**DİKKAT EDİLECEK HUSUSLAR**

1. Son ödeviniz dönem projesi grup çalışması şeklinde yürütülecektir. Grup üyeleri en az 2, en çok 3 kişi olacaktır.
2. Proje konuları ders kitabının DOSYALAR Konusu -Konu Sonu Çalışma Sorularından seçilecektir. Grubun sorumlu olduğu proje **Soru numarası=Grup-numaranız%55** şeklinde formülize edilmiştir. Bu yöntemle grubunuza ait proje konunuzu belirleyiniz.

*(Grup numarası tek sayı olan gruplar --- (1015+grup numarası)%55*

*(Örneğin ;5.grup için--- 1020%55=30.soru)*

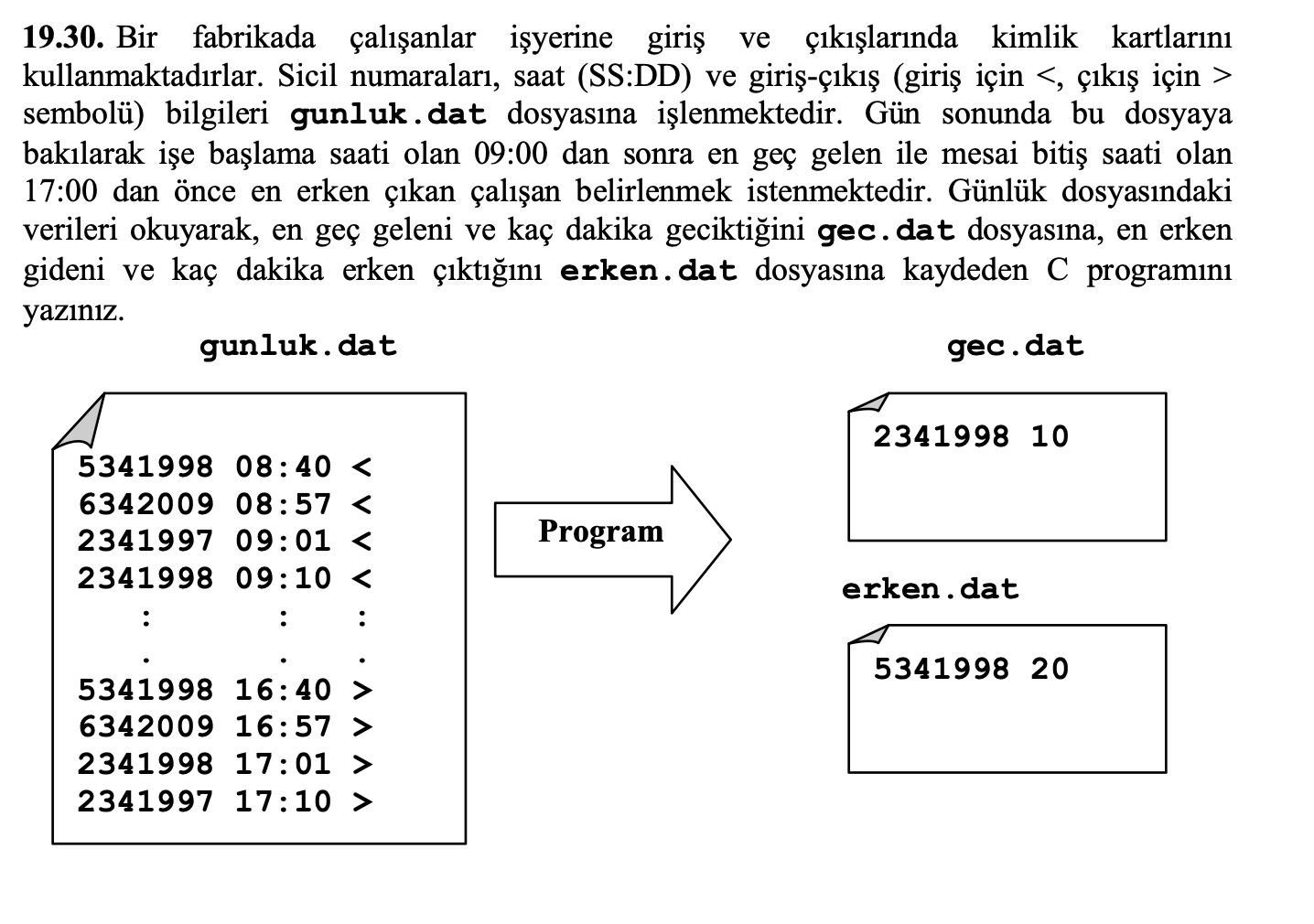
*Grup numarası çift sayı olan gruplar --- (2000+grup numarası)%55*

*(Örneğin ;8.grup için--- 2008%55=28.soru))*

1. Proje konunuzun belirlenmesinden sonra bir toplantı yaparak aranızda görev dağılımı yapınız.
2. Grup üyeleri olarak en az bir toplantı düzenleyerek, bu toplantılardan en az birini raporlayınız. **Kayıt altına aldığınız herhangi bir toplantının raporu 1 sayfayı geçmemelidir.** Proje ekibindeki katılımcılar projedeki görevlerini açıklamalı ve toplantının tarih ve günüde rapora eklenmelidir.
3. Grup projesi tamamlandıktan sonra ilgili ders saatinde proje sunumu için proje ekibinin **tamamı katılmalıdır.** Sunumda, proje ekibi üyelerinden proje sorusunu, akış şemasını ve C kodlarını kısaca açıklaması istenmektedir. Ekibin görev dağılımı proje raporunda belirtilmelidir.
4. Proje raporunuzun içeriği: ***Proje Konusu, Akış Şeması, C Program kodları, Ekran çıktıları*** olmalıdır.
5. Proje dosyalarınızın github ortamına yüklenmiş olması gerekmektedir. Github linki proje raporuna eklenmelidir.
6. Raporunuzun ismi: ***şubeno***-***grupnumaranız-proje*** olmalıdır. Proje raporunuzu .doc yada .docx formatında **Google Classroom’a** yükleme (upload) yapınız.
7. Son teslim tarihinden sonraki teslimler değerlendirilmeyecektir.
8. Son Teslim Tarihi ve saati : 20/12/2024 Cuma saat 23:59

Algoritma ve Programlamaya Giriş Dersi

Dönem Sonu Projesi



Şube No: 2

Grup Numarası : 10

Grup Üyeleri : Yusuf Kuru, Rıza Maden, Emirhan Fırtına

Github: https://github.com/rizamaden/Giris-Cikis-Kontrol

Toplantı Raporu:

**Toplantı Tarihi ve Günü:** 17 Aralık 2024, Salı  
**Toplantı Süresi:** 21:00 - 23:00  
**Toplantı Yeri:** Çevrimiçi (Zoom)  
**Katılımcılar: Emirhan Fırtına, Rıza Maden, Yusuf Kuru**

#### ****Toplantı Amacı:****

Proje kapsamında üstlenilen görevlerin son durumunu değerlendirmek, geri kalan işleri planlamak ve projenin zamanında tamamlanmasını sağlamak.

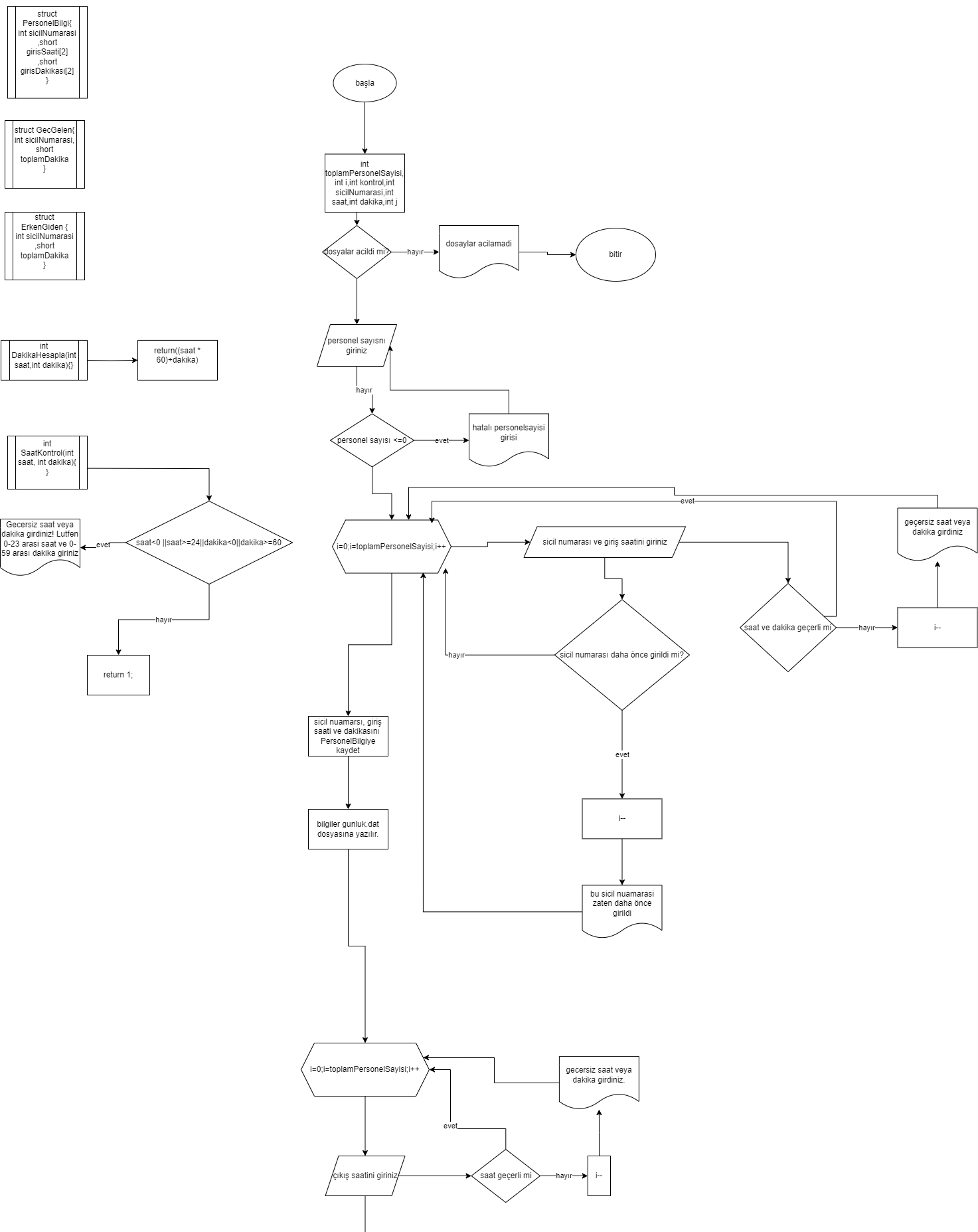
### Görevler:

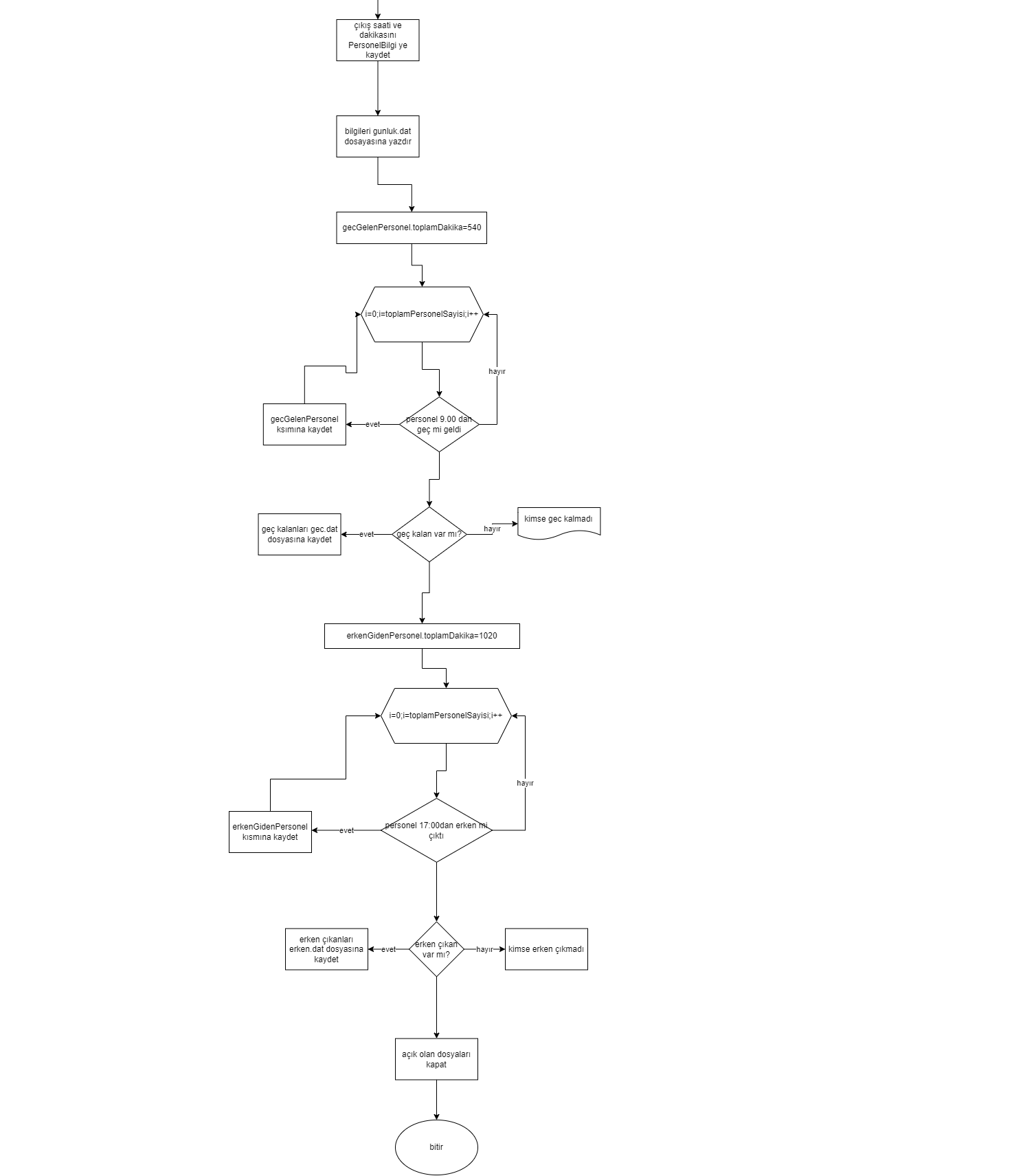
1. **Emirhan Fırtına ve Rıza Maden:**
   * Kod yazımı ve testleri tamamladılar. Kodun çalışmasını sağlayarak proje için gereken tüm işleri bitirdiler.
2. **Yusuf Kuru:**
   * Akış şemasını hazırladı ve ekipten onay aldı. Raporu birleştirme ve teslim etme işlerini planladı.

#### ****Toplantı Notları:****

* Proje ilerleyişinde bir aksaklık bulunmadığı gözlemlendi.
* Kod yazımı ve akış şeması gibi teknik işler başarıyla tamamlandı.
* Raporun son hali 20 Aralık 2024'te teslim edilecek.

Proje Akış Şeması:





Program Kodu:

#include <stdio.h>  
  
struct PersonelBilgi { // İşe gelen personellerin bilgilerini tutan struct  
 int sicilNumarasi; // Sicil numarasi tutan degisken  
 short girisSaati[2]; // Saati tutan degisken 0 numaralı indis giris saati 1 numaralı indis cikis saati  
 short girisDakikasi[2]; // Dakika tutan degisken 0 numaralı indis giris dakikasi 1 numarali indis cikis dakikasi  
};  
struct GecGelen { // Gec gelen personelin bilgilerini tutan struct  
 int sicilNumarasi; // Gec gelen personelin sicil numarasini tutan degisken  
 short toplamDakika; // Gec gelen personelin saat 9:00'dan itibaren kac dakika geciktigini tutan degisken  
};  
struct ErkenGiden { // Erken gelen personelin bilgilerini tutan struct  
 int sicilNumarasi; // Erken gelen personelin sicil numarasini tutan degisken  
 short toplamDakika; // Erken gelen personelin 17:00'dan kac dakika once gittigini tutan degisken  
};  
  
  
  
int dakikaHesapla(int saat, int dakika) { // Bu projede yapilacak karsilastirmalar icin ihtiyacimiz olan dakika verisini bu fonksiyon ile elde ederiz  
 return (saat \* 60) + dakika;  
}  
  
  
  
int saatkontrol(int saat,int dakika) { // Girilen saat ,saat formatina uygun olmazsa 0 eğer uygunsa 1 donduren kontrol fonksiyonu  
 if (saat < 0 || saat >= 24 || dakika < 0 || dakika >= 60) {  
 printf("Gecersiz saat veya dakika girdiniz! Lutfen 0-23 arasi saat ve 0-59 arasi dakika girin.\n");  
 return 0;  
 }  
 return 1;  
}  
  
  
  
int main() {  
 FILE \*gunlukDosya, \*erkenDosya, \*gecDosya; //Dosya isaretcilerinin tanimlanmasi  
  
 gunlukDosya = fopen("gunluk.dat", "w"); //Girilen sicilnolar saatler gunlukDosya ya  
 erkenDosya = fopen("erken.dat", "w"); //Erken gidenin bilgileri erkenDosya ya  
 gecDosya = fopen("gec.dat", "w"); //Gec gelenin bilgileri gecDosya ya yazilacaktir  
  
 if (gunlukDosya == NULL || erkenDosya == NULL || gecDosya == NULL) { //Dosya acilamadiysa kullanıcıya bildiri veren ve programi sonlandiran kosul  
 printf("Dosyalar acilamadi");  
 return 0;  
 }  
  
 int toplamPersonelSayisi; //Kac kisinin ise gelecegini tutan degisken  
  
 printf("Kac tane personel bugun ise gelecek\n");  
  
  
 do { // toplamPersonelSayisi kontrolu  
 scanf("%d", &toplamPersonelSayisi);  
 if (toplamPersonelSayisi<=0) printf("Hatali personelsayisi girisi\n");  
  
 }while (toplamPersonelSayisi <= 0);  
  
  
  
 struct PersonelBilgi personeller[toplamPersonelSayisi]; // Tum personellerin bilgisini tutan yapi  
 struct GecGelen gecGelenPersonel; // Gec kalan personelin bilgsini tutan yapi  
 struct ErkenGiden erkenGidenPersonel; // Erken giden personelin bilgisini tutan yapi  
  
  
 for (int i = 0; i < toplamPersonelSayisi; i++) { // Kullanicidan ilk olarak Sicil No,Giris saatlerini aldiran ve dosyaya yazdiran dongu  
  
 int kontrol=0; // Sicil numarası kontrolu icin degisken  
 int sicilNumarasi, saat, dakika; // Kullanicidan alinan verileri olasi hatalara karsi tutan degisken  
  
 printf("%d. Personel icin Sicil No ve Giris Saati Girin(SS:DD seklinde giriniz):\n", i + 1);  
 scanf("%d %d:%d", &sicilNumarasi, &saat, &dakika);  
  
 for (int j=i-1; j >= 0 ; j--) { // Sicil numaralarini kontrol etmek icin kullanilan dongu  
 if(personeller[j].sicilNumarasi == sicilNumarasi){ // Sicil numaralarindan herhangi birinde eslesme varsa kontrol degiskeni 1 olur  
 printf("Bu sicil numarasi zaten daha once girildi\n");  
 kontrol=1;  
  
 }  
 }  
 if(kontrol==1) { // Kontol degiskeni 1 ise dongumuzde hataya yer vermemek icin  
 i--; // sayacimizi 1 azaltip kalan kısmi atlariz  
 continue;  
 }  
  
 if(saatkontrol(saat,dakika) == 0) { // saatkontrol fonksiyonuna gore saat gecersizse dongumuzde hataya yer vermemek icin  
 i--; // sayacimizi 1 azaltip kalan kısmi atlariz  
 continue;  
 }  
  
  
 // Bir hata ile karsılasılmazsa personel bilgileri yapının icindeki degiskenlere atanir  
 personeller[i].sicilNumarasi = sicilNumarasi;  
 personeller[i].girisSaati[0] = saat;  
 personeller[i].girisDakikasi[0] = dakika;  
  
 // Yapi degiskenlerinin icindeki degerler gunluk dosyasina kaydedilir  
 fprintf(gunlukDosya, "%d %d:%d <\n", personeller[i].sicilNumarasi, personeller[i].girisSaati[0], personeller[i].girisDakikasi[0]);  
 }  
  
 for (int i = 0; i < toplamPersonelSayisi; i++) { // Kullanicidan onceden alinan Sicil no lara gore gore cikis saatlerini aldiran dongu  
 int saat, dakika;  
  
 printf("%d sicil numarali personelin Cikis saatini giriniz\n", personeller[i].sicilNumarasi);  
 scanf("%d:%d", &saat, &dakika);  
  
 if (saatkontrol(saat, dakika) == 0) { // saatkontrol fonksiyonuna gore saat gecersizse dongumuzde hataya yer vermemek icin  
 i--; // sayacimizi 1 azaltip kalan kısmi atlariz  
 continue;;  
 }  
  
 // Bir hata ile karsılasılmazsa personellerin cikis verileri yapi degiskenlerine atanir  
 personeller[i].girisSaati[1] = saat;  
 personeller[i].girisDakikasi[1] = dakika;  
  
 // Yapi degiskenlerinin icindeki degerler gunluk dosyasina kaydedilir  
 fprintf(gunlukDosya, "%d %d:%d >\n", personeller[i].sicilNumarasi, personeller[i].girisSaati[1], personeller[i].girisDakikasi[1]);  
 }  
  
 gecGelenPersonel.toplamDakika = 540; // Giris saati kontrolu icin referans dakika (540/60 = saat 09:00)  
 for (int i = 0; i < toplamPersonelSayisi; i++) { // Personel sayisi kadar dongu donecek  
  
 // Giris saati 09:00'dan sonra olan ve en gec gelen personeli bulmak icin yapilan karsilastirma  
 if (dakikaHesapla(personeller[i].girisSaati[0], personeller[i].girisDakikasi[0]) > gecGelenPersonel.toplamDakika && dakikaHesapla(personeller[i].girisSaati[0], personeller[i].girisDakikasi[0]) > 540) {  
 // Kosul saglanirsa degerler gecGelenPersonel yapisinin degiskenlerine atanir  
 gecGelenPersonel.sicilNumarasi = personeller[i].sicilNumarasi;  
 gecGelenPersonel.toplamDakika = dakikaHesapla(personeller[i].girisSaati[0], personeller[i].girisDakikasi[0]);  
 }  
 }  
  
 if (gecGelenPersonel.toplamDakika == 540) { // Kimsenin gec kalip kalmadigini kontrol eden ifade  
 fprintf(gecDosya,"Kimse Gec Kalmadi\n"); // Kimse gec kalmadiysa yazilacak ifade  
 }  
 else fprintf(gecDosya, "%d %d", gecGelenPersonel.sicilNumarasi, gecGelenPersonel.toplamDakika - 540); // Deger 540'dan fazla ise yazilacak ifade  
  
  
 erkenGidenPersonel.toplamDakika = 1020; // Cikis saati kontrolu icin referans dakika (1020/60 = saat 17:00)  
 for (int i = 0; i < toplamPersonelSayisi; i++) { // Personel sayisi kadar dongu donecek  
  
 // Cikis saati 17:00'dan once olan ve en erken giden personeli bulmak icin yapilan karsilastirma  
 if (dakikaHesapla(personeller[i].girisSaati[1], personeller[i].girisDakikasi[1]) < erkenGidenPersonel.toplamDakika && dakikaHesapla(personeller[i].girisSaati[1], personeller[i].girisDakikasi[1]) < 1020) {  
 // Kosul saglanirsa degerler erkenGidenPersonel yapisinin degiskenlerine atanir  
 erkenGidenPersonel.sicilNumarasi = personeller[i].sicilNumarasi;  
 erkenGidenPersonel.toplamDakika = dakikaHesapla(personeller[i].girisSaati[1], personeller[i].girisDakikasi[1]);  
 }  
 }  
  
 if (erkenGidenPersonel.toplamDakika == 1020) { // Kimsenin gec kalip kalmadigini kontrol eden ifade  
 fprintf(erkenDosya,"Kimse Erken Cikmadi\n"); // Kimse gec kalmadiysa yazilacak ifade  
 }  
 else fprintf(erkenDosya, "%d %d", erkenGidenPersonel.sicilNumarasi, 1020-erkenGidenPersonel.toplamDakika); // Deger 1020'den az ise yazilacak ifade  
  
 //Dosyalari kapatiyoruz  
 fclose(gunlukDosya);  
 fclose(erkenDosya);  
 fclose(gecDosya);  
  
 return 0;  
}

Ekran Çıktıları:

