PROGRAMLAMA DİLLERİ

DÖNEM PROJESİ

Hazırlayan

Sıddık Said Aydoğan, 11011052

Arş. Grv. Dr.M. Amaç Güvensan

Programlama Dilleri

İÇİNDEKİLER

BÖLÜM 1. YÖNTEM

1. 1 Problem ve Çözümünün İfadesi………………………………

BÖLÜM 2. PROGRAM……………………………………………

BÖLÜM 3. UYGULAMA

**BÖLÜM 1**

**YÖNTEM**

* 1. **Problem ve Çözümünün İfadesi**

***Verilen problem:*** Bir şirket için çalışanların çalışma zaman çizelgelerinden yararlanarak;

* Aylık çalışma saati.
* Çalışılmayan gün sayısı.
* Şubeyi açma/kapatma sayısı.
* Haftasonu çalışılmayan gün sayısı.
* Bir çalışanın şubeyi kapatıp ertesi gün açma sayısı.
* Aynı saatte çalışmaya başlayıp ve çalışmayı bitirinlerin bulunması.
* Bir çalışanın şubeyi açma/ çalışılmayan gün / şubeyi kapatma eylemini tekrarlama sayısı.

Yukardaki bilgiler yeni bir csv dosyasında istenmiştir. Çalışanların çizelgeleri haftalık olarak csv dosyası şeklinde programa verilecektir.

***Çözüm:***

**Kullanıcıdan Alınan Veriler:**

* Şube açılış ve kapanış saatleri.
* Haftalık zaman çizelgesinin yer aldığı dosya ismi.

**Çıktılar:**

* İstenilen hesaplamalar yapılır ve sonuç Rapor.csv olarak kaydedilir.

Csv dosya formatına göre noktalı virgül ( ; ) ile bilgiler birbirinden ayrılmaktadır. Bu bilgileri strtok fonksiyonu kullanarak her ; arasındaki karakter katarlarını almayı sağlarız. Alınan bilgiler şubeyi açma ve kapatma saatleridir. Her çalışanın bilgisi yapıya(CALISAN isimli structure) yerleştirilir. Çalışanların bulunduğu dizi sayesinde fonksiyonlara gönderilerek hesaplamalar yapılması sağlanır.

**Kullanılan Fonksiyon Listesi:**

void yapiyaYerlestir(char\*dizi[20],CALISAN \* clisan,int hafta);

void saatHesapla(CALISAN \*c);

void acmaKapamaHesapla(CALISAN \*c,float subeA,float subeK);

void ertesiGunAcmaSayisi(CALISAN \*c,float subeA,float subeK);

void haftasonuCalismamaSayisi(CALISAN \*c);

void acmisCalismadanKapatmis(CALISAN \*c,float subeA,float subeK);

void isArkadaslariMatrisiniDoldur(CALISAN \*calisanlar,int calisanSayisi,int \*\* isArkadaslari);

void dosyayiOlustur(CALISAN \*calisanlar,int cSayisi,int \*\* isArkadaslari);

**Kullanılan CALISAN Yapısı:**

typedef struct

{

char CalisanId[5];

char \*CalisanAdi;

char CalismaTipi[10];

float acilisSaatleri[28];

float kapanisSaatleri[28];

float aylikCalismaSaati;

int acmaSayisi;

int kapatmaSayisi;

int kapatipHemenAcmaSayisi;

int hsonuCalismamaSayisi;

int calisilmayanGunSayisi;

int acmisCalismadanKapatmis; // Acilis/Calisilmayan Gun/Kapanis

} CALISAN;

**BÖLÜM 2**

**PROGRAM**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

typedef struct

{

char CalisanId[5];

char \*CalisanAdi;

char CalismaTipi[10];

float acilisSaatleri[28];

float kapanisSaatleri[28];

float aylikCalismaSaati;

int acmaSayisi;

int kapatmaSayisi;

int kapatipHemenAcmaSayisi;

int hsonuCalismamaSayisi;

int calisilmayanGunSayisi;

int acmisCalismadanKapatmis; // Acilis/Calisilmayan Gun/Kapanis

} CALISAN;

void yapiyaYerlestir(char\*dizi[20],CALISAN \* clisan,int hafta); void saatHesapla(CALISAN \*c);

void acmaKapamaHesapla(CALISAN \*c,float subeA,float subeK); void ertesiGunAcmaSayisi(CALISAN \*c,float subeA,float subeK); void haftasonuCalismamaSayisi(CALISAN \*c);

void acmisCalismadanKapatmis(CALISAN \*c,float subeA,float subeK); void isArkadaslariMatrisiniDoldur(CALISAN \*calisanlar,int calisanSayisi,int \*\* isArkadaslari);

void dosyayiOlustur(CALISAN \*calisanlar,int cSayisi,int \*\* isArkadaslari);

int main()

{

char dosyaAdi[20];

int i,j,cSayisi=0;

FILE \*fp;

char buffer[200];

char \*ayristirilanDegerler[20];

CALISAN \* calisanlar,\* tmp;

float acilisD,kapanisD,subeA,subeK;

puts("Subenin acilis saatini veriniz(or-> 05:00)\n");

scanf("%f:%f",&subeA,&acilisD);

acilisD=acilisD/60;

subeA+=acilisD;

puts("Subenin kapanis saatini veriniz(or-> 02:00)\n");

scanf("%f:%f",&subeK,&kapanisD);

kapanisD=kapanisD/60;

subeK+=kapanisD;

for (i=0; i<4; i++ )

{

printf("%d. Haftanin dosya adini giriniz.",i+1);

scanf("%s",dosyaAdi);

fp=fopen(dosyaAdi,"r");

if(fp==NULL)

{

puts("Dosya acilamadi");

exit(EXIT\_FAILURE);

}

fgets(buffer,200,fp); // ilk 2 satir atlaniyor.

fgets(buffer,200,fp);

j=0;

while (!feof(fp))

{

fgets(buffer,200,fp);

ayristirilanDegerler[j] = strtok(buffer,";"); //strtok string'i tokenlere ayirmak icin kullaniliyor.

if(strcmp(ayristirilanDegerler[0],"\n")!=0)

{

while(ayristirilanDegerler[j]!=NULL)

{

j++;

ayristirilanDegerler[j]=strtok(NULL,";"); //strtok a ozgu kullanim bicimi. Ayristirmaya devam etmesini sagliyor.

}

if(i==0)

{

if(cSayisi==0)

calisanlar = (CALISAN \*)calloc(1,sizeof(CALISAN));

else

{

tmp = calisanlar;

calisanlar = (CALISAN \*)realloc(tmp,(cSayisi+1) \* sizeof(CALISAN));

}

}

yapiyaYerlestir(ayristirilanDegerler,&calisanlar[cSayisi],i+1);

cSayisi++;

j=0;

fgets(buffer,200,fp); // bilgilerin altindaki 2 satir atlaniyor.

fgets(buffer,200,fp);

}

}

if(i!=3) // Son kiside calisan sayisini tekrar sifirlamamak icin.

cSayisi=0; // cSayisi bir indis gibi kullanildigi icin her turda sifirlaniyor.

fclose(fp);

}

puts("Dosya okuma islemleri bitti.\n Calisanlarla ilgili bilgiler hesaplaniyor.");

//DOSYA OKUMALAR BITTI

// /HESAPLAMA ISLEMLERI /

for(i=0; i<cSayisi; i++)

{

saatHesapla(&calisanlar[i]);

acmaKapamaHesapla(&calisanlar[i],subeA,subeK);

ertesiGunAcmaSayisi(&calisanlar[i],subeA,subeK);

haftasonuCalismamaSayisi(&calisanlar[i]);

acmisCalismadanKapatmis(&calisanlar[i],subeA,subeK);

}

int \*\* isArkadaslari=(int\*\*)calloc(cSayisi,sizeof(int));

for(i=0; i<cSayisi; i++)

{

isArkadaslari[i]=(int\*)calloc(cSayisi,sizeof(int));

}

isArkadaslariMatrisiniDoldur(calisanlar,cSayisi,isArkadaslari);

dosyayiOlustur(calisanlar,cSayisi,isArkadaslari);

puts("Hesaplamalar tamamlandi.\n Rapor.csv dosyasi olusturuldu.");

return 0;

}

//CSV dosyasi olusturma.

void dosyayiOlustur(CALISAN \*calisanlar,int cSayisi,int \*\* isArkadaslari)

{

FILE \*fp;

fp=fopen("Rapor.csv","w");

if(fp==NULL)

{

puts("Dosya acilamadi");

exit(EXIT\_FAILURE);

}

fprintf(fp,"Calisan ID;Calisan Adi;Calisma Tipi;Aylik Calistigi Saat;Sube Acma Sayisi;Sube Kapatma Sayisi;Kapatip Ertesi Gun Acma Sayisi;Haftasonu Calismama Sayisi;Calisilmayan Gun Sayisi;Acilis/CalisilmayanGun/Kapanis Sayisi;\n");

int i,j;

for(i=0; i<cSayisi; i++)

{ fprintf(fp,"%s;%s;%s;%.1f;%d;%d;%d;%d;%d;%d;\n",calisanlar[i].CalisanId,calisanlar[i].CalisanAdi,calisanlar[i].CalismaTipi,calisanlar[i].aylikCalismaSaati, calisanlar[i].acmaSayisi,calisanlar[i].kapatmaSayisi,calisanlar[i].kapatipHemenAcmaSayisi,calisanlar[i].hsonuCalismamaSayisi,calisanlar[i].calisilmayanGunSayisi,calisanlar[i].acmisCalismadanKapatmis);

}

for(i=0; i<cSayisi; i++)

{

for(j=0; j<cSayisi; j++)

{

if(isArkadaslari[i][j]!=0)

fprintf(fp,"%s kisisi %s ile %d kez beraber calisti.;\n",calisanlar[i].CalisanAdi,calisanlar[j].CalisanAdi,isArkadaslari[i][j]);

}

}

fclose(fp);

}

//String dizisinin 20 elemanli oldugu biliniyor.

//Saatler burda ayristilicak.

void yapiyaYerlestir(char \* dizi[20],CALISAN \* clisan,int hafta)

{

int i,j;

float s,d;

if(hafta==1)

{

i = strlen(dizi[1]);

clisan->CalisanAdi = (char\*)malloc(i+1);

strcpy(clisan->CalisanId,dizi[0]);

strcpy(clisan->CalisanAdi,dizi[1]);

strcpy(clisan->CalismaTipi,dizi[2]); }

j=3;

for(i=hafta\*7-7; i<hafta\*7; i++)

{

sscanf(dizi[j],"%f:%f",&s,&d);

d=d/60;

clisan->acilisSaatleri[i]=s+d;

j++;

sscanf(dizi[j],"%f:%f",&s,&d);

d=d/60;

clisan->kapanisSaatleri[i]=s+d;

j++;

}

}

// Aylik Calisma Saati - Calisilmayan Gun Sayisi Hesaplaniyor.

void saatHesapla(CALISAN \*c)

{

int i;

float saat,acilis,kapanis;

c->aylikCalismaSaati=0;

c->calisilmayanGunSayisi=0;

for(i=0; i<28; i++)

{

kapanis=c->kapanisSaatleri[i];

acilis=c->acilisSaatleri[i];

if(acilis!=0)

{

if(kapanis<acilis)

{

kapanis+=24;

}

saat=kapanis-acilis;

c->aylikCalismaSaati+=saat;

} else c->calisilmayanGunSayisi++;

}

}

// Acma - Kapatma Hesaplaniyor.

void acmaKapamaHesapla(CALISAN \*c,float subeA,float subeK)

{

c->kapatmaSayisi=0;

c->acmaSayisi=0;

int i;

float acilis,kapanis;

for(i=0; i<28; i++)

{

kapanis=c->kapanisSaatleri[i];

acilis=c->acilisSaatleri[i];

if(kapanis==subeK)

c->kapatmaSayisi++;

if(acilis==subeA)

c->acmaSayisi++;

}

}

// Kapatip Ertesi Gun Acma Hesaplaniyor.

void ertesiGunAcmaSayisi(CALISAN \*c,float subeA,float subeK)

{

int i;

c->kapatipHemenAcmaSayisi=0;

for(i=0; i<28; i++)

{

if(i+2<28)

{

if(c->kapanisSaatleri[i]==subeK)

{

if(c->acilisSaatleri[i+1]==subeA)

{

c->kapatipHemenAcmaSayisi++;

i+=2;

}

}

}

}

}

// Haftasonu calisilmayan gun sayisi

void haftasonuCalismamaSayisi(CALISAN \*c)

{

int i;

c->hsonuCalismamaSayisi=0;

// Ilk haftasonu 6. gun.

for(i=5; i<28; i+=7)

{

// c.tesi durumu

if(c->acilisSaatleri[i]==0)

c->hsonuCalismamaSayisi++;

// pazar durumu

if(c->acilisSaatleri[i+1]==0)

c->hsonuCalismamaSayisi++;

}

}

// Acilis/Calisilmayan Gun/Kapanis i tekrarlama sayisinin hesaplanmasi

void acmisCalismadanKapatmis(CALISAN \*c,float subeA,float subeK)

{

int i;

float acilis;

c->acmisCalismadanKapatmis=0;

for(i=0; i<28; i++)

{

acilis=c->acilisSaatleri[i];

if(acilis==subeA)

{

if(i+2 < 28)

{

if(c->acilisSaatleri[i+1]==0)

{

if(c->kapanisSaatleri[i+2]==subeK)

{

c->acmisCalismadanKapatmis++;

i+=2;

}

}

}

}

}

}

// Ayni anda ise baslayanlar ve bitirenlerin bulunmasi.

void isArkadaslariMatrisiniDoldur(CALISAN \*calisanlar,int calisanSayisi,int \*\* isArkadaslari)

{

int i,j,k,gun=0;

for(i=0; i<calisanSayisi; i++)

{

for (j=0; j<calisanSayisi; j++)

{

if(i!=j) // Aynı kisi icin kontrol edilmemesi icin.

{

for(k=0; k<28; k++)

{

if(calisanlar[i].acilisSaatleri[k]!=0 && calisanlar[j].acilisSaatleri[k]!=0) // Ayni anda calismama dahil edilmemesi icin.

{

if(calisanlar[i].acilisSaatleri[k]==calisanlar[j].acilisSaatleri[k] && calisanlar[i].kapanisSaatleri[k]==calisanlar[j].kapanisSaatleri[k])

gun++;

}

}

}

isArkadaslari[i][j]=gun;

gun=0;

}

}

}

**BÖLÜM 3**

**UYGULAMA**

