

ANKARA UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BİLGİSAYAR PROGRAMLAMA II

BAHAR 2023-24

LAB 6-1 Quiz

Prof. Dr. Semra GÜNDÜÇ

Tarih: 29/03/2024

İlk 5 koltuğun birinci sınıf, sonraki koltukların da ekonomi sınıfı olduğu toplamda 10 koltuktan oluşan tek bir uçak için bir havayolu rezervasyon sistemini programlamanız istenmektedir. Programınız uçakta tüm koltuklar dolana kadar çalışacaktır. Uçaktaki oturma düzeni aşağıdaki gibi varsayılacaktır:

Birinci Sınıf	Birinci Sınıf	Birinci Sınıf	Birinci Sınıf	Birinci Sınıf	Ekonomi	Ekonomi	Ekonomi	Ekonomi	Ekonomi
---------------	---------------	---------------	---------------	---------------	---------	---------	---------	---------	---------

Programınız öncelikli olarak kullanıcıya aşağıdaki seçenekleri sunarak başlamalıdır: "Lütfen "birinci sınıf" için 1, "ekonomi" için 2 değerini giriniz:".

Eğer kullanıcı 1 değerini girerse, programınız birinci sınıf bölümünde bir koltuk tahsis etmelidir (1-5 arası koltuklar). Eğer kullanıcı 2 değerini girerse, programınız ekonomi bölümünde bir koltuk tahsis etmelidir (6-10 arası koltuklar). Daha sonra programınız, kişinin koltuk numarasını ve koltuğun birinci sınıf mı yoksa ekonomi sınıfı mı olduğunu belirten bir biniş kartı basmalıdır. Uçağın oturma düzenini temsil etmek için tek boyutlu bir dizi (*array*) kullanınız. Tüm koltukların boş olduğunu belirtmek için dizinin tüm öğelerine başlangıçta 0 değerlerini atayınız. Koltuk atanırken ilgili koltuğun artık dolu olduğunu belirtmek için dizinin ilgili değerini 1'e çeviriniz. Koltuk atamaları her sınıf için kendi içerisinde sıra ile gerçekleştirilecektir.

Programınız önceden atanmış bir koltuğun satışını asla yapmamalıdır. Birinci sınıf bölümü dolduğunda, programınız kişiye ekonomi bölümüne yerleştirilmesinin kabul edilebilir olup olmadığını sormalıdır (veya tam tersi). Kullanıcının cevabı evet ise uygun koltuk atamasını yapınız. Kullanıcının cevabı hayır ise "Sonraki uçuş 3 saat sonra kalkıyor." mesajını ekrana yazdırınız.

**NOT:** Lütfen size verilen input-output dosyalarını inceleyiniz ve oradaki formata sadık kalınız. Programınız Yes-No seçiminde "Y", "y", "N", "n" harflerinin tamamında çalışmalıdır.

**Örnek:**

Lütfen "birinci sınıf" için 1, "ekonomi" için 2 değerini giriniz:

2 → **Kullanıcının girdiği değer**

Ekonomide koltuk atamanız 6.

Lütfen "birinci sınıf" için 1, "ekonomi" için 2 değerini giriniz:

1 → **Kullanıcının girdiği değer**

Birinci sınıfta koltuk atamanız 1.

Lütfen "birinci sınıf" için 1, "ekonomi" için 2 değerini giriniz:

2 → **Kullanıcının girdiği değer**

Ekonomide koltuk atamanız 7.

...  
...  
...

Lütfen "birinci sınıf" için 1, "ekonomi" için 2 değerini giriniz:

**1 → Kullanıcının girdiği değer**

Birinci sınıf bölümü dolmuştur. Ekonomi bölümünde oturmak ister misiniz (Y veya N) ?:

**n → Kullanıcının girdiği değer**

Sonraki uçuş 3 saat sonra kalkıyor.

Lütfen "birinci sınıf" için 1, "ekonomi" için 2 değerini giriniz:

**1 → Kullanıcının girdiği değer**

Birinci sınıf bölümü dolmuştur. Ekonomi bölümünde oturmak ister misiniz (Y veya N) ?:

**y → Kullanıcının girdiği değer**

Ekonomide koltuk atamanız 9.

Lütfen "birinci sınıf" için 1, "ekonomi" için 2 değerini giriniz:

**2 → Kullanıcının girdiği değer**

Ekonomide koltuk atamanız 10.

Bu uçuşa ait tüm koltuklar dolmuştur.

### **Dosya Teslimi:**

1- Dosyanızı öğrenci\_numaranız.c olarak isimlendiriniz → örnek 87456933.c

2- Dosyanızı ekampüs sisteminde ilgili yere yükleyiniz.

### **Derleme İşlemi:**

//normal derleme

gcc öğrenci\_numaranız.c -o çalıştırılabilirDosyaİsminiz

./çalıştırılabilirDosyaİsminiz

//.txt dosyasının input olarak kullanımı

./çalıştırılabilirDosyaİsminiz<input.txt

// çıktının .txt dosyasına yazdırılması

./çalıştırılabilirDosyaİsminiz<input.txt>myoutput.txt

// iki dosyanın karşılaştırılması

diff -w dosya1 dosya2

### **Örnek:**

gcc 87456933.c -o myprogram

./myprogram<input1.txt>myoutput1.txt

diff -w output1.txt myoutput1.txt

**LÜTFEN, SİZE VERİLEN I/O FORMATLARINA DİKKAT EDİNİZ!**