



**BLM3021**

**ALGORİTMA ANALİZİ**

**DÖNEM PROJESİ**

**Kitap Öneri Sistemi**

## **Problem:**

Günümüzde Youtube, Netflix, Amazon, Pinterest gibi internet ortamında milyonlarca kullanıcısı olan pek çok firma makine öğrenmesi tabanlı tavsiye sistemleri ile kullanıcılara kişiselleştirilmiş öneriler sunmaktadır. Bu çalışmada işbirlikçi filtre (collaborative filtering) yöntemi ile bir kişinin önceki seçimlerine bakarak yeni kitap öneren bir sistem tasarlanacak ve gerçekleştirilecektir.

## **ÇÖZÜM**

### **Fonksiyonlar ve yaptıkları işlemler:**

**struct kullanıcılar:** BUTUN USERLARI TUTACAK STRUCTIMIZ

**char okur[30]=USER ISMI**

**int okunanKitaplar[8]=OKUNAN KITAPLARA SIRASIYLA VERILEN  
PUANLAR.ORN:0,1,5,9,0,.....(8 KITAP ICIN)**

**struct kullanıcılar \*next=LINKLI LISTE STRUCTI OLDUGU ICIN AYNI TIPTE NEXTI  
TANIMLANIR.**

**float similarityValue=SADECE U'LU KULLANICILARIN SAHIP OLACAGI,KULLANICININ  
GIRDIGI OKUR ISMINE GORE U'LU KULLANICILARDA OLUSACAK PEARSON(SIM)  
DEGERI.**

**void kullanıcıOlustur(struct kullanıcılar \*tempKullanici) Fonksiyonu :**

Parametre olarak kullanıcılar structini alır ve linkli listenin başına ekleme yapar. Kullanıcı tarafından alınan değer sadece okuyucu adıdır. Geri kalan değerler programın işleyişinde oluşturulur; tüm kitaplara verdiği puanlar vs. Bu şekilde main fonksiyonunda oluşan geçici struct bu fonksiyona gönderilir ve kullanıcı linkli listeye eklenir.

**struct kullanıcılar \*getOkuyucu(char okuyucu[30]) Fonksiyonu:**

Parametre olarak kullanıcı tarafından girilen okuyucu ismini alır ve bu okuyucu ismine sahip bir linkli liste node'u var ise bu node'u döndürür. Fonksiyonun içerisinde geçici bir linkli liste oluşturulup baştan başlatılarak iterasyon yapılır. Her iterasyon esnasında mevcut indisdeki node'un okur adındaki değişkeni, parametre olarak gönderilen değişkenle karşılaştırılır. Uyduğu takdirde istenen node bulunmuş demektir ve return ettirilir.

**float sim(struct kullanıcılar \*a, struct kullanıcılar \*b) Fonksiyonu :**

Parametre olarak a ve b okurlarını, kullanıcılar ismindeki struct yapısında alır. Bu fonksiyonda yapılan işlemler iki kullanıcı arasındaki pearson benzerliği değerinin bulunmasıdır. a olarak gönderilen parametre kullanıcı tarafından girilen NU'lu bir okur adıdır ve b olarak gelen kullanıcı ise linkli liste içindeki bütün U'lu kullanıcılardır.

ra, rb, rap ve rbp değerleri elde edilerek istenen formül kılavuzunda similarity değerleri oluşturulur ve bu değer float şeklinde döndürülür. Döndürüldükten sonra U'lu kullanıcıların node'unda bulunan similarityValue değerine yazılır.

**void sortList() Fonksiyonu :**

Parametre almadan tanımlı linkli listeyi büyükten küçüğe sıralar.

### **void pred(struct kullanicilar \*a,char books[8][50],int N) Fonksiyonu :**

Parametre olarak kullanıcının girmiş olduğu NU'lu bir okur olan a değeri,tüm kitapların bulunduğu string dizisi(char matrisi) ve k değerinin sayısal hali olan N değerini alır.Belirtilen formüldeki gibi ra,rb ve rbp değerleri ile N adet iterasyon ile b değerleri iterasyon esnasında baş harfleri U'lu nodelara gelinmesi kaydıyla sırasıyla işleme sokulur ve pred value değeri elde edilir.a kullanıcısının okumadığı kitapların isimleri ile birlikte pred valueleri ekrana bastırılır.Bu esnada en büyük pred value değerine sahip olan kitap kullanıcıya önerilir.

### **void printUsers(int k) Fonksiyonu :**

Parametre olarak K değerini alır.Fonksiyonun işlevi ise programın akışı dolayısı ile similarity değerleri bulunan nodelardan ve sıralanmış linkli listeden sonra,bu linkli listeyi k adım ekrana bastırır ve kullanıcı k adet benzer U'lu okurları görür.Linkli liste iterasyon işleminde ise her node'daki okur isminin baş harfi U'lu olup olmadığı kontrol edilerek işlem gerçekleştirilir.

### **Main fonksiyonu ve programın ana akışı:**

#### **int main ( int argc , char \*argv [ ] ) Fonksiyonu :**

Main fonksiyonumuzdaki akışı sade ve öz bir şekilde sıralamak gerekilirse;

1. Dosya okuma işlemi yapılır.Eğer dosya bulunamadıysa programa geçilmez ve kullanıcıya hata verilir.Dosya ismi hard coded olarak bulunmaktadır.
2. Dosyanın mevcut olduğu durumda ise veriler okunmaya başlanır ve strtok fonksiyonunun yardımı ile “;” karakteri ile ayırım yapılarak hücre hücre değerler okunur.Satırsal ve sütunsal kontroller ile kitaplar ve kullanıcılar,kitaplara verdiği puanlar ile okunur.Csv dosyasındaki boşluklar o kullanıcının kitaba verdiği puanı 0 olarak okunmamış olarak kabul eder.
3. Gerekli bilgiler geçici struct değişkeninde toplandıktan sonra linkli listeye ekleme yapmak için kullanıcı,**kullaniciOlustur** fonksiyonuna gönderilir ve başa eleman eklenir.
4. Tarama ve ekleme işlemleri bittikten sonra program kullanılmaya tüm verileriyle birlikte hazırdır.
5. Birebir aynı formatta olması kaydı ile kullanıcıdan NU'lu bir okur ismi ve k sayısı alınır.**getOkuyucu** fonksiyonu sayesinde bu okur ismi parametre olarak giderek linkli listeden bu okur ismine sahip olan node döndürülür ve **a** değişkenine atanır.
6. Bu veri girişinin ardından kullanıcının girdiği okur ismi NU'lumu değilmi diye kontrol edilirken verilerin baş harfleri kontrol edilir.Eğer yanlış formatda veya istenen kullanıcı girilmediyse program sonlandırılır.
7. Doğru veri girişlerinin ardından ise similarity hesaplaması,kullanıcının girdiği okura göre tüm nodelar için hesaplanır ve **similarityValue** değişkenlerine atamalar yapılır.
8. Similarityleri olan nodelar linkli liste sıralaması yapan **sortList** fonksiyonu ile büyükten küçüğe sıralanır.
9. Sıralama sonrası kaybolan a değeri için tekrar **getOkuyucu** fonksiyonu yardımıyla node tekrardan a değişkenine atanır.
10. Linkli listemiz zaten büyükten küçüğe doğru sıralandığı için,ilk k tanesinin yazılması kullanıcının girdiği okura en yakın similarity değerine sahip olan okurları göstericektir.Bu sebeble linkli listeyi yazdıran **printUsers** fonksiyonumuza k değişkenimizi gönderiyoruz ve ilk k tane node'ı(okuru)ekrana bastırıyoruz.
11. Programın son bölümünde ise pred hesabı yapılarak kullanıcıyı girdiği okurun(a) okumadığı kitaplara verebileceği tahmini puanlar hesaplanır ve bu puanlardan en yüksek olanı kullanıcıya okuması için önerilir.

## Similaritynin bulunması;

- Ra ve Rb değerleri hesaplanırken;a ve b okurlarının ikisinde okuduğu kitaplara bakılır.
- Rap ve Rbp değerleri hesaplanırken;a ve b okurlarının ikisinde okuduğu kitaplara bakılır.

## Pred değerlerinin bulunması;

- Ra hesaplanırken;a okurunun okumuş olduğu tüm kitaplara bakılır.
- Rb hesaplanırken;b okurunun okumuş olduğu tüm kitaplara bakılır.
- b değerleri,N(k) kadar iterasyon ile similarity değerleri verilen okurlar ile kıyaslanarak elde edilir.

## Ekran çıktıları:

### 1)NU1 için örnek ekran çıktıları;

```
*****KITAP ONERME PROGRAMINA HOSGELDINIZ*****
Lutfen dosyada mevcut olan okuyucu ismini birebir ayni formatta girin:NU1
Benzer kullanıcı sayısı(k):3
U16,0.944911
U5,0.866025
U9,0.848528

THE DA VINCI CODE->3.348084
RUNNY BABBIT->2.726468
Bu durumda önerilecek olan kitap:THE DA VINCI CODE
-----
Process exited after 1.16 seconds with return value 0
Press any key to continue . . .
```

```
*****KITAP ONERME PROGRAMINA HOSGELDINIZ*****
Lutfen dosyada mevcut olan okuyucu ismini birebir ayni formatta girin:NU1
Benzer kullanıcı sayısı(k):20
U16,0.944911
U5,0.866025
U9,0.848528
U12,0.845154
U18,0.700067
U19,0.500000
U6,0.454545
U7,0.157243
U3,0.028916
U11,-0.218218
U4,-0.327327
U14,-0.400000
U13,-0.500000
U17,-0.512272
U20,-0.608922
U8,-0.654654
U1,-0.866025
U10,-0.944911
U15,-0.981980
U2,-1.000000

THE DA VINCI CODE->6.249589
RUNNY BABBIT->0.281904
Bu durumda önerilecek olan kitap:THE DA VINCI CODE
-----
Process exited after 7.855 seconds with return value 0
Press any key to continue . . .
```

## 2)NU2 için örnek ekran çıktıları;

```
*****KITAP ONERME PROGRAMINA HOSGELDINIZ*****
Lutfen dosyada mevcut olan okuyucu ismini birebir ayni formatta girin:NU2
Benzer kullanıcı sayısı(k):3
U11,1.000000
U2,0.981980
U1,0.944911

TRUE BELIEVER->2.344468
THE KITE RUNNER->2.028138
HARRY POTTER
->2.021630
Bu durumda önerilecek olan kitap:TRUE BELIEVER
-----
Process exited after 0.9771 seconds with return value 0
Press any key to continue . . . █
```

```
*****KITAP ONERME PROGRAMINA HOSGELDINIZ*****
Lutfen dosyada mevcut olan okuyucu ismini birebir ayni formatta girin:NU2
Benzer kullanıcı sayısı(k):17
U11,1.000000
U2,0.981980
U1,0.944911
U19,0.866025
U3,0.755929
U5,0.755929
U12,0.755929
U10,0.693375
U17,0.693375
U13,0.500000
U6,0.485662
U20,0.192450
U8,0.142857
U7,0.058026
U16,-0.303433
U9,-0.500000
U14,-0.521286

TRUE BELIEVER->3.039346
THE KITE RUNNER->2.449332
HARRY POTTER
->2.973364
Bu durumda önerilecek olan kitap:TRUE BELIEVER
-----
Process exited after 1.596 seconds with return value 0
Press any key to continue . . . █
```

### 3)NU3 için örnek ekran çıktıları;

```
*****KITAP ONERME PROGRAMINA HOSGELDINIZ*****
Lutfen dosyada mevcut olan okuyucu ismini birebir ayni formatta girin:NU3
Benzer kullanıcı sayısı(k):3
U16,0.500000
U14,0.498058
U15,0.345857

THE WORLD IS FLAT->0.761732
MY LIFE SO FAR->0.760287
Bu durumda önerilecek olan kitap:THE WORLD IS FLAT
-----
Process exited after 1.135 seconds with return value 0
Press any key to continue . . . █
```

```
*****KITAP ONERME PROGRAMINA HOSGELDINIZ*****
Lutfen dosyada mevcut olan okuyucu ismini birebir ayni formatta girin:NU3
Benzer kullanıcı sayısı(k):7
U16,0.500000
U14,0.498058
U15,0.345857
U6,0.114109
U8,-0.080064
U20,-0.243975
U10,-0.288675

THE WORLD IS FLAT->1.253844
MY LIFE SO FAR->1.998020
Bu durumda önerilecek olan kitap:MY LIFE SO FAR
-----
Process exited after 1.746 seconds with return value 0
Press any key to continue . . .
```

#### 4)NU4 için örnek ekran çıktıları;

```
*****KİTAP ONERME PROGRAMINA HOSGELDİNİZ*****
Lutfen dosyada mevcut olan okuyucu ismini birebir ayni formatta girin:NU4
Benzer kullanıcı sayısı(k):3
U2,1.000000
U13,1.000000
U10,0.956183

THE TAKING->1.402734
RUNNY BABBİT->2.019993
Bu durumda önerilecek olan kitap:RUNNY BABBİT
-----
Process exited after 0.9249 seconds with return value 0
Press any key to continue . . .
```

```
*****KİTAP ONERME PROGRAMINA HOSGELDİNİZ*****
Lutfen dosyada mevcut olan okuyucu ismini birebir ayni formatta girin:NU4
Benzer kullanıcı sayısı(k):9
U2,1.000000
U13,1.000000
U10,0.956183
U4,0.866025
U16,0.866025
U3,0.845154
U18,0.762493
U12,0.755929
U9,0.755929

THE TAKING->1.339558
RUNNY BABBİT->1.343514
Bu durumda önerilecek olan kitap:RUNNY BABBİT
-----
Process exited after 1.458 seconds with return value 0
Press any key to continue . . .
```

## 5)NU5 için örnek ekran çıktıları;

```
*****KİTAP ÖNERME PROGRAMINA HOŞGELDİNİZ*****
Lutfen dosyada mevcut olan okuyucu ismini birebir aynı formatta girin:NU5
Benzer kullanıcı sayısı(k):3
U9,0.981980
U18,0.866025
U7,0.852803

TRUE BELIEVER->0.801638
THE KITE RUNNER->-0.796736
HARRY POTTER
->3.566852
Bu durumda önerilecek olan kitap:HARRY POTTER

-----
Process exited after 1.011 seconds with return value 0
Press any key to continue . . .
```

```
*****KİTAP ÖNERME PROGRAMINA HOŞGELDİNİZ*****
Lutfen dosyada mevcut olan okuyucu ismini birebir aynı formatta girin:NU5
Benzer kullanıcı sayısı(k):11
U9,0.981980
U18,0.866025
U7,0.852803
U6,0.648886
U16,0.641689
U14,0.000000
U2,-0.500000
U13,-0.500000
U10,-0.693375
U17,-0.693375
U20,-0.707107

TRUE BELIEVER->4.232428
THE KITE RUNNER->-6.882066
HARRY POTTER
->2.210262
Bu durumda önerilecek olan kitap:TRUE BELIEVER

-----
Process exited after 1.421 seconds with return value 0
Press any key to continue . . .
```

## 6)Yanlış inputlar için ekran çıktıları;

### 6.1.

```
*****KİTAP ÖNERME PROGRAMINA HOŞGELDİNİZ*****
Lutfen dosyada mevcut olan okuyucu ismini birebir aynı formatta girin:U5
Okuyucu algılanamadı.Lutfen dokumandaki mevcut 'NU..' ile başlayan okur ismini BİREBİR AYNI FORMATTA girin.
-----
Process exited after 5.168 seconds with return value 0
Press any key to continue . . .
```



6.2.

```
****KİTAP ONERME PROGRAMINA HOŞGELDİNİZ****
Lutfen dosyada mevcut olan okuyucu ismini birebir aynı formatta girin:nu2
Okuyucu algılanamadı.Lutfen dokumadaki mevcut 'NU..' ile başlayan okur ismini BİREBİR AYNI FORMATTA girin.
-----
```

7)k=3 için oluşan ekran çıktıları sonuç tablosu;

USERS/BOOKS	TRUE BELIEVER	THE DA VINCI CODE	THE WORLD IS FLAT	MY LIFE SO FAR	THE TAKING	THE KITE RUNNER	RUNNY BABBIT	HARRY POTTER
NU1		3.348084					2.726468	
NU2	2.344468					2.028138		2.021630
NU3			0.761732	0.760287				
NU4					1.402734		2.019993	
NU5	0.801638					-0.796736		3.566852

Kodlar:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

struct kullanicilar{    // BUTUN USERLARI TUTACAK STRUCTIMIZ
    char okur[30]; //USER ISMI
    int okunanKitaplar[8]; //OKUNAN KİTAPLARA SIRASIYLA VERİLEN
    PUANLAR.ORN:0,1,5,9,0,.....(8 KİTAP İCİN)
    struct kullanicilar *next; //LINKLI LİSTE STRUCTİ OLDUGU İCİN AYNI TİPTE NEXTİ
    TANIMLANIR.
    float similarityValue; //SADECE U'LU KULLANICILARIN SAHIP OLACAGI,KULLANICININ
    GİRDİĞİ OKUR İSMİNE GÖRE U'LU KULLANICILARDA OLUSACAK PEARSON(SİM) DEĞERİ.
};

struct kullanicilar *head=NULL; //GLOBAL OLARAK TANIMLANAN İLK NULLA ESİT OLAN BOS
NODEUMUZ

void kullaniciOlustur(struct kullanicilar *tempKullanici){ //LINKLI LİSTEYE EKLEME
İSLEMİ,OKUR EKLENİR.
    if(head==NULL){ //İLK ELEMAN EKLENİRKEN
        struct kullanicilar* temp;
        temp=(struct kullanicilar*)malloc(sizeof(struct kullanicilar)); //YER ACILIR
        strcpy(temp->okur,tempKullanici->okur); //PARAMETRE OLARAK VERİLEN STRUCTİN OKUR
        İSMİ İLK NODE'A YAZILIR
        int i; //DONGU DEĞİSKENİ
        for(i=0;i<8;i++){ //BUTUN KİTAP PUANLARININ DEĞERLERİDE AKTARILIR.
            temp->okunanKitaplar[i]=tempKullanici->okunanKitaplar[i];
        }
        temp->next=NULL; //İLK NODE OLDUGU İCİN NEXTİ BOS OLMALIDIR.
        head=temp; //İLK NODE OLDUGU İCİN HEAD OLUR.
    }
    else{ // İKİNCİ VE DAHA SONRASINA AİT OLAN ELEMANLARIMIZI EKLERKEN
        struct kullanicilar* temp;
        temp=(struct kullanicilar*)malloc(sizeof(struct kullanicilar)); //BELLEKTE YER
        ACILIR.
        strcpy(temp->okur,tempKullanici->okur); //PARAMETRE OLARAK GÖNDERİLEN STRUCTİN
        OKUR İSMİ ALINIR.
        int i; //DONGU DEĞİSKENİ.
        for(i=0;i<8;i++){ //TUM KİTAP PUANLARIDA AKTARILIR.
            temp->okunanKitaplar[i]=tempKullanici->okunanKitaplar[i];
        }
        temp->next=head; //BASA EKLENDİĞİ İCİN NEXTİ HEADİ GÖSTERİLİR.
        head=temp; //VE BU ELEMAN ARTIK HEAD OLUR.
    }
}
```

```

}

struct kullanicilar *getOkuyucu(char okuyucu[30]){//PARAMETRE OLARAK GONDERILEN OKUYUCU
ISMINE GORE LINKLI LISTEDEN OKUYUCUYU BULAN FONKSIYON.
    struct kullanicilar *temp;
    temp=head;//HEADDEN BASLA
    while(temp!=NULL){//SONA KADAR GIT
        if(strcmp(temp->okur,okuyucu)==0)//MEVCUT DEGERDEKI OKUYUCU ISMI,PARAMETRE
OLARAK GONDERILEN OKUYUCU ISMINE ESIT ISE
            return temp;//O NODE'U DONDUR.
        temp=temp->next;//BIR SONRAKI NODE'A GEC.
    }
    return 0;
}

float sim(struct kullanicilar *a,struct kullanicilar *b){//SIMILARITY (PEARSON
KATSAYISI)HESABININ YAPILDIGI FONKSIYON.
    float ra=0,rb=0;//RA VE RB DEGERLERININ TANIMLAMALARI.
    int rap=0,rbp=0;//RAp VE RBp DEGERLERININ TANIMLAMALARI.
    int i=0;//DONGU DEGISKENI
    int kitapCounter=0;//RA VE RB DEGERLERI BULUNURKEN TOPLANILARAK BULUNUR VE EN
SONUNDA ORTALAMAYI BULMAK ICIN TUM KITAP SAYISINA ERISILMESI ICIN BU DEGISKEN COUNTER
OLARAK KULLANILIR.

    //KITAP ORTALAMALARI BULUNUR.
    while(i<8){//8 KITAP OLDUGU ICIN 8 ADIM GIT
        if(a->okunanKitaplar[i]!=0 && b->okunanKitaplar[i]!=0){//IKI OKUYUCUNUNDA
OKUMUS OLDUGU KITAPLAR ICIN RA VE RB HESAPLANILIR.
            //RA VE RB ICIN TOPLAYARAK GIDILIR.
            ra+=a->okunanKitaplar[i];
            rb+=b->okunanKitaplar[i];
            kitapCounter+=1;//KITAP SAYISI 1 ARTTIRILIR.
        }
        i++;
    }

    ra=ra/kitapCounter;
    rb=rb/kitapCounter;
    //RA VE RB DEGERLERINDE KITAPLARA VERDIKLERI PUANLARIN TOPLAMI VARDIR.ORTALAMAYI
BULMAK ICIN KITAP SAYISINA BOLUNUR.
    i=0;//AYNI INDIS DEGISKENI KULLANILMASI ICIN i DEGISKENI SIFIRLANIR.

    float simValue=0,pay=0,payda1=0,payda2=0;//simValuede BUTUN HESAPLAMALAR BITTIKTEN
SONRAKI DEGER,pay DEGISKENINDE SIM FORMULUNDEKI PAY'DAKI IFADE,payda1 VE payda2
DEGISKENLERINDE ISE SIRASIYLA SOL VE SAGDAKI IFADELER TUTULUR.

    while(i<8){//rbp degerlerini elde etmek icin b nin okumus oldugu kitaplara bakilir.
        if(a->okunanKitaplar[i]!=0 && b->okunanKitaplar[i]!=0){//IKI OKUYUCUNUN OKUDUGU
KITAPLAR ICIN RAp VE RBp DEGERLERI ELDE EDILIR
            rap=a->okunanKitaplar[i];
            rbp=b->okunanKitaplar[i];
            //DAHA SONRA SIM FORMULU UYGULAMASI ICIN IFADELER ELDE EDILIR.

            pay+=(rap-ra)*(rbp-rb);//SIM FORMULUNDEKI PAY KISMI
            payda1+=pow((rap-ra),2);//SIM FORMULUNDEKI PAYDA KISMININ SOLDAKI IFADESI
            payda2+=pow((rbp-rb),2);//SIM FORMULUNDEKI PAYDA KISMININ SAGDAKI IFADESI
        }
        i++;
    }
    simValue=pay/(sqrt(payda1)*sqrt(payda2));//SIM DEGERIMIZ OLUSTURULDU.
    return simValue;//VE FLOAT TIPINDE DONDURULUYOR.
}

void sortList() { //LINKLI LISTEYI SIM DEGERLERINE GORE BUYUKTEN KUCUGE SIRALIYAN
FONKSIYON.
    //ILK NODE'UMUZ HEADI GOSTERİYOR

```

```

    struct kullanicilar *mevcut = head;
    struct kullanicilar *indis = NULL;
    //SIM DEGERININ TEMPI ICIN temp,OKUR ISIMLERININ TEMPI ICIN tempOkur VE OKUNAN
    KITAPLARA VERILEN PUANLARIN TEMPI ICIN ISE tempOkunanKitaplar DEGISEKENLERI
    TANIMLANILIR.
    float temp;
    char tempOkur[30];
    int tempOkunanKitaplar[8];
    int i;

    while(mevcut != NULL) { //SONA GELINMEDIKCE ISLEM SURER.
        //NODE'UN INDISI MEVCUT NODE'IN NEXTINI TUTAR.
        indis = mevcut->next;

        while(indis != NULL) {
            if(mevcut->similarityValue <= indis->similarityValue) {//EGER mevcut'UN
            SIM VALUESİ indis'IN SIM VALUESINDEN KUCUK VEYA ESIT ISE BU DEGERLER DEGISTIRILIR.

                temp = mevcut->similarityValue;//MEVCUTTAKI SIM VALUE TEMPE ATILIR.
                strcpy(tempOkur,mevcut->okur);//MEVCUTTAKI OKUR ISMIDE GECICI OKUR
                ISMINI TUTAN DIZIYE KOPYALANIR.

                mevcut->similarityValue = indis->similarityValue;//MEVCUTTAKI SIM
                VALUE INDISDEKI SIM VALUEDEKI DEGERI ALIR.
                strcpy(mevcut->okur,indis->okur);//VE MEVCUTTAKI OKUR ISMIDE
                INDISDEKI OKUR ISMINI ALIR.

                for(i=0;i<8;i++){//BIR SONRAKI CYCLE'DA DEGERLERIN KAYBOLMAMASI
                ICIN OKUNAN KITAPLARA VERILEN PUANLARIN AKTARILMASI YAPILIR.
                    tempOkunanKitaplar[i]=mevcut->okunanKitaplar[i];
                    mevcut->okunanKitaplar[i] = indis->okunanKitaplar[i];
                    indis->okunanKitaplar[i]=tempOkunanKitaplar[i];
                }

                indis->similarityValue = temp;//indis'IN SIM VALUESINE TEMP
                DEGISEKENINDEKI SIM VALUE ATILIR.
                strcpy(indis->okur,tempOkur);//indisDEKI OKUR ISMINEDE TEMPDEKI
                OKUR ISMI KOPYALANIR.

            }
            indis = indis->next;//BIR SONRAKI INDIS GOSTERILIR.
        }
        mevcut = mevcut->next; //MEVCUT DEGERI BIR SONRAKI DEGERE GECER.
    }
}

void pred(struct kullanicilar *a,char books[8][50],int N){//PRED VALUE HESAPLAMASI
YAPAN FONKSIYON.PARAMETRE OLARAK KULLANICININ GIRDIGI OKUR,KITAPLAR VE TOPLAM
SEMBOLUNUN SINIRINI BELIRLEYECEK OLAN N(K) ALIR.
    float ra=0,rb=0;//RA VE RB DEGERLERININ TANIMLAMALARI.
    int rbp=0;//Rbp DEGERININ TANIMLANMASI.
    int noKitapCounter=0,i=0,j=0,y=0;//noKitapCounter DEGISEKENI RA BULUNURKEN A
    KULLANICISININ OKUMADIGI KITAP SAYISINI TUTAR.
    float predValue=0,pay=0,payda=0;//PRED VALUEYI OLUSTURURKEN ISLEMLER ICIN PAY,PAYDA
    VE PREDVALUE ADINDA DEGISEKENLERIN TANIMLAMALARI YAPILIR.
    char tempBooks[8][50];//a KULLANICISININ OKUMADIGI KITAPLAR.
    char predBook[50];//SON OLARAK ONERILECEK OLAN KITABI TUTACAK OLAN DEGISEKENIMIZ.
    float preds[8];//PRED DEGERLERINI TUTACAK OLAN DEGISEKEN.

    while(i<8){//KITAP SAYISI KADAR GIT
        ra+=a->okunanKitaplar[i];//TUM KITAPLARI TOPLA
        if(a->okunanKitaplar[i]==0){//A'NIN OKUMADIGI BIR KITAP VAR ISE(PUANI 0 ISE)
            noKitapCounter+=1;//OKUNMAMIS KITAP SAYISINI 1 ARTTIR
            strcpy(tempBooks[y],books[i]);//VE BU KITABI tempBooks ADLI DEGISEKENE AT
            y++;//INDIS DEGERI ARTTIRILIR.
        }
    }
}

```

```

        i++; //GENEL INDIS DEGERI ARTTIRILIR
    }

    ra=ra/(8-noKitapCounter); //RA BULUNUR.
    i=0; //AYNI DEGISKEN INDIS OLARAK KULLANILMAYA DEVAM ETMESI ICIN SIFIRLANILIR

    //PAY VE PAYDADAKI ISLEMLER ICIN B DEGISKENI SUREKLI DEGISECEGINDEN B DEGISKENININ
    TANIMLAMASI YAPILIP BELLEKTE YERI ACILIR.
    struct kullanicilar *b=(struct kullanicilar*)malloc(sizeof(struct kullanicilar));
    b=head; //EN BASA VERILIR. (K ADET KADAR ILERLENECEK.DIZI SIRALI OLDUGU ICIN SADECE
    ISTENEN B DEGERLERI ISLEME GIRECEK.)

    //INDIS DEGISKENLERI SIFIRLANIR.
    j=0;
    y=0;
    int x,bKitapCounter=0; //x INDIS,bKitapCounter ISE B'NIN OKUDUGU KITAPLARIN SAYISINI
    TUTAN DEGISKEN.
    //MEVCUT NODEDAKI OKURUN 'U..'LU BIR KULLANICI OLUP OLMAMASININ KONTROLU ICIN
    TANIMLANAN DEGISKENLER.
    char cmp1;
    char cmp2='U';
    while(i<8){
        if(a->okunanKitaplar[i]==0){ //A'NIN OKUMADIGI BIR KITAP ICIN;
            j=0;
            //SIRALI DIZI OLMASINA RAGMEN U'LU KULLANICILARIN ISLEME ALINDIGININ
            KONTROLU

            b=head; //BASTAN BASLANIR
            cmp1=b->okur[0]; //ILK HARFI ALINIR
            if(cmp1!=cmp2){ //U'LU BIR KULLANICI DEGIL ISE
                while(cmp1!=cmp2){ //U'LU KULLANICIYI GORENE KADAR
                    b=b->next; //ILERLER.
                    cmp1=b->okur[0];
                }
            }

            while(j<N){ //N KADAR ILERLE
                rbp=b->okunanKitaplar[i]; //RbP ELDE EDILDI.

                bKitapCounter=0; //B'NIN OKUDUGU KITAP SAYISI SIFIRLANIR.
                for(x=0;x<8;x++){ //TOPLAM KITAPLAR KADAR ILERLE.
                    if(b->okunanKitaplar[x]!=0){ //BNIN OKUDUGU BIR KITAP ISE.
                        rb+=b->okunanKitaplar[x]; //RB TOPLAMI ICIN GENEL TOPLAMA
                        SOK.

                        bKitapCounter++; //B NIN OKUDUGU KITAPLARI 1 ARTTIR.
                    }
                }
                rb=rb/bKitapCounter; //TOPLAM KITAP PUANINI,B'NIN OKUDUGU KITAP
                SAYISINA BOL VE RBYI BUL.

                pay+=sim(a,b)*(rbp-rb); //PRED HESABININ PAY KISMI
                payda+=sim(a,b); //PRED HESABININ PAYDA KISMI
                rb=0; //RB SIFIRLANIR.
                //ITERASYON ISLEMI
                if(b->next!=NULL){ //B'DEN SONRAKI NODE BOS DEGIL ISE
                    cmp1=b->next->okur[0]; //COMPARE ISLEMI ICIN O NODE'UN BAS
                    HARFINI AL.

                    if(cmp1==cmp2){ //BIR SONRAKI NODE'UN BAS HARFI 'U' ILEMI BASLIYOR
                    KONTROL ET.

                        b=b->next; //U'LU BIR KULLANICI ISE BIR SONRAKI NODE'A GEC.
                    else{ //DEGIL ISE;
                        while(b->next!=NULL && cmp1!=cmp2){ //SONA VE U'LU BIR
                        KULLANICIYA GELMEDIGI SURECE ILERLE.
                            b=b->next;
                            cmp1=b->next->okur[0]; //BIR SONRAKI KULLANICININ BAS
                            HARFINI AL
                        }
                    }
                }
            }
        }
    }

```

```

        if(b->next!=NULL)//MEVCUT INDISDEKI NODE'DA ARTIK 'U'LU BIR
KULLANICI VAR.FAKAT SONA GELINIP GELINMEMESININ KONTROLUDE YAPILIYOR.
        b=b->next;//ITERASYON ISLEMI GERCEKLESIYOR.
    }
    j++;
}

predValue=ra+(pay/payda);//PRED HESABI YAPILIR VE DEGER ELDE EDILIR.
//PAY VE PAYDA BIR SONRAKI B OKURU ICIN SIFIRLANIR.
pay=0;
payda=0;
//EN BUYUK PREDE SAHIP OLANI GECICI KITAP DIZISINDE ILK ELEMANA KOYMA
ISLEMI
    preds[y]=predValue;//BULUNAN PRED DEGERI PREDLERIN BULUNDUGU DIZIYE
ATANIR.
    if(y==0)//ILK DONGUDE OLUP OLMADIGININ KONTROLU y DEGISKENI ILE
YAPILIR.EGER ILK DONGUDE ISE ONERILECEK KITABI TUTAN DEGISKENE DIREK MEVCUT KITABI
KOPYALA.
        strcpy(predBook,tempBooks[y]);
        printf("%s->%f\n",tempBooks[y],predValue);//A'NIN OKUMADIGI KITABI VE
PRED DEGERINI EKRANA BASTIR.
        if(preds[y]>preds[0]){//EGER 1DEN FAZLA KONTROL YAPILDIYSA EN BUYUK
PRED DEGERINI KULLANICIYA ONERMEK ICIN KONTROL YAPILIYOR.
            strcpy(predBook,tempBooks[y]);//EGER MEVCUT PRED DEGERI ILK
DEGERDEN BUYUK ISE ONERILECEK KITAP BU PRED DEGERINE SAHIP OLAN OLUYOR.
            preds[0]=preds[y];//VE PREDIN ILK ELEMANINA EN BUYUK OLAN PRED
DEGERI YAZILIYOR.
        }

        y++;
        predValue=0;//PRED HESABININ SON HALINI TUTAN DEGISKEN SIFIRLANIYOR.
    }
    i++;
}
printf("Bu durumda onerilecek olan kitap:%s",predBook);//KITAP ONERILMESI
YAPILIYOR.
}

void printUsers(int k){ //K ADIM BOYUNCA LINKLI LISTEYI BASTIRIR.
    struct kullanicilar *temp;
    temp=head;
    int i=0;
    //SADECE U'LU KULLANICILARI BASTIRMASI ICIN YAPILAN KARAKTER KONTROLU DEGISKENLERI.
    char cmp1;
    char cmp2='U';
    while(temp!=NULL && i<k){//SONA GELINMEDIGI VE K ADETE ULASMADIGI SURECE
        cmp1=temp->okur[0];//OKURUN ILK HARFINI AL
        if(cmp1==cmp2){//U'LU BIR KULLANICI ISE
            printf("%s,%f\n",temp->okur,temp->similarityValue);//SIM DEGERI ILE
BIRLIKTE EKRANA BASTIR.
            i++;
        }
        temp=temp->next;//ITERASYONA DEVAM EDILIR.
    }
}

int main(int argc, char *argv[]) {
    printf("*****KITAP ONERME PROGRAMINA HOSGELDINIZ*****\n");
    FILE *fp;
    fp = fopen("RecomendationDataSet.csv","r");
    if(fp==NULL)//DOSYA BULUNAMADIYSA.
        printf("Dosya acilirken bir hata olustu.");
    else{//BULUNDUYSA ANA AKISA GECILIR.
        int tempPuanlar[8];
        char *tempS;
        char books[8][50];//8 ELEMENLI,50 KARAKTER UZUNLUKLU KITAPLARI TUTAN CHAR
MATRISI(STRING DIZISI).
    }
}

```

```

    char csvData[512];
    int i=0,j=0,kCounter=0;//i degeri kacinci sutun,j degeri kacinci satir,ilksatir
    degeri ilk satir atlandimi yoksa atlanmadimi kontrolu ve kCounter degeri ise kac
    kullanıcı olduğunu tutar.
    struct kullanicilar *tempKullanici;
    while(fgets(csvData,512,fp)){//Dosyanin okundugu genel blok.
        tempKullanici=(struct kullanicilar*)malloc(sizeof(struct
kullanicilar));//Bellekte yer acilir.
        tempS=strtok(csvData,"");
        while(tempS!=NULL){//Mevcut satirin sirayla sutunlari taraniyor.
            if(i>0 && j==0){//kitaplar aliniyor.
                strcpy(books[i-1],tempS);
            }
            if(i==0 && j>0){//bir kullanıcı elde edildiği kisim.
                strcpy(tempKullanici->okur,tempS);
                kCounter++;
            }
            if(i>0 && j>0){//puanların erişildiği kisim.
                if(tempS=='\0'){//Bosluk var ise 0 puandır.
                    tempPuanlar[i-1]=0;
                }
                else{
                    tempPuanlar[i-1]=atoi(tempS);//string olan gelen puan degerini
atoi fonksiyonu ile integer bir degere ceviriyoruz.
                }
            }

            tempS=strtok(NULL,"");
            i++;
        }//bir satir okundu.Bir sonraki donguye giriste alttaki satir okunacak.
        for(i=0;i<8;i++){
            tempKullanici->okunanKitaplar[i]=tempPuanlar[i];
            tempPuanlar[i]='\0';
        }
        kullanıcıOlustur(tempKullanici);

        j++;//satir artti.
        i=0;//Sutun verisi sifirlandi.
        free(tempKullanici);//temp struct yapimizin sifirlanmasi icin free edilir.
    }

    char okuyucuName[30];//kullanici tarafından girilmesi beklenen okur ismini
    tutacak olan degisken.
    printf("Lutfen dosyada mevcut olan okuyucu ismini birebir ayni formatta
    girin:");
    gets(okuyucuName);//okur ismi alınır.

    struct kullanicilar *a=(struct kullanicilar*)malloc(sizeof(struct
kullanicilar));//Listede mevcutmu diye kontrol edilir ve bellekte yer acilir.
    a=getOkuyucu(okuyucuName);//Liste kontrolu sonucu donen deger alinir.
    char cmp1=okuyucuName[0];//Kullanıcının girdigi okur isminin ilk karakteri
    alinir.
    char cmp2='N';//Okur ismi N ile basliyormu diye kontrol edilmesi icin ikinci bi
    degiskende 'N' karakteri tutulur.

    if(a!=NULL && cmp1==cmp2){//Okuyucu ismi hem var hemde NU lu bir okuyucu ise
    programin isleyisine devam edilir.NUlu okuyucu olup olmadigi girilen okuyucu isminin
    ilk karakterinden anlasilir.
        int k;
        printf("Benzer kullanıcı sayısı(k):");
        scanf("%d",&k);

        //Similaritylerin bulunmasi ve atanmasi.
        cmp2='U';//mevcut node'daki okur U'lu bir degermi degilmi kontrolunun
        yapılmasi icin bu degiskene bu sefer U atilir.
        struct kullanicilar *b=(struct kullanicilar*)malloc(sizeof(struct
kullanicilar));

```

```

        struct kullanicilar *tempB=(struct kullanicilar*)malloc(sizeof(struct
kullanicilar));
        tempB=head;
        while(tempB!=NULL){
            b=getOkuyucu(tempB->okur);
            cmp1=b->okur[0];//U harfi ile compare islemi icin okurun bas harfi
alinir.
            if(cmp1==cmp2){//simdiki nodedaki okur U'lumu diye kontrol edilir.
                b->similarityValue=sim(a,b);//sirayla U'lu okurların sim degerleri
bulunur.
            }
            tempB=tempB->next;
        }

        sortList();//liste siralanir.
        a=getOkuyucu(okuyucuName);//siralanan listeden sonra kullanicinin node
degeri kayboldugu icin girilen kullanıcı adına gore tekrardan node'un bulunması ve a
degiskenine atanması.
        printUsers(k);//sim degerleri bulunan liste k boyunda yazdirilir.
        printf("\n");
        pred(a,books,k);//pred degeri,son bolumu olusturan fonksiyonumuz.
    }
    else//kullanici ismi girisinde bir hata var ise program durur.
        printf("Okuyucu algılanamadi.Lutfen dokumandaki mevcut 'NU..' ile baslayan
okur ismini BIREBIR AYNI FORMATTA girin.");
    }

    return 0;
}

```