**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 satır ve sütun sayısını belirliyoruz.

Bu sayılar matrisi oluştururken kullanılıyor.\*/

#define satir 8

#define sutun 10

int main() {

// Örnek veri dosyasını oluşturuyoruz. Bu dosya, "puanlar.txt" adıyla yazılıyor.

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

// Dosya açma işlemi başarısız olursa bir hata mesajı veriyoruz.

if (dosya == NULL) {

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

return 1;

} else {

// Örnek puanları dosyaya yazıyoruz. Bu veriler matrise yüklenip işlenecek.

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); // İşlem bittikten sonra dosyayı kapatıyoruz.

}

// Kullanıcıdan giriş ve çıkış dosyalarının adını alıyoruz.

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

printf("Puanların tutulduğu dosyanın adını giriniz: ");

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

printf("Sonuçlar hangi dosyaya kaydedilsin: ");

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

// Puanları tutacak matrisi tanımlıyoruz.

int matris[satir][sutun];

// Girdi dosyasını açıyoruz ve puanları okuyoruz.

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

// Dosya açılamazsa hata mesajı gösteriyoruz.

if (girdi == NULL) {

printf("Hata: %s dosyası açılamadı!\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 işlemi bitince dosyayı kapatıyoruz.

}

// Ürünlerin toplam puanlarını hesaplamak için bir dizi tanımlıyoruz.

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

// Ailelerin toplam puanlarını hesaplamak için başka bir dizi tanımlıyoruz.

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

// Matris üzerinden geçerek ürün ve aile toplamlarını hesaplıyoruz.

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

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

toplam\_urun[i] += matris[i][j]; // Ürün için puan ekliyoruz.

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

}

}

// En çok beğenilen ürünü ve en az beğenen aileyi hesaplamak için fonksiyonlarımızı çağırıyoruz.

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

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

// Çıkış dosyasını oluşturuyoruz ve sonuçları yazıyoruz.

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

// Çıkış dosyası açılamazsa hata mesajı gösteriyoruz.

if (cikti == NULL) {

printf("Hata: %s dosyası oluşturulamadı!\n", cikti\_dosyasi);

return 1;

} else {

/\* Sonuçları dosyaya yazıyoruz.

Ürün ve aile numaralarını 1 artırarak insanlara uygun hale getiriyoruz (0'dan başlamasın).\*/

fprintf(cikti, "En çok beğenilen ürün no: %d\n", en\_cok\_begenilen\_urun + 1);

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

fclose(cikti); // İşlem bitince dosyayı kapatıyoruz.

printf("%s dosyası başarıyla oluşturuldu.\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 oluşturuldu

metin, diyagram, ekran görüntüsü, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, diyagram, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma 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