**Bir Muhitin Elektrik Hattı Bilgileri Programı**

INF212 Project #1

Prepared by

1801022071

Alperen Arslan

INF 212 Algorithms and Programming II

Spring 2020

Electronic Engineering

Date Submitted: 27.05.2020

Evaluation/Değerlendirme

|  |  |
| --- | --- |
| Report layout /Rapor düzeni | 10 -- |
| Project Flowchart and functions flow chart /Projenin ve fonksiyonların Akış Diyagramı | 25 -- |
| Programming /Programlama Struct, Functions | 45 -- |
| Questions / Sorular | 20 -- |

|  |
| --- |
| Project Objective /Projenin Amacı |
| Coğrafi bilgi sistemi verilerinden yararlanarak elektrik enerji sisteminin dağıtım şebekesindeki indirici Merkez bilgilerini,müşteri bilgilerini, hat bilgilerini, hat sayısını, toplam hat uzunluğunu, toplam müşteri sayısını, müşteri tiplerine göre müşteri sayısını, indirici merkezin kimlik numarası ve koordinat bilgisini veren bir program yazmak. |
| Description of Problem / Problem Tanımlama |
| Çözmem için verilen problem, programın indirici, müşteri ve hat bilgilerini, bilgilerin belli bir sistematiğe göre yazıldığı .txt dosyasından doğru bir şekilde almak ve doğru biçimde bu bilgileri kaydedip, belli işlemlerde kullanarak komut dosyasına ve .txt dosyasına çıktı vermek. |
| **Description of Method / Metodunun Tanımlaması** |
| Problemi çözerek uyguladığım ilk yöntem .txt dosyasında indirici, müşteri ve hat bilgilerinden kaçar tane olduğunu hesaplamaktı. Belli koşullar koyarak indirici, müşteri ve hat bilgilerinin satır sayılarını saydırarak kaçar tane olduklarını öğrendim. Daha sonra bu bilgileri ve belli string fonksiyonlarını da kullanarak .txt dosyasından bilgileri ilgili struct array yapılarına kaydettim. Problemin bu sayede benim için ilk bölümünü hallettim. İkinci bölümde ise bu bilgilerle benden yapılması istenen işlemleri yapmam gerekiyodu. Genelde .txt dosyasına çıktı vermem gerekiyordu. Mesafe hesaplaması için bir tane genel mesafe hesaplayan fonksiyon kullandım. Bu fonksiyonu kullanarak birkaç yerde farklı çeşitlerde istenen mesafeleri de hesaplatıp yazdırdım. |
| Text of Program/ Programın Kodunu Tanımlaması |
| Programın kodu çok uzun olduğu için raporun sonuna eklendi. |
| User’s guide / Kullanıcı Rehberi |
| Programı kullanacak olan kullanıcıya ilk tavsiyem proje dosyasını bir klasör içine alması olur. Program çalıştırıldığında 2 adet .txt dosyası oluşturucağından dolayı daha verimli bir kullanım için dosyaların bir klasörde olması iyi olur. Aslında programın sistemi genel olarak bilgilerin oldugu .txt dosyasını okuyup, bu bilgileri belli işlemlerde kullanıp başka bir .txt dosyasına yazmak. Kullanıcıdan istenenler ise kontrol edilmek istenen musteri ve hat bilgilerinin id’sidir. Kullanıcı bu bilgileri girdikten sonra program, eğer işlemler sorunsuz bir şekilde tamamlandıysa, yazdırma işlemi tamamlandı uyarısı vererek kapanacaktır. Program bulunduğu klasöre iki adet .txt dosyası oluşturucaktır. Bunlardan Sonuclar.txt adlı dosyada indirici sayisi, musteri sayisi, hat sayisi, indirici bilgileri, kullanıcının talep ettiği musteri ve hat bilgileri gibi bilgilerin yazılı olacaktır. Musteri\_kus\_ucusu\_mesafe adlı ikinci dosyada ise sistemde bulunan her müsteri ile indirici arasındaki kuş uçuşu mesafenin tek tek yazılı olacaktır. Programda olası bir sınırlama bulunmamaktadır. Eğer kullanıcı yanlış id bilgileri girerse zaten program ilgili .txt dosyasına ilgili id’de müşteri bulunamadı şeklinde uyarı verecektir. Programın olası hata vermesinin olsa olsa sebebi bilgilerin alınmasını istenen .txt dosyasının programın bilgi okuma algoritmasına uygun olmayacak şekilde girilmiş olması olabilir. Bu hatanın alınmasını en aza indirmek için satır sayma ve belli string fonksiyonları kullanılmıştır fakat dosyalarda büyük bozulmalar olursa program hata verebilir. |
| Results of the solution / Programın Sonuçları Program çalıştırıldığı zaman aşağıdaki görselde olduğu gibi kullanıcıdan kontrol edilmek istenen hat ve müşterilerin id’si istenecektir.  Program çalışmayı bitirdiği zaman iki adet .txt dosyası oluşturur. Bunlarda Sonuclar isimli olanda istenen bilgilerin çoğu yazılı olur. Eğer kullanıcının girdiği id’ye ait bir musteri yok ise aşağıdaki gibi bir görselle karşılaşırız.    Eğer sistemde varolan musteri ve hat id’si girilirse aşağıdaki gibi bir görsel alınır.  Programın oluşturduğu ikinci .txt dosyasında ise her bir müşterinin indirici ile arasındaki kuşuçuşu mesafe yazılı olacaktır. |
|  |
| Flowchart of the Program / Programın Akış Diyagramı |
|  |
| Conclusion and Remarks / Sonuç ve Notlar |
| Programım gayet düzgün çalışmakta. .txt dosyasından belli bir düzene göre yazılmış olsa da binlerce veriyi alıp, o verileri doğru şekilde ilişkili oldukları yapılara kaydetmek oldukça zordu. Gerçekten zor zamanlar yaşattı bu işi doğru şekilde yapmaya çalışmak. Türkçe – İngilizce farketmeksizin olabildiğince çok kaynak araştırmak zorunda kaldım. Hatta araştırmanın verdiği heyecanla birçoğunu da kaynakça’ya yazmayı unutmuş olabilirim. Aslında proje, genel olarak bu dönem verilen projelerde de olduğu gibi, bana nasıl araştırma yapmam gerektiğini öğretti diyebilirim. Öğrendiğim en önemli şeyin bu olduğunu düşünüyorum. Yerli yabancı onlarca forumu, başka insanların neler yaptığını, onların karşılaştıkları sorunlara karşı ne gibi öneriler aldıklarını, o sorunları nasıl aştıklarını ve o önerileri ,tabi ki de birebir aynı sorunlar olmadığı için, nasıl kendi koduma uyarlayabileceğimi öğrendim.Tabi ki de c dili ile alakalı da birçok şey öğrendim. Özellikle string fonksiyonları ile alakalı birçok fonksiyon ve onları nasıl kullanabileceğimi öğrendim. Ama dediğim gibi en önemlisi sorunlarla karşılaştığım zamanlarda onları aşmak için nasıl araştırma yapmam gerektiğini büyük ölçüde öğrendim diyebilirim.  Program gayet kullanıcı dostu diyebilirim. Kullanıcıdan beklenen kullanıcı rehberinde de anlatıldığı gibi, sadece bütün verileri ve kod dosyalarını aynı klasörde bir araya toplamak. Veri dosyalarında bir bozulma olmadığı sürece program sıkıntısız çalışacaktır. Eğer txt dosyalarının isimlerinde değişiklik yapıldıysa bu değişikliklerin kod’a da yapılması gerekir. |
| References / Kaynaklar |
| * Kodlakafa.com * Alldatasheet.com * Stackoverflow.com * Algoritmaornekleri.com * Arduinolibraries.com * Github.com * Sanatsalbilgi.com * Theengineeringprojects.com * Rapidtables.com * Koddefteri.com * Erhanulman.com * Forum.donanimhaber.com * Cagataycebi.com * Forum.csharpnedir.com * En.wikipedi0.org * Hayaletveyap.com * Grafiboyz.blogcu.com * Aliozcan.org * Gelecegiyazanlar.turkcell.com.tr * Iucoders.com * Sanalkurs.net * Buraksenyurt.com * Microcore.com * Link24in.wordpress.com * Soyunmezoglubaris.blogspot.com * Motorobit.com * Kodumundunyasi.net |

**KOD:**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include <stdbool.h>

#include<math.h>

#define okunanDosya "Realistic\_system.txt" *//verilerin okundugu dosyanin ismi degistirildiyse sadece buranin degistirilmesi yeterlidir.*

**typedef** **struct** indirici**{**

**int** id**;**

**float** x1**;**

**float** y1**;**

**float** x2**;**

**float** y2**;**

**}**indirici**;**

**typedef** **struct** musteri**{**

**int** id**;**

**int** tip**;**

**float** x1**;**

**float** y1**;**

**float** x2**;**

**float** y2**;**

**}**musteri**;**

**typedef** **struct** hat**{**

**int** id**;**

**float** x1**;**

**float** y1**;**

**float** x2**;**

**float** y2**;**

**}**hat**;**

**float** mesafeHesaplayici**(float** x1**,** **float** x2**,float** y1**,float** y2**);**

**int** main**(){**

**char** ch**[**100**];**

FILE **\***p **=** fopen**(**okunanDosya**,"r");**

**char** **\***checker **=** NULL**;**

**int** indiriciSayisi **=** 0**;**

**int** musteriSayisi **=** 0**;**

**int** hatSayisi **=** 0**;**

**int** tipMesken **=** 0**;**

**int** tipTicari **=** 0**;**

**int** tipSanayi **=** 0**;**

*//dinamik sekilde array olusturmak icin oncelikle sayilar tespit edilecek*

**if((**p**=**fopen**(**okunanDosya**,"r"))** **!=**NULL**){**

**while(!**feof**(**p**)){**

fgets**(**ch**,**100**,**p**);**

checker **=** strstr**(**ch**,** **"INDIRICI\_ID");**

**if(**checker **==** ch**){**

**while(true){**

fgets**(**ch**,**100**,**p**);**

checker **=** strstr**(**ch**,** **"END");**

**if(**checker **==** ch**){**

**break;**

**}** **else** **{**

indiriciSayisi**++;**

**}**

**}**

**}**

checker **=** strstr**(**ch**,** **"MUSTERI\_ID");**

**if(**checker **==** ch**){**

**while(true){**

fgets**(**ch**,**100**,**p**);**

checker **=** strstr**(**ch**,** **"END");**

**if(**checker **==** ch**){**

**break;**

**}** **else** **{**

musteriSayisi**++;**

**}**

**}**

**}**

checker **=** strstr**(**ch**,** **"HAT\_ID");**

**if(**checker **==** ch**){**

**while(true){**

fgets**(**ch**,**100**,**p**);**

checker **=** strstr**(**ch**,** **"END");**

**if(**feof**(**p**)** **||** checker **==** ch**){**

**break;**

**}** **else** **{**

hatSayisi**++;**

**}**

**}**

**}**

**}**

**}** **else{**

printf**("Dosya Bulunamadi.");**

exit**(**1**);**

**}**

fclose**(**p**);**

*// sayilar tespit edilince, bu sayi buyukluklerinde arrayler olusturulacak*

indirici indiriciler**[**indiriciSayisi**];**

musteri musteriler**[**musteriSayisi**];**

hat hatlar**[**hatSayisi**];**

**char** **\*** token**;**

**int** tabIndex **=** 0**;**

**int** arrayIndex **=** 0**;**

**if((**p**=**fopen**(**okunanDosya**,"r"))** **!=**NULL**){**

**while(!**feof**(**p**)){**

fgets**(**ch**,**100**,**p**);**

checker **=** strstr**(**ch**,** **"INDIRICI\_ID");**

**if(**checker **==** ch**){**

printf**("indiriciler isleniyor\n");**

**while(true){**

fgets**(**ch**,**100**,**p**);**

checker **=** strstr**(**ch**,** **"END");**

**if(**checker **==** ch**){**

**break;**

**}** **else** **{**

token **=** strtok**(**ch**,** **"\t");**

indiriciler**[**arrayIndex**].**id **=** atoi**(**token**);**

tabIndex **=** 1**;**

**while(** token **!=** NULL **)** **{**

token **=** strtok**(**NULL**,** **"\t");**

**switch** **(**tabIndex**)**

**{**

**case** 1**:**

indiriciler**[**arrayIndex**].**x1 **=** strtof**(**token**,** NULL**);**

**break;**

**case** 2**:**

indiriciler**[**arrayIndex**].**y1 **=** strtof**(**token**,** NULL**);**

**break;**

**case** 3**:**

indiriciler**[**arrayIndex**].**x2 **=** strtof**(**token**,** NULL**);**

**break;**

**case** 4**:**

indiriciler**[**arrayIndex**].**y2 **=** strtof**(**token**,** NULL**);**

**break;**

**}**

tabIndex**++;**

**}**

arrayIndex**++;**

**}**

**}**

printf**("indiriciler islendi\n");**

**}**

arrayIndex **=** 0**;**

checker **=** strstr**(**ch**,** **"MUSTERI\_ID");**

**if(**checker **==** ch**){**

printf**("musteriler isleniyor\n");**

**while(true){**

fgets**(**ch**,**100**,**p**);**

checker **=** strstr**(**ch**,** **"END");**

**if(**ch **==** checker**){**

**break;**

**}** **else** **{**

token **=** strtok**(**ch**,** **"\t");**

musteriler**[**arrayIndex**].**id **=** atoi**(**token**);**

tabIndex **=** 1**;**

**while(** token **!=** NULL **)** **{**

token **=** strtok**(**NULL**,** **"\t");**

**switch** **(**tabIndex**)**

**{**

**case** 1**:**

musteriler**[**arrayIndex**].**tip **=** atoi**(**token**);**

**if(**musteriler**[**arrayIndex**].**tip **==** 1**)**tipMesken**++;**

**if(**musteriler**[**arrayIndex**].**tip **==** 2**)**tipTicari**++;**

**if(**musteriler**[**arrayIndex**].**tip **==** 3**)**tipSanayi**++;**

**break;**

**case** 2**:**

musteriler**[**arrayIndex**].**x1 **=** strtof**(**token**,** NULL**);**

**break;**

**case** 3**:**

musteriler**[**arrayIndex**].**y1 **=** strtof**(**token**,** NULL**);**

**break;**

**case** 4**:**

musteriler**[**arrayIndex**].**x2 **=** strtof**(**token**,** NULL**);**

**break;**

**case** 5**:**

musteriler**[**arrayIndex**].**y2 **=** strtof**(**token**,** NULL**);**

**break;**

**}**

tabIndex**++;**

**}**

arrayIndex**++;**

**}**

**}**

printf**("musteriler islendi\n");**

**}**

arrayIndex **=** 0**;**

checker **=** strstr**(**ch**,** **"HAT\_ID");**

**if(**checker **==** ch**){**

printf**("hatlar isleniyor\n");**

**while(true){**

fgets**(**ch**,**100**,**p**);**

checker **=** strstr**(**ch**,** **"END");**

**if(**feof**(**p**)** **||** checker **==** ch**){**

**break;**

**}** **else** **{**

token **=** strtok**(**ch**,** **"\t");**

hatlar**[**arrayIndex**].**id **=** atoi**(**token**);**

tabIndex **=** 1**;**

**while(** token **!=** NULL **)** **{**

token **=** strtok**(**NULL**,** **"\t");**

**switch** **(**tabIndex**)**

**{**

**case** 1**:**

hatlar**[**arrayIndex**].**x1 **=** strtof**(**token**,** NULL**);**

**break;**

**case** 2**:**

hatlar**[**arrayIndex**].**y1 **=** strtof**(**token**,** NULL**);**

**break;**

**case** 3**:**

hatlar**[**arrayIndex**].**x2 **=** strtof**(**token**,** NULL**);**

**break;**

**case** 4**:**

hatlar**[**arrayIndex**].**y2 **=** strtof**(**token**,** NULL**);**

**break;**

**}**

tabIndex**++;**

**}**

arrayIndex**++;**

**}**

**}**

printf**("hatlar islendi\n");**

**}**

**}**

**}** **else{**

printf**("Dosya Bulunamadi.");**

exit**(**1**);**

**}**

fclose**(**p**);**

FILE **\***p2 **=** fopen**("Sonuclar.txt","w");**

*//Kontrol edilmek istenilen musteri ve hat id'leri daha sonra dosyaya yazılmak uzere aliniyor*

**int** k**;**

**int** yanitMusteri**=**0**;**

**int** yanitHat**=**0**;**

printf**("Kontrol etmek istediginiz musteri ID'sini giriniz: ");**

scanf**("%d",&**yanitMusteri**);**

printf**("Kontrol etmek istediginiz hat ID'sini giriniz: ");**

scanf**("%d",&**yanitHat**);**

*//Bu dongu eger kullaniciden alinan id mevcut ise bilgilerini yazdiriyor*

**int** ayirac **=** 0**;**

**for(**k**=**0**;**k**<**musteriSayisi**;**k**++){**

**if(**musteriler**[**k**].**id **==** yanitMusteri**){**

**switch(**musteriler**[**k**].**tip**){**

**case** 1**:**

fprintf**(**p2**,"Musteri Tipi: MESKEN(1)\n");**

**break;**

**case** 2**:**

fprintf**(**p2**,"Musteri Tipi: Ticari(2)\n");**

**break;**

**case** 3**:**

fprintf**(**p2**,"Musteri Tipi: Sanayi(3)\n");**

**break;**

**}**

fprintf**(**p2**,"Istenilen musteri bilgileri: ID X1 Y1 X2 Y2\n");**

fprintf**(**p2**," %d %.2f %.2f %.2f %.2f\n",**musteriler**[**k**].**id**,**musteriler**[**k**].**x1**,**musteriler**[**k**].**y1**,**musteriler**[**k**].**x2**,**musteriler**[**k**].**y2**);**

ayirac **=** 1**;**

**break;**

**}**

**}**

**if(**ayirac**==**0**)**fprintf**(**p2**,"Girilen id ye ait musteri bulunmamaktadir...\n");**

*//İstenilen bilgiler olusturulan dosyaya yaziliyor*

fprintf**(**p2**,"Indirici Sayisi: %d\n",** indiriciSayisi**);**

fprintf**(**p2**,"Musteri Sayisi : %d\n",** musteriSayisi**);**

fprintf**(**p2**,"Hat Sayisi : %d\n",** hatSayisi**);**

fprintf**(**p2**,"\n");**

fprintf**(**p2**,"Mesken Tipi Musteri Sayisi: %d\n",**tipMesken**);**

fprintf**(**p2**,"Ticari Tipi Musteri Sayisi: %d\n",**tipTicari**);**

fprintf**(**p2**,"Sanayi Tipi Musteri Sayisi: %d\n",**tipSanayi**);**

fprintf**(**p2**,"\n");**

fprintf**(**p2**,"Indirici bilgileri: ID X1 Y1 X2 Y2\n");**

*//Indirici bilgileri yazdiriliyor//Birden fazla indirici durumu olmasına karsilik for ile yazildi*

**for(**k**=**0**;**k**<**indiriciSayisi**;**k**++){**

fprintf**(**p2**," %d %.2f %.2f %.2f %.2f\n",**indiriciler**[**k**].**id**,**indiriciler**[**k**].**x1**,**indiriciler**[**k**].**y1**,**indiriciler**[**k**].**x2**,**indiriciler**[**k**].**y2**);**

**}**

fprintf**(**p2**,"\n");**

*//Hatlarin toplam uzunlugu icin islemler*

**float** hatToplamUzunluk**=**0**;**

**int** i**;**

**for(**i**=**0**;**i**<**hatSayisi**;**i**++){**

hatToplamUzunluk **+=** mesafeHesaplayici**(**hatlar**[**i**].**x1**,**hatlar**[**i**].**x2**,**hatlar**[**i**].**y1**,**hatlar**[**i**].**y2**);**

**}**

fprintf**(**p2**,"Toplam Hat Uzunlugu: %f m'dir.",**hatToplamUzunluk**);**

fclose**(**p2**);**

printf**("Yazdirma islemi tamamlandi.");**

FILE **\***p3 **=** fopen**("Musteri\_kus\_ucusu\_mesafe.txt","w");**

**float** kusUcusuMesafe**=**0**;**

**int** a**=**0**;**

fprintf**(**p3**,"ID TIP KUS\_UCUSU\_MESAFE\n");**

**for(**i**=**0**;**i**<**musteriSayisi**;**i**++){**

kusUcusuMesafe **=** mesafeHesaplayici**(**indiriciler**[**a**].**x2**,**musteriler**[**i**].**x1**,**indiriciler**[**a**].**y2**,**musteriler**[**i**].**y1**);**

fprintf**(**p3**,"%d %d %f\n",**musteriler**[**i**].**id**,**musteriler**[**i**].**tip**,**kusUcusuMesafe**);**

**}**

fclose**(**p3**);**

printf**("\nKus Ucusu mesafeler yazdirildi...");**

**return** 0**;**

**}**

**float** mesafeHesaplayici**(float** x1**,** **float** x2**,float** y1**,float** y2**){**

**float** mesafe**=**0**;**

**float** araIslem**=**0**;**

araIslem **=** pow**(**x2**-**x1**,**2**)** **+** pow**(**y2**-**y1**,**2**)** **;**

mesafe **=** sqrt**(**araIslem**);**

**return** mesafe**;**

**}**