



INF 212 Algorithms and Programming II

**2020-2021 Spring
Electronic Engineering**

**Proje ID:
2**

**Project Name:
Pharmacy
Program**

School ID	Name	Surname
1901022022	Selimhan	Aygün

Program bir eczane programıdır. Bu program sayesinde sadece yönetici, eczane ve ilaç bilgilerini ekleyip düzenleyebilir. Programı kullanan diğer herkes ise ilaç ve eczaneler ile bilgileri görüntüleyebilir ve spesifik olarak eczane ve ilaç araması yapabilir.

Problem, arayüzler arasında geçişleri kolay olan kullanılabilir ve anlaşılabilir bir eczane programının yapılmasıdır. Dolayısıyla programı kullananın istediğinde girdiği bilgiyi saklamalı ve ihtiyaç olduğunda görüntülemelidir.

Bu kapsamda programa dahil edilmesi gereken şeyler dosya işlemleridir. Problemi çözmek için dosyaya yazma ve dosyadan okumanın düzgün bir şekilde yapılması gerekmektedir.

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>
#include <conio.h>
#include <cstdlib>
```

Tanımladığım kütüphanelerin hepsini bir yerlerde kullandım. Standart kütüphane olan <iostream>, dosya işlemleri yapmada <fstream>, string işlemlerinde ise <string> kütüphanesi işime yaradı. Diğer kütüphaneler ise daha çok dosyadaki yazıyı düzenleme ve arayüzü daha kullanılabilir hale getirmek için kullanıldı.

```
Eczane Programina Hosgeldiniz
```

- ```

1.Eczane Girisi
2.Musteri Girisi
3.Programi Sonlandir
```

```
Secmek istediginizi giriniz:
```

Programın ana menüsü bu şekilde gözüküyor. Eczane girişi seçilirse önceden tanımlanan şifre soruluyor Şifreyi **123** olarak tanımladım. Eğer şifre yanlışsa şifreyi tekrar soruyor doğru şifre girilene kadar başka bir işlem yapılamıyor.

```
Yonetici Menusune Hosgeldiniz
```

- ```
-----  
1.Eczane Listesi Olustur  
2.Eczane Ekle  
3.Ilac Ekle  
4.Ilac Duzenle  
5.Ana Menu
```

```
Secmek istediginizi giriniz:
```

Şifre doğru bir şekilde girildikten sonra yönetici menüsüne ulaşılabilir.

1.1'deki Eczane listesi oluştur eğer "eczane_listesi.dat" dosyası mevcut değilse dosyayı kodun mevcut konumunda sadece oluşturuyor ve "**Dosya basariyla olusturuldu**" çıktısını veriyor ve yönetici menüsüne geri dönüyor. Eğer dosya mevcut ise "**eczane dosyasi mevcut**" çıktısını veriyor ve yönetici menüsüne geri dönüyor. Bu kısımda if-else dışında dosya açma ve kontrolü için aşadaki işlemleri yaptım.

```
fstream file;  
file.open("eczane_listesi.dat", ios::in | ios::out);  
  
if(choice==1)  
{  
    if(file)  
    {  
        cout<<"Eczane dosyasi mevcut"<<endl;  
        file.close();  
    }  
  
    else{  
        file.close();  
        file.open("eczane_listesi.dat", ios::out | ios::in | ios::app);  
        cout<<"Dosya basariyla olusturuldu"<<endl;  
    }  
  
    goto flag;  
}
```

1.2'deki Eczane ekleme bölümünde ise class Eczane'nin içindeki eczane_olustur() fonksiyonu customer_menu()' de açılan bir Eczane objesiyle gönderiliyor. Teker teker değişkenler alındıktan sonra customer_ekle() adındaki class Eczane dışındaki bir fonksiyona parametreler ile gönderilip dosyaya yazdırılıyor. Sonrada dosyaya yazdırılan değişkenler ekrana bastırılıyor.

```
Eczane ID:      4
Eczane Adi:     Emel
Eczane Adresi:  Ferahevler
Eczane dosyaya eklendi
Eczane ID:      4
Eczane Adi:     Emel
Eczane Adresi:  Ferahevler

Baska eczane eklemek ister misiniz? (y/n) :
```

Setf ve setw komutları <iomanip> kütüphanesinde bulunan fonksiyonlardır. Setf'de yaptığım ios::left ile yazıları sola sabitleyip setw() ile aralarındaki boşluğu belirleyebiliyorum.

Cin ile class daki değerleri direkt alabiliyorum çünkü fonksiyonum class'ın içinde daha önce tanımlanmıştı. Ardından eczn_ekle'ye parametrelerle gönderip dosyayı açıp file<< yöntemiyle düzgün bir şekilde eczaneyi ekletiyorum.

```
void Eczane::eczn_olustur()
{
    cout.setf(ios::left);

    cout<<setw(18)<<"Eczane ID: ";
    cin>>eczn_id;

    cout<<setw(18)<<"Eczane Adi: ";
    cin>>eczn_isim;

    cout<<setw(18)<<"Eczane Adresi: ";
    cin>>eczn_adres;

    eczn_ekle(eczn_id, eczn_isim, eczn_adres);

    eczn_ilac_sayfasi();
}

void eczn_ekle(int eczn_id, string eczn_isim, string eczn_adres)
{
    fstream file;
    file.open("eczane_listesi.dat", ios::in | ios::out | ios::app);

    file<<setw(18)<<eczn_id
        <<setw(18)<<eczn_isim
        <<setw(18)<<eczn_adres
        <<endl;
}
```

```
void Eczane::eczn_ilac_sayfasi()
{
    fstream file2;
    file2.open(eczn_isim + "_ilac" + ".dat", ios::out | ios::in | ios::app);
    file2.close();
}
```

Yukarıdaki fonksiyonda görmüş olduğumuz eczn_ilac_sayfasi fonksiyonu eczn_olustur() fonksiyonu içinde class'a dair değerler alındıktan sonra direkt çağırılıp almış olduğumuz eczn_isim değişkeni ile _ilac.dat ı birleştirerek ilaçlara ait dosyayı oluşturuyor. Class içinde olduğu için ayrıca get_eczn_isim() fonksiyonunu kullanmaya gerek kalmıyor.

```
ID          ECZANE ADI      ADRES
-----
1           Ayca      Maslak
2           Denizci    Istinye
3           Derman     Kirecburnu
4           Emel       Ferahevler
Ilac eklemek istediginiz eczane adini giriniz
Emel

Ilac ID:      5

Ilac Adi:     Parol

Ilac Fiyati:  7

Ilac Sayisi:  30
Ilac dosyaya eklendi
Ilac ID:      5
Ilac Adi:     Parol
Ilac Fiyati:  7
Ilac Sayisi:  30

Baska ilac eklemek ister misiniz? (y/n) :
```

1.3'deki ilaç ekleme bölümünde ise önce var olan eczaneleri gösteren bir liste çıkıyor. Ardından kullanıcıya ilaç eklenmek istediği eczaneyi soruyor. Eczane ismi düzgün bir şekilde girildiğinde ise ilaca ait verileri teker teker alıyor. Ardından dosyaya eklendi bilgisini verip dosyaya yazılan verileri gösteriyor.

Kod üstteki kodu ise üstteki kod ile büyük çoğunlukla aynı.

1.4'deki ilaç düzenle kısmında ise yine dosyadaki eczanelerin listesi çıkıyor ve ilaç listesinde değişiklik yapılacak eczane kullanıcıya soruluyor. Eğer kullanıcı doğru bir şekilde ismi girerse eczaneye ait ilacın listesi ve bilgisini değiştirmek istediği ilacın ID'si soruluyor. Kullanıcı ID'ye listeden bakıp değiştirmek ilacın

ID'sini yazdıktan sonra ise yeni bilgiler kullanıcıdan isteniyor. Kullanıcı bilgileri bitirdikten sonra bilgilerin başarıyla değiştirildiğine dair bilgi verilip dosyanın yeni hali gösteriliyor.

```
ID          ECZANE ADI      ADRES
-----
1           Ayca        Maslak
2           Denizci     Istinye
3           Derman      Kirecburnu
4           Emel        Ferahevler

Ilac listesinde degisiklik yapmak istediginiz eczaneyi giriniz:
Denizci
ILAC ID      ILAC ADI      ILAC FIYATI    ILAC SAYISI
-----
24           Majezik      16             84
5            Parol        6              43
21           Stimpack     80             5
Bilgilerimi degistirmek istediginiz ilacin ID'sini giriniz: 24

Yeni ilac ID giriniz : 24

Yeni ilac ismi giriniz: Majezik

Yeni fiyat bilgisi giriniz : 17

Yeni sayi bilgisi giriniz : 100
Duzenleme islemi basarili
-----
ILAC ID      ILAC ADI      ILAC FIYATI    ILAC SAYISI
-----
24           Majezik      17            100
5            Parol        6             43
21           Stimpack     80             5
```



Denizci_ilac - Not Defteri

Dosya	Düzen	Biçim	Görünüm	Yardım
		24	Majezik	17
		5	Parol	6
		21	Stimpack	80

ilac_duzenle(); fonksiyonunu get ve setlerle uğraşmamak için Ilac class'ının içine koydum. Admin_menu(); kısmında eczane ismini alıyorum ve ilac_duzenle fonksiyonuna parametre ile gönderiyorum.

```

else if(choice==4)
{
    system("CLS");
    eczaneleri_goster();
    cout<<"\nİlac listesinde degisiklik yapmak istediginiz eczaneyi giriniz: "<<endl;
    cin>>eczane_ad;
    ilac.ilac_listesi(eczane_ad);
    ilac.ilac_duzenle(eczane_ad);
    cout<<"Duzenleme islemi basarili"<<endl;
    cout<<"-----"<<endl;
    ilac.ilac_listesi(eczane_ad);
    goto flag;
}

```

Class içinde kullandığım için değişkenlere erişmek için a adında bir nesne tanımlıyorum. Ayrıca kullanıcıya değiştirmek istediği ilacın id'sini soracağım için bir de integer tipinde id adında bir değişken tanımladım.

Dosyayı okuyacağımız için ifstream kullanıp parametreden aldığımız isim ile dosyayı okuma işlemi başlattım. Parametre ile gelen eczane isminin doğruluğunu kontrol etmek için if(file) kullandım true döndürürse düzenleme işlemlerini gerçekleştirecek. Ayrıca burada dosyayı düzenlerken verileri kaybetmemek için temp.dat adında bir dosya açıyoruz. Önce aramadığımız ilaç verilerini sırayla bu temp dosyasına geçireceğiz sonra düzenlemek istediğimiz verileri kullanıcıdan alıp temp'e atayacağız bunların hepsini while ve if statement kullanarak yapacağımız için aradığımız verinin hangi satırda olmasının bir önemi olmayacak.

```

void Ilac::ilac_duzenle(string eczn_isim)
{
    Ilac a;
    int id;
    ifstream file(eczn_isim + "_ilac" + ".dat");

    if(file)
    {

        cout<<"Bilgilerimi degistirmek istediginiz ilacin ID'sini giriniz: ";
        cin>>id;

        fstream file2;
        file2.open("temp.dat",ios::out | ios::in | ios::app);
    }
}

```

While ile daha önceden girilen tüm değişkenler sıra ile okunuyor. file>> kullanmamın sebebi daha önceden dosyaya file<< ile yazdığımız değişkenleri aynı şekilde sıra ile okuması. While içinde file>>a.ilac_id kullanmamın sebebi ise her satırın başındaki ilac_id'nin, var olmayana kadar yani false olana kadar devam etmesini istememden kaynaklı (!file.eof kullanınca son satırı bir daha yazdırdığından dolayı kullanmadım).

Girilen id'yi dosyadaki id ile kontrol eden if(a.ilac_id!=id) kısmında eğer aradığı id'yi bulamaz ise diğerlerini değiştirmeyip direkt olarak açmış olduğumuz temp dosyasına atıyor. Bulduğu zaman ise kullancıdan yeni veriler istiyor ve onları temp dosyasına atıyor.

```

while(file>>a.ilac_id)
{
    file>>a.ilac_isim;
    file>>a.ilac_fiyati;
    file>>a.ilac_sayisi;

    if(a.ilac_id!=id)
    {
        file2<<setw(15)<<a.ilac_id
            <<setw(15)<<a.ilac_isim
            <<setw(15)<<a.ilac_fiyati
            <<setw(15)<<a.ilac_sayisi
            <<endl;
    }
    else if(a.ilac_id==id){
        cout.setf(ios::left);
        cout<<setw(15)<<"\nYeni ilac ID giriniz : ";
        cin>>a.ilac_id;
        cout<<setw(15)<<"\nYeni ilac ismi giriniz : ";
        cin>>a.ilac_isim;
        cout<<setw(15)<<"\nYeni fiyat bilgisi giriniz : ";
        cin>>a.ilac_fiyati;
        cout<<setw(15)<<"\nYeni sayı bilgisi giriniz : ";
        cin>>a.ilac_sayisi;
        file2<<setw(15)<<a.ilac_id<<setw(15)<<a.ilac_isim<<setw(15)<<a.ilac_fiyati<<setw(15)<<a.ilac_sayisi<<endl;
    }
}

```

İşlemimizi bitirdikten sonra dosyaları kapatıyoruz ve eski verilerin olduğu ilk dosyayı siliyoruz. Ardından ilk dosyanın adını temp dosyasına veriyoruz.

```

file.close();
file2.close();
remove((eczn_isim + "_ilac" + ".dat").c_str());
rename("temp.dat", (eczn_isim + "_ilac" + ".dat").c_str());
}

```

if(file) false döndürürse aşağıdaki çıktıyı vermesi için else kısmına bunu ekledim.

```

else
{
    cout<<"Bu isimde bir eczane bulunamadı!"<<endl;
}

```

```
Musteri Menusune Hosgeldiniz
*****

1.Ilac Ara
2.Eczane Listele
3.Herhangi Bir Eczaneyi Goster
4.Herhangi Bir Eczaneye Ait Ilac Listesi
5.Ana Menu

Secmek istediginizi giriniz:
```

2.1'deki ilaç arama kısmında ise önce aramak istenilen ilacın adı soruluyor. `ilac_ara()` fonksiyonunu `Ilac` sınıfında tanımladığım için nesneyi ve aranmak istenen ismi parametreye yazarak fonksiyonu çağırıyorum.

```
if(choice==1)
{
    system("cls");
    flag2:
    cout<<"\n\n Aramak istediginiz ilacin ismini yaziniz: ";
    cin>>ilac_adi;
    ilc.ilac_ara(ilac_adi);

    cout<<"\n\nBir daha ilac aramak ister misiniz? (y/n) : ";
    ch = getche();

    if(ch=='y' || ch=='Y')
        goto flag2;
    else
        goto flag;
}
```



```

void Ilac::ilac_ara(string ilac_adi)
{
    Eczane A;

    fstream file;
    fstream file2;
    int counter=0;
    int eczn_id=A.get_eczn_id();
    string eczn_isim=A.get_eczn_isim();
    string eczn_adres=A.get_eczn_adres();
    file.open("eczane_listesi.dat", ios::in);

```

ilac_ara() fonksiyonu Ilac sınıfında bulunduğundan dolayı private olan Eczane verilerine direkt olarak erişemiyorum. Bu yüzden **Eczane A** nesnesi tanımladım. Bu nesneyi ve get fonksiyonlarını kullanarak hepsini bir değişkene tanımladım çünkü değişkene tanımlamadan while içinde düzgün bir şekilde kullanılamıyor. Ayrıca 2 farklı file kullanacağımız için 2 farklı file tanımladım. Bunlardan ilki olan "eczane_listesi.dat" ı okunmak üzere açtım.

```

while(file>>eczn_id)
{
    file>>eczn_isim;
    file>>eczn_adres;

    file2.open(eczn_isim + "_ilac" + ".dat", ios::in);

    while(file2>>ilac_id)
    {
        file2>>ilac_isim;
        file2>>ilac_fiyati;
        file2>>ilac_sayisi;

        if(ilac_isim.compare(ilac_adi)==0){
            cout<<"\nİlacın Bulunduğu Eczane Bilgileri: "<<endl;
            cout<<"-----"<<endl;
            cout<<setw(15)<<"\nEczane ID: " <<eczn_id
                <<setw(15)<<"\nEczane Adi: " <<eczn_isim
                <<setw(15)<<"\nEczane Adres: " <<eczn_adres<<endl;

            cout<<"\nAradığınız İlac Bilgileri: "<<endl;
            cout<<"-----"<<endl;
            ilac_goster();
            counter++;
        }

    }
    file2.close();
}
file.close();
if(counter==0){
    cout<<"Girmis olduğunuz ilac hiçbir eczanede bulunamamıştır.\n"
    <<"Lutfen büyük-küçük harf uyumunu kontrol edip tekrar deneyiniz"<<endl;
}

```

Eczane_listesi dosyasında her satırın okunduğunda o eczaneye ait listeyi okutmak üzere açtım. Parametre ile girilen ilaç adını compare ile kıyasladım. Aynı olduğunda ise eczane bilgilerini ve ilacın bilgilerini ekrana bastırdım. Bunun için daha önceden yazmış olduğum ilac_goster() fonksiyonunu kullandım. İlac_ara fonksiyonu class içinde olduğundan dolayı dosyadan okunduğunda çağırılan ilac_goster() fonksiyonu okunan verileri başarıyla ekrana basabiliyor.

Counter ise döngü if'e girdiğinde artıyor. Başlangıç değeri sıfıra eşitse hiçbir türlü if'e girmediğinden dolayı "tekrar deneyiniz" çıktısı vermesi için döngü dışında kontrol ediliyor ve doğruysa çıktıyı basıyor.

```

void Ilac::ilac_goster()
{
    cout.setf(ios::left);

    cout<<setw(15)<<"Ilac ID: "<<ilac_id<<endl;
    cout<<setw(15)<<"Ilac Adi: "<<ilac_isim<<endl;
    cout<<setw(15)<<"Ilac Fiyati: "<<ilac_fiyati<<endl;
    cout<<setw(15)<<"Ilac Sayisi: "<<ilac_sayisi<<endl;
}

```

Çıktı görüldüğü üzere doğru bir şekilde sonuç veriyor.

```

Aramak istediginiz ilacin ismini yaziniz: Xanax

Ilacin Bulundugu Eczane Bilgileri:
-----

Eczane ID: 3
Eczane Adi: Derman
Eczane Adres: Kirecburnu

Aradiginiz Ilac Bilgileri:
-----

Ilac ID:      72
Ilac Adi:     Xanax
Ilac Fiyati:  23
Ilac Sayisi:  18

Bir daha ilac aramak ister misiniz? (y/n) :

```

2.2'deki Eczane listeleme kısmında ise eczaneleri_goster() fonksiyonu çağırılarak var olan eczaneler listeleniyor.

ID	ECZANE ADI	ADRES
1	Ayca	Maslak
2	Denizci	Istinye
3	Derman	Kirecburnu
4	Emel	Ferahevler

Bunu da göstermek adına class dışında aldım. Bu yüzden verileri okutmak üzere get fonksiyonu ile teker teker alıp değişkene tanımlayıp öyle okutuyorum. Counter'ı estetik bir görüntü elde etmek için kullandım.

```
void eczaneleri_goster()
{
    Eczane A;
    fstream file;
    int counter=0;
    int eczn_id=A.get_eczn_id();
    string eczn_isim=A.get_eczn_isim();
    string eczn_adres=A.get_eczn_adres();
    file.open("eczane_listesi.dat", ios::in);

    while(file>>eczn_id)
    {
        file>>eczn_isim;
        file>>eczn_adres;

        if(counter==0){
            cout.setf(ios::left);
            cout<<setw(15)<<"ID"<<setw(15)<<"ECZANE ADI"<<setw(15)<<"ADRES"<<endl;
            cout<<"-----"<<endl;
            counter++;
        }
        cout<<setw(15)<<eczn_id<<setw(15)<<eczn_isim<<setw(15)<<eczn_adres<<endl;
    }
    file.close();
}
```

2.3'deki Herhangi Bir Eczaneyi Göster kısmında ise yine class içinde tanımladığım eczn_ara() fonksiyonunu nesne ve parametre ile çağırıyoruz.

```
else if(choice==3)
{
    flag3:
    cout<<"Aramak Istediginiz Eczane Adini Giriniz: "<<endl;
    cin>>eczn_adi;
    e.eczn_ara(eczn_adi);

    cout<<"\n\nBir daha eczane aramak ister misiniz? (y/n) : ";
    ch = getch();

    if(ch=='y' || ch=='Y')
        goto flag3;
    else
        goto flag;
}
```

Kullanıcıdan aldığımız ismi doya okurken kontrol ediyor eğer isim aynı ise eczaneleri gösteriyor.

```
void Eczane::eczn_ara(string isim)
{
    fstream file;
    int counter=0;
    file.open("eczane_listesi.dat", ios::in);

    while(file>>eczn_id)
    {

        file>>eczn_isim;
        file>>eczn_adres;

        if(eczn_isim.compare(isim)==0)
        {
            eczn_goster();
            counter++;
        }

    }
    if(counter==0)
        cout<<"Boyle bir eczane bulunamamistir"<<endl;
    file.close();
}
```

Çıktı ise bu şekilde

```
Aramak Istediginiz Eczane Adini Giriniz:
Ayca
Eczane ID:      1
Eczane Adi:     Ayca
Eczane Adresi:  Maslak

Bir daha eczane aramak ister misiniz? (y/n) :
```

2.4'deki Herhangi Bir Eczaneye İlaç Listesi ise İlac sınıfına ait nesne ve kullanıcıdan o an alınan parametre ile ilac_listesi() fonksiyonunu çağırıyoruz. Parametreden gelen isimle oluşan bir dosyanın var olup

olmadığını kontrol etmek üzere if(file) kullandım. Doğru olduğu durumlarda ise değişkenleri ekrana yazdırdım.

```
void Ilac::ilac_listesi(string isim)
{
    int counter=0;

    ifstream file(isim + "_ilac" + ".dat");

    if(file)
    {
        while(file>>ilac_id)
        {
            file>>ilac_isim;
            file>>ilac_fiyati;
            file>>ilac_sayisi;

            if(counter==0) {
                cout.setf(ios::left);
                cout<<setw(15)<<"ILAC ID"<<setw(15)<<"ILAC ADI"<<setw(15)<<"ILAC FIYATI"<<setw(15)<<"ILAC SAYISI"<<endl;
                cout<<"-----"<<endl;
                counter++;
            }

            cout<<setw(15)<<ilac_id<<setw(15)<<ilac_isim<<setw(15)<<ilac_fiyati<<setw(15)<<ilac_sayisi<<endl;

        }
    }
    else{
        cout<<"Bu isimde bir eczane bulunamadı!"<<endl;
    }
}
```

```
else if(choice==4)
{
    flag4:
    cout<<"\nIlac Listesine Bakmak Istedginiz Eczane Adini Giriniz: "<<endl;
    cin>>eczn_adi;
    ilc.ilac_listesi(eczn_adi);

    cout<<"\n\nBaska liste aramak ister misiniz? (y/n) : ";
    ch = getche();

    if(ch=='y' || ch=='Y')
        goto flag4;
    else
        goto flag;
}
else if(choice==5)
```

Çıktı bu şekilde

Ilac Listesine Bakmak Istedginiz Eczane Adini Giriniz:

Ayca

ILAC ID	ILAC ADI	ILAC FIYATI	ILAC SAYISI
---------	----------	-------------	-------------

5	Parol	7	50
5	Parol	7	50
5	Anan	54	6
5	Parol	6	32

Baska liste aramak ister misiniz? (y/n) :

Genel olarak programın işleyişi bu şekildedir. Read ve write komutlarını benim yaptığıma göre fazla zahmetli olduğu için kullanmadım. Bu tarz hem programcı tarafından anlaşılır hem de kullanıcı dostu bir program olduğunu düşünüyorum.

Set fonksiyonlarını kullanabileceğim düzgün bir yer bulamadığım için yazmadım. Çünkü var olan fonksiyonlarımız zaten sınıfın içinde olduğu için private kısımlara kolaylıkla erişilebiliyordu.

Bazı fonksiyonları da ulaşması daha kolay olduğu için sınıfın içine yazdım sınıfın içine yazmadığım örnekler de (eczaneleri_goster()) kodumda mevcuttur.

Constructor'ları girdim ama kullanacak bir yer bulamadığım için kullanmadım. Programda yanlış bir kullanım yaptığımda default constructor'ın çalıştığını gözlemledim.

Hata kısmında ise

-Private fonksiyonlara yanlış bir şekilde erişmeye çalıştığımda çıkan hatalar

-While içinde eof kullanılınca son satırın iki kez görüntülenmesi

-değişken adıyla dosya oluşturmada çıkan string hatası

-Syntax hataları

ile karşılaştım.

REFERANSLAR:

<https://www.cplusplus.com/doc/tutorial/files/>

https://www.tutorialspoint.com/cplusplus/cpp_files_streams.htm

<https://stackoverflow.com/questions/21589353/cannot-convert-stdbasic-stringchar-to-const-char-for-argument-1-to-i/21589609>

<https://stackoverflow.com/questions/21647/reading-from-text-file-until-eof-repeats-last-line>

<http://www.tutorialdost.com/Cpp-Programming-Tutorial/62-Cpp-File-Handling-IO-Read-Write-Object.aspx>