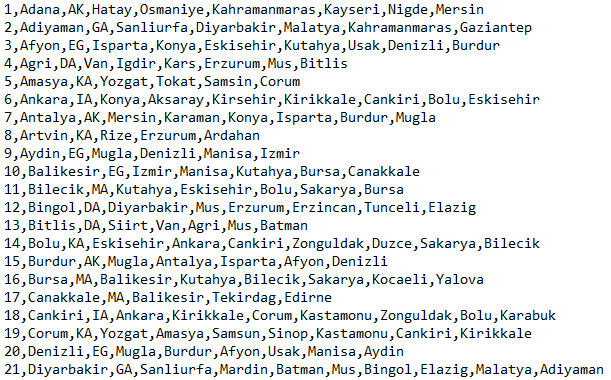
**Özet - Coğrafi Bilgi Sistemi (CBS), dünya üzerindeki karmaşık sosyal, ekonomik, çevresel vb. sorunların çözümüne yönelik mekâna/konuma dayalı karar verme süreçlerinde kullanıcılara yardımcı olmak üzere, büyük hacimli coğrafi verilerin; toplanması, depolanması, işlenmesi, yönetimi, mekânsal analizi, sorgulaması ve sunulması fonksiyonlarını yerine getiren donanım, yazılım, personel, coğrafi veri ve yöntem bütünüdür. Bu proje kapsamında, şehirlerin komşuluk ilişkilerini dikkate alan çok basit bir harita verisi üzerinde bağlı listelerle ilgili temel işlevleri yerine getiren bir uygulama yapılmasıdır.**

**Anahtar Kelimeler – bağlı liste, dügüm,yapılar, traverse, dosya okuma**

I. GİRİŞ

Programımız bir dosya içerisinde belirtilmiş plaka kodu, şehir ismi, şehrin bölgesi ve komşularından oluşan verileri okuyarak belirli bir bellek boyutu alanında tutup bağlı liste veri yapısına uygun şekilde tanımlanmıstır.

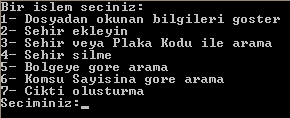


*Şekil 1: sehirler.txt*

II. YÖNTEM

Programımızı C programlama dili kullanılarak yapılmış olup çeşitli çoklu seçimli koşul yapıları kullanılarak (if-else,switch-case,while,for vs.) ve bazı özel fonksiyonlar ile geliştirilmiştir.

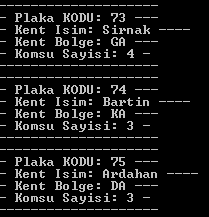
III. DENEYSEL SONUÇLAR

 İlk başta kullanıcı programı açarken karşısına bir konsol ekranı(siyah ekran veya cmd.exe) penceresi açılıyor. Bu pencerede ilgili işlemleri yapmak için karşısına bir menü gelip “Hangi işlemi yapmak istersiniz?” sorusunun altında çıkan işlemlerinin numarası hangisi ise ona basarak o işlemi gerçekleştiriyor.

Şekil 2:Kullanıcı Menüsü

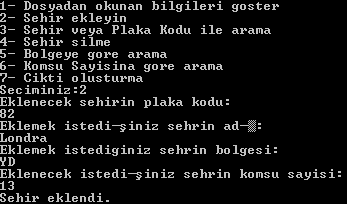
Kullanıcı yukarıda şekil 2’deki gibi bir menüden işlem seçip ilgili işlem hangisi ise onu yapacağından bahsetmiştim.

Eğer kullanıcı 1 numaralı işlemi seçer ise “sehirler.txt” dosyasındaki verilerin bağlı listeye tanımlanmış halini göreceklerdir.



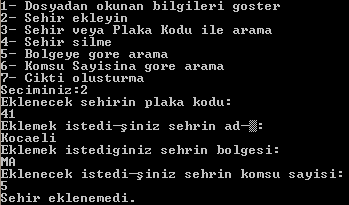
Şekil 3: Kullanıcı 1’e basması sonucu oluşan listeleme

Eğer kullanıcı 2 işlemini seçer ise kullanıcıdan bir takım bilgiler alarak ilgili bağlı listeye şehir eklemesi yapacaktır. Burada normalde komşuluk ilişkisi olarak yeni komşu ekleme işlemini maalesef yapamıyor.



Şekil 4:Kullanıcı 2. işlemi seçtiği zamanda şehir ekleme yapılması

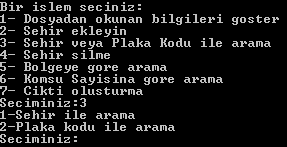
Bu şehir ekleme sırasında ilgili şehir var ise o şehir bir daha eklenemiyor. Örneğin “Kocaeli” şehrini baz alalım. Kocaeli dosya verilerinde var iken yeniden bir ekleme yapıldığında onu eklemeyecektir. Sebebi plaka kodu kontrolü ve sehir kontrolü yapmasıdır.



Şekil 5: Bir şehir varsa o şehrin eklenememesi örneği

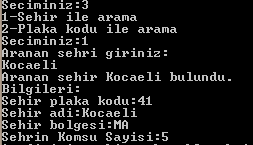
Eğer kullanıcı 3 işlemini seçerse arayacağı şehrin bilgilerine isim üzerinden veya plaka kodu üzerinden görebiliyor.

Kullanıcının karşısına önce nasıl bir arama yapılacağı soruluyor.Daha sonra ilgili bilgiler girildikten sonra şehir bilgileri gösteriliyor.



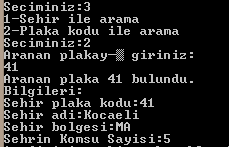
Şekil 6:Kullanıcı 3 işleminde arama işlemini yapması

Kullanıcı şehir ile aramayı seçerse, ilgili şehir ismini girerek aradığı şehrin ismini ve bilgilerine ulaşabilir.



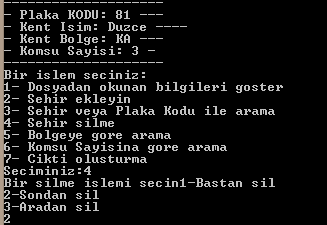
Şekil 7: Şehir ile arama gerçekleştirimi

Kullanıcı plaka kodu ile aramayı seçerse, ilgili plaka kodunu girerek bilgileri görebilir.

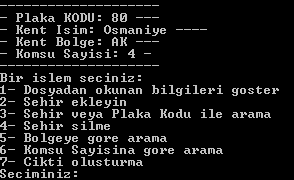


Şekil 8: Plaka Kodu ile arama gerçekleştirimi

Eğer kullanıcı 4 işlemini seçerse şehir silme işlemini gerçekleştiriyor.



Şekil 8: Sondan şehir silme işlemi uygulaması



Şekil 9: İşlemin sonucu

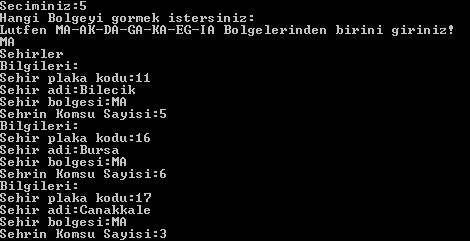
Burada en son gördügünüz “Düzce” şehrimiz artık işlem uygulandığında siliniyor.

Eğer kullancı 5 işlemini seçerse bölgeye göre listede arama yapıp bölgedeki tüm şehirlerin bilgileri gösteriliyor.



Şekil 10: Bölge ile Arama Gerçekleştirimi

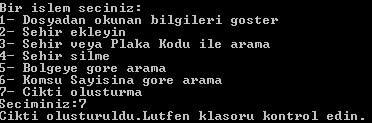
Gördüğünüz şekilde bir arama yaptırmak gerekli bölge kodlarını söylüyor ve onları yazarak şehirleri görebiliyoruz.



Şekil 11: Arama sonucu gösterilen şehirler

Eğer kullanıcı 6. işlemi seçerse aynı 5. işlemdeki gibi o işlemde de komşu sayıları aynı olan şehirleri sıralayacaktır.

Son olarak 7.işlem ise en sonunda bu ekleme veya silme olayları bittiğinde çıktı almak isterse eline bir .txt formatında bir dosya alıyor.



Şekil 12: Çıktı oluştuğunun bildirilmesi



Şekil 13: cikti.txt nin bulunması

IV.YALANCI KOD

**Şehir düğüm yapısını gösteren kod:**

struct node{

int plakaveri;

char \*sehir;

char \*bolge;

int komsu\_sayisi;

struct node \*prev;

struct node \*next;

struct node2 \*next2;

};

**Düğüm yapısınının oluşmasını sağlayan kod:**

struct node\* NewNode(int plakakod,char \*sehirad,char \*sehirbolge,int komsu\_sayi){

struct node \*Yenidugum=(struct node\*)malloc(sizeof(struct node));

Yenidugum->plakaveri=plakakod;

Yenidugum->sehir=sehirad;

Yenidugum->bolge=sehirbolge;

Yenidugum->komsu\_sayisi=komsu\_sayi;

Yenidugum->next=NULL;

Yenidugum->prev=NULL;

return Yenidugum;

};

**Listeye atılmasını sağlayan kod:**

int SehirEkle(int plaka,char \*ad,char \*bolge,int kmsu\_sayi){

struct node \* eklenecek\_sehir = NewNode(plaka,ad,bolge,kmsu\_sayi);

if (start == NULL)

{

start = eklenecek\_sehir;

}

else

{

temp=NULL;

temp=start;

while (temp->next != NULL)

{

if(strcmp(temp->sehir,eklenecek\_sehir->sehir)==0 || temp->plakaveri==eklenecek\_sehir->plakaveri){

return 0;

exit(1);

}else

temp = temp->next;

}

eklenecek\_sehir->prev = temp;

temp->next = eklenecek\_sehir;

return 1;

}

}

**Listeyi yazdırmayı sağlayan kod:**

void yazdir(){

temp=start;

while(temp!=NULL){

printf("--------------------\n");

printf("- Plaka KODU: %d ---\n",temp->plakaveri);

printf("- Kent Isim: %s ----\n",temp->sehir);

printf("- Kent Bolge: %s ---\n",temp->bolge);

printf("- Komsu Sayisi: %d -\n",temp->komsu\_sayisi);

printf("--------------------\n");

temp=temp->next;

}

}

V. SONUÇ

Sonuç olarak bu projede bağlı liste yapısının nasıl olduğunu ve nasıl kullanıldığını anladım. Büyük bir sıkıntı olan komşuluk iişkisi anlayamadığım için yapamadım. Ama geliştirirek kendimi bu sıkıntıdan kurtulacağımı düşünüyorum.

VI.KAYNAKÇA

1.Break ve Exit Komutu. (n.d.). Retrieved from http://cdiliogren.blogspot.com/2008/10/break-ve-exit-komutu.html

2.C ile Dosya İşlemleri. (2008, 22). Retrieved from http://bilgisayarkavramlari.sadievrenseker.com/2008/10/22/c-ile-dosya-islemleri/

3.Search Doubly Linked List In C. (n.d.). Retrieved from https://www.tutorialspoint.com/learn\_c\_by\_examples/search\_doubly\_linked\_list\_in\_c.htm

4.Çift Yönlü Ba'l? Liste Tüm ?'lemler. (2017, November 12). Retrieved from https://www.algoritmauzmani.com/cift-yonlu-bagli-liste-tum-islemler/