran çıktısında tekrar-tekrar yazmaktansa bunu da bir fonksiyon haline getirip gerekli yerlerde çağırdık.

**•void yazdir(UCUS);**

“baslikYazdir” fonksiyonunda olduğu gibi bir uçuşa ait bilgileri bu fonksiyon yardımıyla yazdırıyoruz. Bu fonksiyon, varolan ve uçuş numarası alınan uçuşun istenilen bilgilerini ekrana yazdırıyor.

**•void biletYazdir(BILET \*);**

Bir uçuşa ait satılan biletler ekrana bastırılıyor. Bu fonksiyona parametre olarak uçuş düğümündeki bilet işaretçisinin kendisi call-by-value ile aktardık.

**•void biletListele(UCUS \*\*);**

Kullanıcıdan TC kimlik numarası alınarak, hashUcusNo[] dizisi baştan aşağı gezilerek, varolan uçuş düğümlerindeki biletler (varsa) kontrol ediliyor. Eğer bulunrsa o uçuşun bilgileri ekrana bastırılıyor. Bu fonksiyonun paramatresi, işaretçi dizisinin ilk elemanını adresi.

**•void ucusGoruntule(UCUS \*);**

Bu fonksiyonda kullanıcıdan görüntülemek istediği uçuşun numarası alınıyor ve daha sonra bu dosyaya parametre olarak gönderilen dosyadan gerekli satır ulaşılıp ekrana o uçuşa ait bilgiler yazdırılıyor.

**•void kucukHarfeCevir(char \*);**

Projemizde bizden küçük-büyük harf ayrımı yapılmaması istenmiş. Eğer kalkış-varış yerleri uçuş eklenirken alındığı gbi düğümde saklansaydı, kullanıcı o kelimeyi harfleri büyük-küçük karışık girerse istediğibilgiye ulaşamayabilirdi. Bizde o yüzden kalkış ve varış yerleri alındığında bellekte bütün harfleri küçük yapıp saklıorz. Kullanıcı ne girerse girsin arama yapmadan önce de o isimler küçük harfe çevrilip öyle arama yapılıyor.

**•void kalkisYerineGoreListele(UCUS \*);**

Kullanıcıdan sorgulamak istediği ilin adı alınıyor ve o ilin baş harfine göre bağlı olduğu düğümün ilk elemanına gidiliyor. Her düğümün kalkış yeri ile karşılaştırma yapılıyor ve eğer eşleşen olursa o uçuşun bilgileri ekrana basılıyor.

**•void kalkisvarisYerineGoreListele(UCUS \*);**

Kullanıcıdan sorgulamak istediği kalkış ve varış yerlerinin adı alınıyor ve kalkış adının baş harfine göre bağlı olduğu düğümün ilk elemanına gidiliyor. Kalkış yeri aynı olan her düğümün varış yeri, girilen varış yeri ile karşılaştırıyor ve eğer eşleşen olursa o uçuşun bilgileri ekrana basılıyor.

**•void ucusVeBiletListele(UCUS \*\*);**

Kullanıcıdan bilgilerini görmek istediği uçuşun numarası alınarak, uçuş bilgileri ve o uçuşa ilişkin satılan biletlerin bilgileri (varsa), yolcuların TC kimlik numaralarına göre artan sırada listeleniyor. Ayrıca toplam bilet sayısı, toplam kazanç ve ortalama bilet fiyatı da tutulan değişkenler yardımıyla ekrana bastırılıyor. Parametre olarak hashUcusno[] dizisinin başlangıç indexinin adresi gönderiliyor.

**•void dolulugaGoreListele(UCUS \*);**

Koltuk doluluk oranı %50’nin altında olan uçuşların bilgileri, kalkış yerlerine göre alfabetik sırada listeleniyor. Kalkış yerlerinin baş harflerine göre sıraladığımız hash tablosu geziliyor. Varolan uçuşların doluluk oranları hesaplanarak belirlenen limitin altında ise ekran o uçuş yazdırılıyor.

**•void kalkisYerineGoreSiraliEkle(UCUS \*,UCUS \*);**

Ucuş eklenirken kalkış yerinin baş harfine göre ilgili hash düğümünün adresi fonksiyona gönderiliyor. Kalkış yerlerine göre sıralarken iki kıstasımız var. Birincisi uçuşlar kalkış yerlerine göre alfabetik sırada olmalı; ikincisi ise kalkış yeri aynı olan uçuşlar kalkış zamanlarına göre küçükten büyüğe sıralı olmalı.

**•void tcKimligeGoreSiraliEkle(BILET \*\*,BILET \*);**

Bu fonksiyona ucus düğümünün içindeki struck bilet işaretçisi için ayırdığımız yerin adresi ve eklenecek bilet düğümü gönderiliyor. Biletler TC kimlik numaralarına göre artan sırada sıralanağı için gerekli filtrelemelerden sonra uygun yere ekleniyor.

**•int ucusNoAl();**

Programın çok fazla yerinde kullanıcıdan uçuş numarasını istiyoruz ve her seferinde kontrol ediyoruz. Biz ise programları olabildiğince kısaltmak istediğimiz için çok yerde kullanılan bu sorguyu fonksiyon yapmaya karar verdik. Kontrol fonksiyonda yapılıyor ve geriye uçuş numarasını döndürüyor.

**•int menuSecim();**

Koşullu bir döngü içerisinde ekrana menüyü yazdıran ve kullanıcıdan aldığı seçime göre gerekli fonksiyonları çağıran fonksiyon.

**•float biletFiyatHesapla(UCUS);**

Boş koltuk oranına göre bilet fiyatı hesaplıyor. Analiz kısmında belirtildiği gibi bu hesaplama için “if-else” merdiveni yerine çok daha kısa ve etkin olan bir “for döngüsü” kullanılıyor.

**•UCUS \*dugumCikar(UCUS \*,UCUS \*);**

Cikarilacak olan düğümün uçuş numarası kullanıcıdan alınıyor ve kalkış yerine göre hash tablosundaki ilgili liste-başı düğümünün adresi fonksiyona gönderiliyor. Daha sonra ilgili düğüme ulaşıyor ve düğümler arasındaki bağlarda gerekli değişimleri yapılıyor ve o düğüm geri döndürülüyor.

**•BILET \*tcKimligeGoreAraVeCikar(BILET \*\*,char \*);**

Bilet iptali için kullandığımız bu fonksiyon, kullanıcıdan biletini silmek istediği uçuşun numarasını alıyor ve o uçuşun struct bilet tipindeki başlangıç adresinin adresi ve kullanıcının TC kimlik numarası parametre olarak fonksiyona gönderiliyor. Daha sonra o uçuştaki bilet (varsa) ilgili düğümler değiştirilerek bilet düğümü geri döndürülüyor.

Programcı Kataloğu

Analiz, tasarım, gerçekleştirim, test ve raporlama için yaklaşık 15-20 saat harcandı.

Kaynak Kod

// Havayolu Bilet Satış Sistemi

Havayolu Bilet Satış Sistemi

// Sorgulama Programı

#include <stdio.h>

#include <stdlib.h>

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <ctype.h>**

**#include <windows.h>**

**#define MAX 21**

**struct bilet{**

**char TC[12];**

**float fiyat;**

**struct bilet \*sonraki;**

**};**

**struct ucus{**

**int ucusNo;**

**char kalkisYeri[MAX];**

**char varisYeri[MAX];**

**int zaman;**

**int kapasite;**

**int bosKoltuk;**

**float fiyat;**

**struct ucus \*kalkisYerineGoreSonraki;**

**struct bilet \*sonraki;**

**};**

**typedef struct bilet BILET;**

**typedef struct ucus UCUS;**

**void beklet(long int);**

**void baslikYazdir();**

**void yazdir(UCUS);**

**void biletYazdir(BILET \*);**

**void biletListele(UCUS \*\*);**

**void ucusGoruntule(UCUS \*);**

**void kucukHarfeCevir(char \*);**

**void kalkisYerineGoreListele(UCUS \*);**

**void kalkisvarisYerineGoreListele(UCUS \*);**

**void ucusVeBiletListele(UCUS \*\*);**

**void dolulugaGoreListele(UCUS \*);**

**void kalkisYerineGoreSiraliEkle(UCUS \*,UCUS \*);**

**void tcKimligeGoreSiraliEkle(BILET \*\*,BILET \*);**

**int ucusNoAl();**

**int menuSecim();**

**float biletFiyatHesapla(UCUS);**

**UCUS \*dugumCikar(UCUS \*,UCUS \*);**

**BILET \*tcKimligeGoreAraVeCikar(BILET \*\*,char \*);**

**int main()**

**{**

**int i,secim,ucusNo,biletSay;**

**float fiyat;**

**char sorgu,zaman[5],TC[12];**

**UCUS \*hashUcusNo[9000]={NULL};**

**UCUS hashListeBasi[26];**

**UCUS \*ucus,\*cikarilan;**

**BILET \*bilet,\*onceki,\*gecici;**

**system("title Havayolu Bilet Satis Sistemi");**

**system("color 8f");**

**printf("Program yukleniyor");**

**for(i=0;i<3;i++)**

**{**

**beklet(150000000);**

**printf(".");**

**}**

**printf("\n");**

**for(i=0;i<26;i++)**

**hashListeBasi[i].kalkisYerineGoreSonraki=&hashListeBasi[i];**

**do{**

**secim=menuSecim();**

**switch(secim)**

**{**

**case 1:**

**ucusNo=ucusNoAl();**

**if(hashUcusNo[ucusNo]==NULL)**

**{**

**ucus=malloc(sizeof(UCUS));**

**ucus->ucusNo=ucusNo;**

**printf("Kalkis yeri: ");**

**scanf("%s",ucus->kalkisYeri);**

**kucukHarfeCevir(ucus->kalkisYeri);**

**printf("Varis yeri: ");**

**scanf("%s",ucus->varisYeri);**

**kucukHarfeCevir(ucus->varisYeri);**

**printf("Kalkis zamani (ss:dd): ");**

**scanf("%s",zaman);**

**ucus->zaman=((zaman[0]-48)\*1000+(zaman[1]-48)\*100+(zaman[3]-48)\*10+zaman[4]-48);**

**printf("Kapasite: ");**

**scanf("%d",&ucus->kapasite);**

**printf("Baslangic fiyati: ");**

**scanf("%f",&ucus->fiyat);**

**ucus->bosKoltuk=ucus->kapasite;**

**hashUcusNo[ucusNo]=ucus;**

**ucus->sonraki=NULL;**

**kalkisYerineGoreSiraliEkle(&hashListeBasi[toupper(ucus->kalkisYeri[0])-'A'],ucus);**

**printf("\tUcus eklendi\n");**

**}**

**else**

**{**

**printf("\t%d numarali ucus zaten var!\n",ucusNo+1000);**

**}**

**break;**

**case 2:**

**ucusNo=ucusNoAl();**

**if(hashUcusNo[ucusNo]==NULL)**

**{**

**printf("\t%d numarali ucus yok!\n",ucusNo+1000);**

**}**

**else**

**{**

**ucus=hashUcusNo[ucusNo];**

**printf("Yeni kalkis saati (ss:dd): ");**

**scanf("%s",zaman);**

**cikarilan=dugumCikar(&hashListeBasi[toupper(ucus->kalkisYeri[0])-'A'],ucus);**

**ucus->zaman=((zaman[0]-48)\*1000+(zaman[1]-48)\*100+(zaman[3]-48)\*10+zaman[4]-48);**

**kalkisYerineGoreSiraliEkle(&hashListeBasi[toupper(ucus->kalkisYeri[0])-'A'],ucus);**

**printf("\tKalkis zamani guncellendi\n");**

**}**

**break;**

**case 3:**

**ucusNo=ucusNoAl();**

**if(hashUcusNo[ucusNo]==NULL)**

**{**

**printf("\t%d numarali ucus yok!\n",ucusNo+1000);**

**}**

**else**

**{**

**ucus=hashUcusNo[ucusNo];**

**do{**

**printf("Satin almak istediginiz bilet sayisi: ");**

**scanf("%d",&biletSay);**

**if(biletSay>10)**

**printf("\tBir seferde en fazla 10 bilet satin alabilirsiniz!\n");**

**}while(biletSay>10);**

**if(ucus->bosKoltuk<biletSay)**

**{**

**printf("\tBu ucusta yeterli sayida bos yer yok!\n");**

**}**

**else**

**{**

**fiyat=biletFiyatHesapla(\*ucus);**

**printf("Bir biletin fiyati: %.2f\n"**

**"Toplam fiyat: %.2f\n",fiyat,biletSay\*fiyat);**

**printf("\tOnayliyor musunuz? (e,h) ");**

**fflush(stdin);**

**scanf("%c",&sorgu);**

**if(sorgu=='e')**

**{**

**printf("\n Yolcularin TC kimlik numaralarini giriniz...\n");**

**beklet(200000000);**

**for(i=0;i<biletSay;i++)**

**{**

**bilet=malloc(sizeof(BILET));**

**printf("%4d. yolcunun TC kimlik numarasi: ",i+1);**

**scanf("%s",bilet->TC);**

**bilet->fiyat=fiyat;**

**tcKimligeGoreSiraliEkle(&(ucus->sonraki),bilet);**

**ucus->bosKoltuk--;**

**}**

**printf("\tBilet satisi tamamlandi\n");**

**}**

**}**

**}**

**break;**

**case 4:**

**ucusNo=ucusNoAl();**

**if(hashUcusNo[ucusNo]==NULL)**

**{**

**printf("\t%d numarali ucus yok!\n",ucusNo+1000);**

**}**

**else**

**{**

**ucus=hashUcusNo[ucusNo];**

**onceki=ucus->sonraki;**

**if(onceki!=NULL) //satilan bilet varsa**

**{**

**gecici=onceki->sonraki;**

**while(gecici!=NULL)**

**{**

**onceki=gecici;**

**gecici=gecici->sonraki;**

**free(onceki); //satilan biletler siliniyor**

**}**

**free(gecici);**

**}**

**ucus=dugumCikar(&hashListeBasi[toupper(ucus->kalkisYeri[0])-'A'],ucus);**

**free(ucus); //ucus dugumu siliniyor**

**hashUcusNo[ucusNo]=NULL;**

**printf("\tUcus iptal edildi\n");**

**}**

**break;**

**case 5:**

**ucusNo=ucusNoAl();**

**if(hashUcusNo[ucusNo]==NULL)**

**{**

**printf("\t%d numarali ucus yok!\n",ucusNo+1000);**

**}**

**else**

**{**

**printf("TC kimlik numarasi: ");**

**scanf("%s",TC);**

**if((gecici=tcKimligeGoreAraVeCikar(&(ucus->sonraki),TC))==NULL)**

**printf("\tBu ucus icin bilet satin alinmamistir!\n");**

**else**

**{**

**free(gecici);**

**ucus->bosKoltuk++;**

**printf("\tBilet iptal edildi\n");**

**}**

**}**

**break;**

**case 6:**

**kalkisYerineGoreListele(hashListeBasi);**

**break;**

**case 7:**

**kalkisvarisYerineGoreListele(hashListeBasi);**

**break;**

**case 8:**

**ucusVeBiletListele(hashUcusNo);**

**break;**

**case 9:**

**dolulugaGoreListele(hashListeBasi);**

**break;**

**case 10:**

**biletListele(hashUcusNo);**

**break;**

**case 11:**

**ucusGoruntule(hashListeBasi);**

**break;**

**case 12:**

**printf("\nProgram kapatiliyor");**

**for(i=0;i<3;i++)**

**{**

**beklet(150000000);**

**printf(".");**

**}**

**printf("\n");**

**break;**

**}**

**}while(secim!=12);**

**system("pause");**

**return 0;**

**}**

**void beklet(long int n)**

**{**

**int i;**

**for(i=0;i<n;i++) {}**

**}**

**void baslikYazdir()**

**{**

**printf("\n"**

**"Ucus No Kalkis Yeri Varis Yeri Zaman Kapasite Bos Koltuk Doluluk %%\n"**

**"------- ----------- ---------- ----- -------- ---------- ---------\n");**

**}**

**void yazdir(UCUS ucus)**

**{**

**ucus.kalkisYeri[0]=toupper(ucus.kalkisYeri[0]);**

**ucus.varisYeri[0]=toupper(ucus.varisYeri[0]);**

**printf("%7d %-11s %-10s %02d:%02d %8d %10d %9d\n",**

**ucus.ucusNo+1000,ucus.kalkisYeri,ucus.varisYeri,**

**ucus.zaman/100,ucus.zaman%100,ucus.kapasite,ucus.bosKoltuk,**

**100\*(ucus.kapasite-ucus.bosKoltuk)/ucus.kapasite);**

**}**

**void biletYazdir(BILET \*bas) //kuyrugun ilk dugumunun adresi call-by-value ile aktariliyor**

**{**

**int sayac=0;**

**float toplamFiyat=0;**

**while(bas!=NULL)**

**{**

**printf("%11s %7.2f\n",bas->TC,bas->fiyat);**

**toplamFiyat+=bas->fiyat;**

**bas=bas->sonraki; //bas 1 oteleniyor ama main'de degismiyor**

**sayac++;**

**}**

**printf("\nToplam bilet sayisi: %d\n"**

**"Toplam kazanc: %.2f\n"**

**"-----------------------------\n"**

**"Ortalama bilet fiyati: %.2f\n",sayac,toplamFiyat,(float)toplamFiyat/sayac);**

**}**

**void biletListele(UCUS \*\*hashUcusNo)**

**{**

**int i,sayac=0;**

**char TC[12];**

**UCUS \*ucus;**

**BILET \*gecici;**

**printf("\nTC kimlik no: ");**

**scanf("%s",TC);**

**for(i=0;i<9000;i++)**

**{**

**if((ucus=hashUcusNo[i])==NULL || ucus->sonraki==NULL)**

**continue;**

**else**

**{**

**gecici=ucus->sonraki;**

**while(gecici!=NULL)**

**{**

**if(!strcmp(TC,gecici->TC))**

**{**

**sayac++;**

**if(sayac==1)**

**printf("\n"**

**"Ucus No Kalkis Yeri Varis Yeri Zaman Fiyat\n"**

**"------- ----------- ---------- ----- -------\n");**

**ucus->kalkisYeri[0]=toupper(ucus->kalkisYeri[0]);**

**ucus->varisYeri[0]=toupper(ucus->varisYeri[0]);**

**printf("%7d %-11s %-10s %02d:%02d %7.2f\n",**

**ucus->ucusNo+1000,ucus->kalkisYeri,ucus->varisYeri,**

**ucus->zaman/100,ucus->zaman%100,gecici->fiyat);**

**}**

**gecici=gecici->sonraki;**

**}**

**}**

**}**

**if(sayac==0)**

**printf("\tBu yolcu bilet satin almamistir!\n");**

**}**

**void ucusGoruntule(UCUS \*hash)**

**{**

**int i;**

**UCUS \*gecici;**

**baslikYazdir();**

**for(i=0;i<26;i++)**

**{**

**gecici=hash[i].kalkisYerineGoreSonraki;**

**while(gecici!=&hash[i])**

**{**

**yazdir(\*gecici);**

**gecici=gecici->kalkisYerineGoreSonraki;**

**}**

**}**

**}**

**void kucukHarfeCevir(char \*str)**

**{**

**int i;**

**for(i=0;i<strlen(str);i++)**

**str[i]=tolower(str[i]);**

**}**

**void kalkisYerineGoreListele(UCUS \*hashListeBasi)**

**{**

**char kalkisYeri[MAX];**

**UCUS \*listeBasi,\*simdiki;**

**printf("\nKalkis yeri: ");**

**scanf("%s",kalkisYeri);**

**kucukHarfeCevir(kalkisYeri);**

**listeBasi=&hashListeBasi[toupper(kalkisYeri[0])-'A'];**

**simdiki=listeBasi->kalkisYerineGoreSonraki;**

**if(simdiki==listeBasi)**

**printf("\t%s sehrinden kalkan bir ucus yok!\n",kalkisYeri);**

**else**

**{**

**baslikYazdir();**

**while(simdiki!=listeBasi)**

**{**

**if(!strcmp(simdiki->kalkisYeri,kalkisYeri))**

**yazdir(\*simdiki);**

**simdiki=simdiki->kalkisYerineGoreSonraki;**

**}**

**}**

**}**

**void kalkisvarisYerineGoreListele(UCUS \*hashListeBasi)**

**{**

**char kalkisYeri[MAX],varisYeri[MAX];**

**UCUS \*listeBasi,\*simdiki;**

**printf("\nKalkis yeri: ");**

**scanf("%s",kalkisYeri);**

**kucukHarfeCevir(kalkisYeri);**

**listeBasi=&hashListeBasi[toupper(kalkisYeri[0])-'A'];**

**simdiki=listeBasi->kalkisYerineGoreSonraki;**

**if(simdiki==listeBasi)**

**printf("\t%s sehrinden kalkan bir ucus yok!\n",kalkisYeri);**

**else**

**{**

**printf("Varis yeri: ");**

**scanf("%s",varisYeri);**

**kucukHarfeCevir(varisYeri);**

**baslikYazdir();**

**while(simdiki!=listeBasi)**

**{**

**if(strcmp(simdiki->kalkisYeri,kalkisYeri)==0 && strcmp(simdiki->varisYeri,varisYeri)==0)**

**yazdir(\*simdiki);**

**simdiki=simdiki->kalkisYerineGoreSonraki;**

**}**

**}**

**}**

**void ucusVeBiletListele(UCUS \*\*hashUcusNo)**

**{**

**int ucusNo;**

**UCUS \*ucus;**

**ucusNo=ucusNoAl();**

**if((ucus=hashUcusNo[ucusNo])==NULL)**

**printf("\t%d numarali ucus yok!\n",ucusNo+1000);**

**else**

**{**

**baslikYazdir();**

**yazdir(\*ucus);**

**if(ucus->sonraki==NULL)**

**printf("\n\tBu ucus icin bilet satilmamistir!\n");**

**else**

**{**

**printf("\n"**

**"TC Kimlik No Fiyat\n"**

**"------------ -------\n");**

**biletYazdir(ucus->sonraki);**

**}**

**}**

**}**

**void dolulugaGoreListele(UCUS \*hashListeBasi)**

**{**

**int i;**

**UCUS \*gecici;**

**baslikYazdir();**

**for(i=0;i<26;i++)**

**{**

**if(hashListeBasi[i].kalkisYerineGoreSonraki==&hashListeBasi[i])**

**continue;**

**else**

**{**

**gecici=hashListeBasi[i].kalkisYerineGoreSonraki;**

**while(gecici!=&hashListeBasi[i])**

**{**

**if((float)100\*(gecici->kapasite-gecici->bosKoltuk)/gecici->kapasite<50)**

**yazdir(\*gecici);**

**gecici=gecici->kalkisYerineGoreSonraki;**

**}**

**}**

**}**

**}**

**void kalkisYerineGoreSiraliEkle(UCUS \*listeBasiPtr,UCUS \*yeni)**

**{**

**UCUS \*onceki,\*gecici;**

**onceki=listeBasiPtr;**

**gecici=listeBasiPtr->kalkisYerineGoreSonraki;**

**while(gecici!=listeBasiPtr && strcmp(yeni->kalkisYeri,gecici->kalkisYeri)>0)**

**{**

**onceki=gecici;**

**gecici=gecici->kalkisYerineGoreSonraki;**

**}**

**while(strcmp(yeni->kalkisYeri,gecici->kalkisYeri)==0 && yeni->zaman>gecici->zaman)**

**{**

**onceki=gecici;**

**gecici=gecici->kalkisYerineGoreSonraki;**

**}**

**onceki->kalkisYerineGoreSonraki=yeni;**

**yeni->kalkisYerineGoreSonraki=gecici;**

**}**

**void tcKimligeGoreSiraliEkle(BILET \*\*bas,BILET \*yeni)**

**{**

**BILET \*gecici, \*onceki;**

**if(\*bas==NULL) //kuyruk bossa**

**{**

**yeni->sonraki=NULL;**

**\*bas=yeni;**

**}**

**else**

**{**

**if(strcmp(yeni->TC,(\*bas)->TC)<0) //en basa eklenecekse**

**{**

**yeni->sonraki=\*bas;**

**\*bas=yeni;**

**}**

**else //araya ya da sona eklenecekse**

**{**

**onceki=\*bas;**

**gecici=(\*bas)->sonraki;**

**while((gecici!=NULL) && strcmp(gecici->TC,yeni->TC)<0) //eklenecek uygun yer araniyor**

**{**

**onceki=gecici;**

**gecici=gecici->sonraki;**

**}**

**yeni->sonraki=gecici; //gecici NULL ise en sona, degilse onceki dugumu ile gecici dugumu arasina ekleniyor**

**onceki->sonraki=yeni;**

**}**

**}**

**}**

**int ucusNoAl()**

**{**

**int ucusNo;**

**printf("\n");**

**do{**

**printf("Ucus numarasini giriniz: ");**

**fflush(stdin);**

**scanf("%d",&ucusNo);**

**if(ucusNo<1000)**

**printf("\tUcus numarasi 1000'den kucuk olamaz!\n");**

**else if(ucusNo>9999)**

**printf("\tUcus numarasi 9999'dan buyuk olamaz!\n");**

**}while(ucusNo<1000 || ucusNo>9999);**

**return ucusNo-1000;**

**}**

**int menuSecim()**

**{**

**int secim;**

**beklet(300000000);**

**printf("\n"**

**"\t 1) Yeni bir ucus ekleme\n"**

**"\t 2) Bir ucusun kalkis zamanini guncelleme\n"**

**"\t 3) Bir ucus icin bilet satma\n"**

**"\t 4) Bir ucusu iptal etme\n"**

**"\t 5) Bir ucus icin satilan bir bileti iptal etme\n"**

**"\t 6) Bir yerden kalkan ucus bilgilerini goruntuleme\n"**

**"\t 7) Kalkis ve varis yerine gore ucus bilgilerini goruntuleme\n"**

**"\t 8) Bir ucusun ve o ucus icin satilan bilet bilgilerini goruntuleme\n"**

**"\t 9) Doluluk orani %%50'nin altindaki ucuslari goruntuleme\n"**

**"\t10) Bir yolcunun biletlerini goruntuleme\n"**

**"\t11) Tum ucuslari goruntuleme\n"**

**"\t12) Cikis\n"**

**"\n");**

**do{**

**printf("\tSecim: ");**

**fflush(stdin);**

**scanf("%d",&secim);**

**if(secim<1 || secim>12)**

**printf("\t\tHatali secim!\n");**

**}while(secim<1 || secim>12);**

**//system("cls");**

**return secim;**

**}**

**float biletFiyatHesapla(UCUS ucus)**

**{**

**int i;**

**float fiyat,BKO;**

**BKO=100-((float)100\*(ucus.kapasite-ucus.bosKoltuk)/ucus.kapasite);**

**fiyat=ucus.fiyat;**

**for(i=0;i<10;i++)**

**{**

**if(BKO<=(float)((10-i)\*10.0) && BKO>(float)((10-i-1)\*10.0)) //dongunun her donusunde yeni araliklar hesaplaniyor**

**{**

**fiyat\*=(float)(1.0+i/10.0); //kosulu saglayan katsayi hesaplanip fiyata yansitiliyor**

**break;**

**}**

**}**

**return fiyat;**

**}**

**UCUS \*dugumCikar(UCUS \*listeBasiPtr,UCUS \*silinecek)**

**{**

**UCUS \*gecici,\*onceki;**

**onceki=listeBasiPtr;**

**gecici=listeBasiPtr->kalkisYerineGoreSonraki;**

**while(gecici!=listeBasiPtr && gecici->ucusNo!=silinecek->ucusNo)**

**{**

**onceki=gecici;**

**gecici=gecici->kalkisYerineGoreSonraki;**

**}**

**onceki->kalkisYerineGoreSonraki=gecici->kalkisYerineGoreSonraki;**

**return gecici;**

**}**

**BILET \*tcKimligeGoreAraVeCikar(BILET \*\*bas,char \*TC)**

**{**

**BILET \*gecici,\*onceki;**

**if(\*bas==NULL) //kuyruk bossa**

**{**

**return NULL;**

**}**

**else**

**{**

**if(strcmp((\*bas)->TC,TC)==0) //ilk dugum cikariliyorsa**

**{**

**gecici=\*bas;**

**\*bas=(\*bas)->sonraki;**

**}**

**else**

**{**

**onceki=\*bas;**

**gecici=onceki->sonraki;**

**while(gecici!=NULL && strcmp(gecici->TC,TC)!=0) //cikarilacak dugum araniyor**

**{**

**onceki=gecici;**

**gecici=gecici->sonraki;**

**}**

**if(gecici==NULL) //dugum bulunamadiysa**

**return NULL;**

**else**

**onceki->sonraki=gecici->sonraki;**

**}**

**}**

**return gecici;**

**}**