**1.ŞUBE-13.GRUP-DOSYALAR(19.38)**

**Proje Konusu:** Bu proje ailelerin ürün beğenilerini içeren bir matris verisi üzerinde işlem yaparak, en çok beğenilen ürünü ve en az beğenen aileyi bulmayı amaçlamaktadır. Program, kullanıcıdan alınan dosya adlarıyla veri dosyasını okuyarak, ürünler ve ailelerin beğeni puanları üzerinden toplamları hesaplar. Son olarak, bu verilerle en çok beğenilen ürün ve en az beğenen aileyi tespit eder ve sonuçları yeni bir dosyaya kaydeder.

**Hazırlayan Kişiler:**

Muhammed Ali ATİK 21381616601

Mohammad Musawer MOSSINI 23181616412

**Görev Dağılımı:**

Muhammed Ali ATİK:Akış diyagramı,enCokBegenilenUrun,enAzBegenenAile fonksiyonları

Mohammad Musawer MOSSINI:Main fonksiyonu,akış diyagramı

**Grup Toplantısı 1:**

13/12/2024 tarihi Cuma gününde ekip arkadaşım Musawer ile yapacağımız proje hakkında kafamızda genel bir fikir oluşturmayı ve projeyi tam olarak anlayarak yapmayı amaçladık.Genel bir fikir oluşturduktan sonra projeyi yapmaya başlayarak görev dağılımını yaptık.İlk olarak projede kullanacağımız için çok boyutlu matris oluşturduk.Dosya işlemlerini,karar yapılarını ve döngüleri kodlarımızda kullandık.Taslak şeklinde yazdığımız kodları GitHub’a yükleyerek bu kodu geliştirmek için bazı değişiklikler yaptık.

**Grup Toplantısı 2:**

15/12/2024 tarihi Pazar günü GitHuba yüklenen kodlarda bazı değişiklikler yaptıktan sonra projeyi son haline getirmeyi başardık.Bu tarihte proje için gerekli olan raporu hazırladık ve Github’a yükledik.Yukarıda belirtilen görev dağılımına göre kodlarımızı kendi aramızda paylaştık ve sunum için hazırlandık.

**KODLAR**

**enCokBegenilenUrun Fonksiyonu:**

/\* En çok beğenilen ürünü bulan fonksiyon.

Ürünlerin toplam puanlarını alıyor ve en yüksek olanını buluyor.\*/

int enCokBegenilenUrun(int toplam\_urun[]) {

// Başlangıçta en çok beğenilen ürünü sıfır olarak varsayıyoruz.

int en\_cok\_begenilen\_urun = 0;

// Her ürünü karşılaştırıyoruz. Eğer bir ürün daha fazla puan aldıysa, onu en çok beğenilen olarak güncelliyoruz.

for (int i = 1; i < satir; i++) {

if (toplam\_urun[i] > toplam\_urun[en\_cok\_begenilen\_urun]) {

en\_cok\_begenilen\_urun = i;

}

}

return en\_cok\_begenilen\_urun;

}

**enAzBegenenAile Fonksiyonu:**

/\* En az beğenen aileyi bulan fonksiyon.

Ailelerin toplam puanlarını alıyor ve en düşük puanı bulan aileyi döndürüyor.\*/

int enAzBegenenAile(int toplam\_aile[]) {

// Başlangıçta en az beğenen aileyi sıfır olarak varsayıyoruz.

int en\_az\_begenen\_aile = 0;

// Her aileyi kontrol ediyoruz. Eğer bir aile daha az puan vermişse, onu en az beğenen olarak güncelliyoruz.

for (int j = 1; j < sutun; j++) {

if (toplam\_aile[j] < toplam\_aile[en\_az\_begenen\_aile]) {

en\_az\_begenen\_aile = j;

}

}

return en\_az\_begenen\_aile;

}

**Main:**

#include <stdio.h>

/\* Sabitlerle satir ve sutun sayisini belirliyoruz.

Bu sayilar matrisi olustururken kullaniliyor.\*/

#define satir 8

#define sutun 10

int main() {

// Ornek veri dosyasini olusturuyoruz. Bu dosya, "puanlar.txt" adi ile yaziliyor.

FILE \*dosya = fopen("puanlar.txt", "w");

// Dosya acma islemi basarisiz olursa bir hata mesaji veriyoruz.

if (dosya == NULL) {

printf("Hata: Dosya olusturulamadi!\n");

return 1;

} else {

// Ornek puanlari dosyaya yaziyoruz. Bu veriler matrise yuklenip islenecek.

fprintf(dosya, "0 1 2 1 2 2 1 0 1 0\n"

"2 3 3 0 3 2 3 3 4 1\n"

"1 3 4 1 4 2 1 4 2 3\n"

"3 4 2 3 0 3 4 2 4 2\n"

"4 1 3 2 2 1 3 0 1 0\n"

"4 4 3 2 1 4 4 1 4 1\n"

"1 3 1 3 2 4 1 4 3 4\n"

"2 2 2 3 4 2 3 4 3 3\n");

fclose(dosya); // Islem bittikten sonra dosyayi kapatiyoruz.

}

// Kullanicidan giris ve cikis dosyalarinin adini aliyoruz.

char girdi\_dosyasi[50], cikti\_dosyasi[50];

printf("Puanlarin tutuldugu dosyanin adini giriniz: ");

scanf("%49s", girdi\_dosyasi);

printf("Sonuclar hangi dosyaya kaydedilsin: ");

scanf("%49s", cikti\_dosyasi);

// Puanlari tutacak matrisi tanimliyoruz.

int matris[satir][sutun];

// Giris dosyasini aciyoruz ve puanlari okuyoruz.

FILE \*girdi = fopen(girdi\_dosyasi, "r");

// Dosya acilamazsa hata mesaji gosteriyoruz.

if (girdi == NULL) {

printf("Hata: %s dosyasi acilamadi!\n", girdi\_dosyasi);

return 1;

} else {

// Matrisi dosyadan okuyarak dolduruyoruz.

for (int i = 0; i < satir; i++) {

for (int j = 0; j < sutun; j++) {

fscanf(girdi, "%d", &matris[i][j]);

}

}

fclose(girdi); // Okuma islemi bitince dosyayi kapatiyoruz.

}

// Urunlerin toplam puanlarini hesaplamak icin bir dizi tanimliyoruz.

int toplam\_urun[satir] = {0};

// Ailelerin toplam puanlarini hesaplamak icin baska bir dizi tanimliyoruz.

int toplam\_aile[sutun] = {0};

// Matris uzerinden gecerek urun ve aile toplamlarini hesapliyoruz.

for (int i = 0; i < satir; i++) {

for (int j = 0; j < sutun; j++) {

toplam\_urun[i] += matris[i][j]; // Urun icin puan ekliyoruz.

toplam\_aile[j] += matris[i][j]; // Aile icin puan ekliyoruz.

}

}

// En cok begenilen urunu ve en az begenen aileyi hesaplamak icin fonksiyonlarimizi cagiriyoruz.

int en\_cok\_begenilen\_urun = enCokBegenilenUrun(toplam\_urun);

int en\_az\_begenen\_aile = enAzBegenenAile(toplam\_aile);

// Cikis dosyasini olusturuyoruz ve sonuclari yaziyoruz.

FILE \*cikti = fopen(cikti\_dosyasi, "w");

// Cikis dosyasi acilamazsa hata mesaji gosteriyoruz.

if (cikti == NULL) {

printf("Hata: %s dosyasi olusturulamadi!\n", cikti\_dosyasi);

return 1;

} else {

/\* Sonuclari dosyaya yaziyoruz.

Urun ve aile numaralarini 1 artirarak insanlara uygun hale getiriyoruz (0'dan baslamasin).\*/

fprintf(cikti, "En cok begenilen urun no: %d\n", en\_cok\_begenilen\_urun + 1);

fprintf(cikti, "En az begenen aile no: %d\n", en\_az\_begenen\_aile + 1);

fclose(cikti); // Islem bitince dosyayi kapatiyoruz.

printf("%s dosyasi basariyla olusturuldu.\n", cikti\_dosyasi);

}

    return 0;

}

**AKIŞ DİYAGRAMI**diyagram, taslak, metin, teknik çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, diyagram, plan, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, diyagram, ekran görüntüsü, plan içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, diyagram, çizgi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

**EKRAN ÇIKTILARImetin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**GitHub:** https://github.com/MUSAWER-MOSSINI/c-odevi