05/01/2021

Yapısal Programlama Dönem Projesi

“If a man wishes to pursue happiness, then he ought never to use scanf”

-Unknown

İçerik

Yapılan Bazı Tercihler ve Sebepleri

Yapılan tercihlerin ilki için Dr.Öğr.Üyesi Yunus Emre SELÇUK hocanın önerisi doğrultusunda hareket edilmiştir.

Silme ve güncelleme işlemleri için geçici bir dosya kullanmak

fseek, strlen gibi fonksiyonların bazı platformlarda farklı sonuçlar vermesinden dolayı kodun farklı platformlarda doğru bir şekilde çalışmasını sağlayan yollardan biri olan bu metod hocanın önerisi doğrultusunda tercih edilmiştir.

Kullanıcı sistemi tasarlamak

Bazı işlemlerin (eklenen dersleri listelemek, ders bırakmak, vb.) belli bir kişi tarafından yapılması gerektiğinden programda üç farklı kullanıcı türü tanımlıdır. Bu kullanıcı türleri şu şekildedir:
Admin: Hoca, kurs ve öğrenci için ekleme, silme ve güncelleme işlemlerinin yanında kredi limiti, ders sayısı limiti güncelleme gibi işlemleri de yapabilir.
Hoca: Tanımlanan kursları listeleme ve onların bazı detaylarını değiştirme yetkisine sahip olan bu kullanıcı, aynı zamanda ona tanımlı olan bir kursun öğrencilerini de bir dosyaya yazdırabilir.
Öğrenci: Ders ekleme, bırakma yetkilerinin yanında aldığı veya alabileceği dersleri listeleyebilir.

Veritabanı bilgileri dosyasını oluşturmak

Kredi limiti, ders sayısı gibi bilgileri her seferinde kullanıcıya sormak yerine saklı tutulacağı “db_info.txt” isimli bir dosya oluşturulmuştur. Dosyanın içinde veritabanının içerdiği öğrenci, hoca, kurs ve log sayıları da saklanmaktadır.

fscanf, scanf ve feof fonksiyonlarını kullanmamak

scanf ve fscanf fonksiyonlarının ‘\n’ ve ‘\r’ gibi karakterleri okumamalarından kaynaklı kirlenen bufferın temizlenmesi gerekir. fflush gibi fonksiyonlar bazı platformlarda istenen sonucu vermediğinden bufferın kirlenmesini önleyecek şekilde fgets fonksiyonu kullanılmıştır. Eldeki string üzerinde yapılan işlemlerle okunan ‘\n’ karakteri (buffer için bir etki yaratmadan) daha sonra karakter dizisinden temizlenebilir. feof fonksiyonu da linux tabanlı işletim sistemlerinde istenen sonucu döndürmediğinden fgets fonksiyonun dosya veya buffer bitiminde döndürdüğü NULL değerine göre hareket edilmiştir.

Şifreleme Fonksiyonu

Kullanıcı adları ve şifreler basit bir şifreleme yöntemiyle saklanmaktadır. “users.txt” isimli dosyada tutulan satırların her biri kullanıcının türünü, adını ve şifresini saklar. Şifre oluşturduğunda saklanan yazıların sırası ve karakter uzunlukları şu şekildedir:

Kullanıcı adı uzunluğu: ‘!’ karakterinin değerine magic number ve kullanıcı adının karakter sayısı eklenir.
Kullanıcı türü: ‘[‘ karakterinin değerine magic number ve kullanıcı türünü belirten sayı eklenir.
Magic number: 1-5 arasında seçilen rastgele bir sayının ‘:’ karakterinin değerine eklenmesiyle oluşur.
Kullanıcı adı: kullanıcı adını oluşturan karakterlere magic number eklenir.
Şifre: şifreyi oluşturan karakterlere magic number eklenir.

Örnek olarak 19011013 kullanıcı adlı p@ssWord şifreli bir kullanıcıyı ele alalım. Oluşan magic number 1 olsun. Anlatılan işlemler uygulanınca oluşan şifreli yazı şu şekilde olur:
Kullanıcı adı uzunluğu: 8 + 1 + ‘!’ = ’*’
Kullanıcı türü: 3 + 1 + ‘[‘ = ‘_’
Magic number: 1 + ‘:’ = ‘;’

Programın Test Edildiği Ortamlar

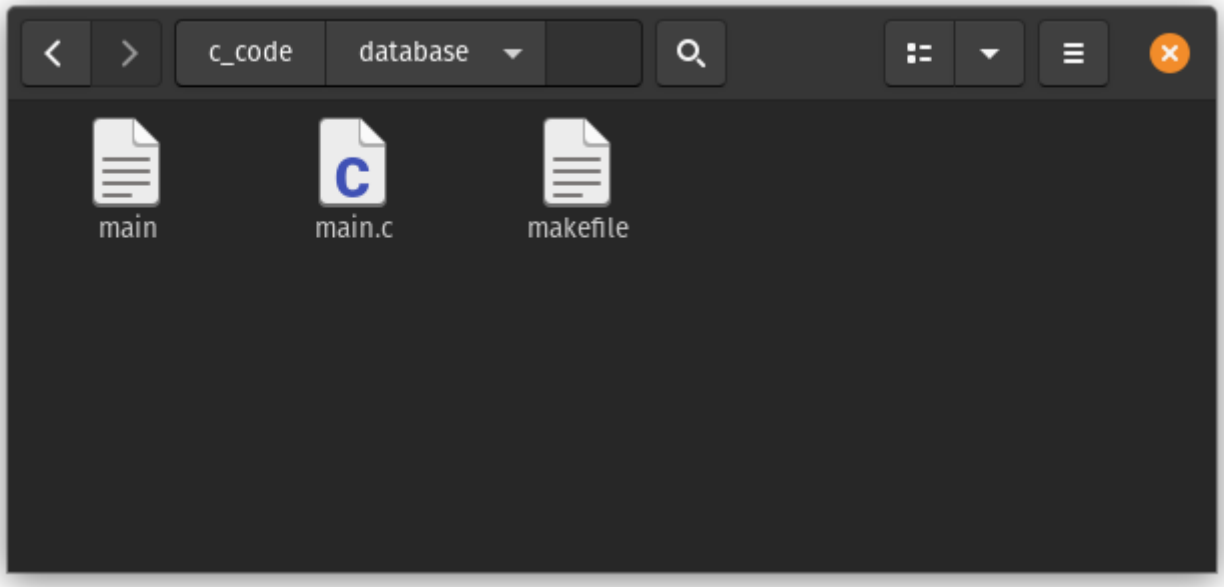
	İşletim Sistemi	Compiler
1	GNU / Linux Tabanlı Pop!_OS	gcc
2	MS Windows	gcc
3	macOS Big Sur	gcc

Program ayrıca Valgrind aracıyla test edilmiş olup yapılan hiçbir testte bellek sızıntısı tespit edilmemiştir. Valgrind aracı kullanılarak uzun bir test sonrasında programın bitimindeki çıktı:

```
==12204== HEAP SUMMARY:
==12204==  in use at exit: 0 bytes in 0 blocks
==12204==   total heap usage: 1,634 allocs, 1,634 frees, 976,555 bytes
allocated
==12204==
==12204== All heap blocks were freed -- no leaks are possible
==12204==
==12204== For lists of detected and suppressed errors, rerun with: -s
==12204== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Olumlu Durum Testleri ve Kullanım Kılavuzu

Bu bölümde programda kodlanan işlemlerin nasıl gerçekleştirilebilecekleri anlatılmıştır. Test sürecinde bazı uç durumlar (öğrenci bir dersi almışken o dersin silinmesi kredisini ve ders sayısını etkiler, olmayan bir hocanın derse eklenmesi, vb.) ve olumsuz durumlar (yanlış şifre girişi, kapasite veya kredi sınırına ulaşmış bir dersi eklemek, vb.) için kapsamlı testler yapılmış olup raporun daha uzun olmamasını sağlamak için ayrıca belirtilmemiştir.
Her işlem ilgili kullanıcı başlığının altında sunulmuştur. Kullanıcı girişi turuncu renkle gösterilmiştir.
Program ilk çalıştırıldığında bulunduğu klasörün durumu ayrıca klasöre yeni dosya ekleyen adımlarların ardında bir ekran alıntısı ile gösterilecektir. Klasörün programın ilk çalışması öncesindeki durumu:



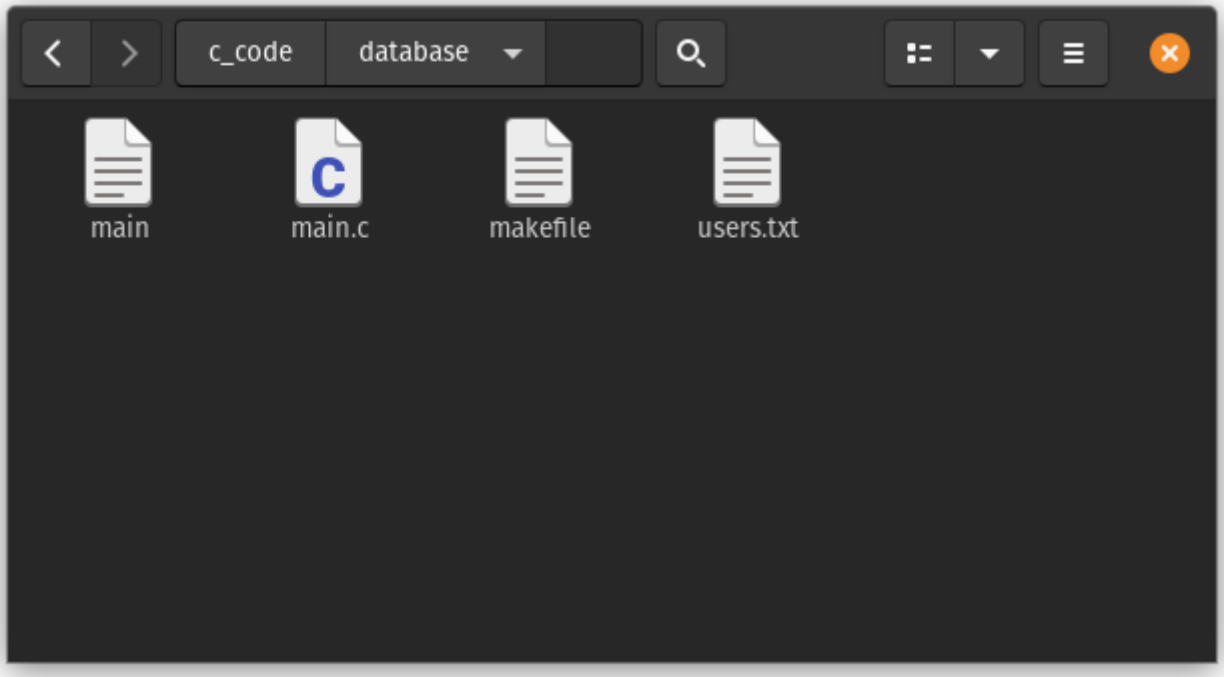
Giriş

Programı çalıştırdıktan sonra kullanıcı türü seçilerek kullanıcı adı ve şifre girilir. Kullanıcıya ait bir şifre yoksa ve ilgili dosyada kayıtlı ise yeni bir şifre oluşturması istenir. Admin hesabı için programı denerken bir sorun yaşanmaması için her yeni bir kullanıcı ismi girildiğinde şifresi yok ise kontrol yapılmadan yeni bir şifre istenir.

- Admin kullanıcısı belirleyeceği bir **kullanıcı adı** ile giriş yapabilir.
- Öğrenci kullanıcısı **öğrenci numarası** ile giriş yapabilir.
- Hoca kullanıcısı **hoca ID** ile giriş yapabilir.

```
Choose user type or type 'e' to exit
[1] Admin, [2] Instructor, [3] Student: 1
Username: admin
Password:
Hi admin. Create a password: admin
Success. Log in using your password.
```

“users.txt” dosyası oluşturuldu.



Dosyanın içeriği:

```
';benjobenjo
```

Admin

Admin kullanıcısının karşılaştığı ilk menüde yapmak istediği işlemin numarasını girerek ilgili alt menüye geçebilir.

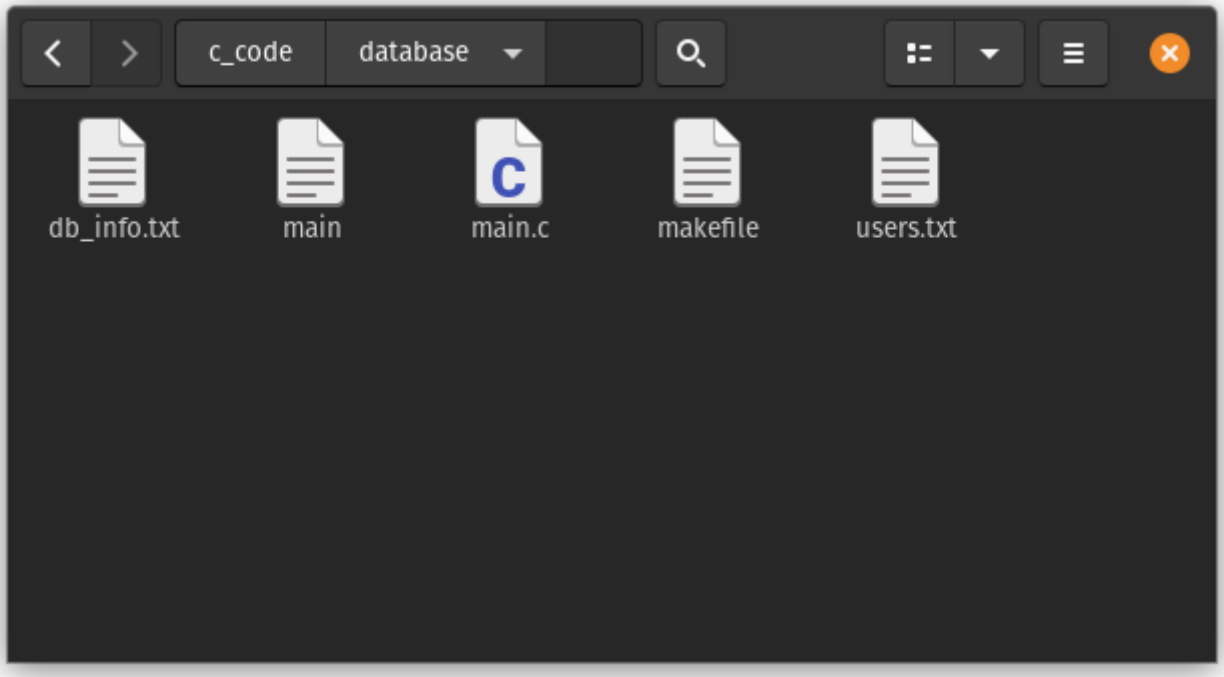
```
Welcome admin!
[1] Change settings
[2] Instructor operations
[3] Student operations
[4] Course operations
[5] Log out
Select from above [1-5]:
```

Ayarları değiştirmek

Bu menüde admin kullanıcısı kredi limiti ve ders sayısı limiti gibi ayarları değiştirebilir ve veritabanında saklanan verilerin sayısı hakkında bilgi edinebilir.

```
The database has:
0 courses, 0 instructors, 0 students, 0 enrollment logs
Current settings:
Credit limit: 30
Lesson Number Limit: 10
[1] Change settings
[2] Back
Select from above [1-2]: 1
```

“db_info.txt” dosyası oluşturuldu.



Dosyanın içeriği:

```
30,10,0,0,0,0
```

Ayar güncelleme adımı:

```
Already enrolled students won't be affected!
Press Enter to keep (old value)
(30) Enter new Credit limit : 20
(10) Enter new  Lessons number limit :
Updating as: 20,10,0,0,0,0
Updated 30 successfully
```

Güncelleme sonrası dosya içeriği:

```
20,10,0,0,0,0
```

Hoca işlemleri

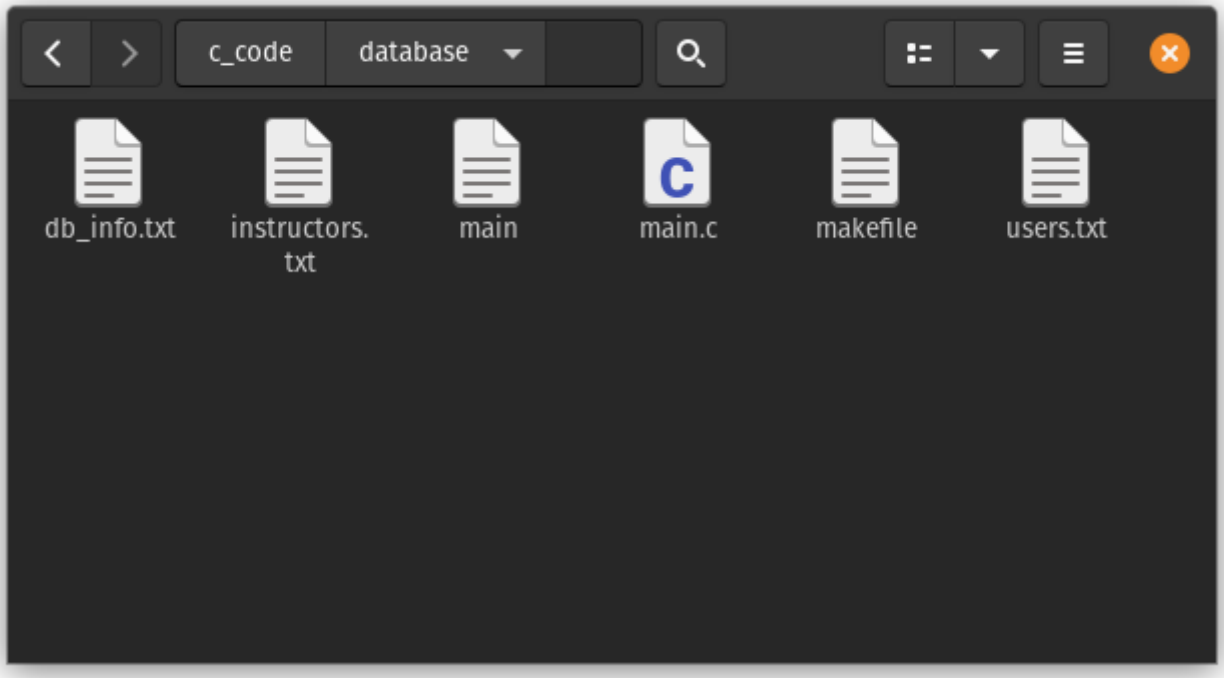
Bu menüde admin kullanıcısı hoca listeleme, ekleme, güncelleme ve silme gibi işlemleri yapabilir.

```
[1] List instructors
[2] Add an instructor
[3] Update an instructor
[4] Remove an instructor
[5] Back
Select from above [1-5]:
```

Hoca ekleme işlemi:
İşlem ilk defa yapıldığından yeni bir dosya oluşturulacaktır.

```
Enter instructor details:
ID: yselcuk
Name: Yunus Emre
Surname: SELCUK
Title: Dr
Warning: Couldn't open the file instructors.txt.
(The following WILL delete any existing file with the same name)
Should I (re)create it? (y/n): y
info: Created instructors.txt successfully
```

"instructors.txt" dosyası oluşturuldu.



Dosya içeriği:

```
yselcuk,Yunus Emre,SELCUK,Dr
```

Hoca güncelleme işlemi:

```
Enter instructor's ID: yselcuk
Press Enter to keep (old value)
(Yunus Emre) Enter new  Name :
(SELCUK) Enter new  Surname : Selcuk
(Dr) Enter new  Title: Dr.
Updating as: yselcuk,Yunus Emre,Selcuk,Dr.
Updated yselcuk successfully
```

Dosya içeriği:

```
yselcuk,Yunus Emre,Selcuk,Dr.
```

Hoca silme işlemi:
Hoca ekleme adımı birden fazla kez tekrarlanırsa ve sahte adıyla bir kullanıcı oluşturulursa.
Dosya içeriği:

```
aelbir,Ahmet,Elbir,Dr.
diri,Banu,Diri,Prof.Dr.
fcakmak,Furkan,Cakmak,Lect.
gbiricik,Goksel,Biricik,Dr.
hoilhan,Hamza Osman,Ilhan,Dr.
irem,Hafiza Irem,Turkmen,Dr.
sahte,not real,really,Fake
yselcuk,Yunus Emre,Selcuk,Dr.
```

Oluşan uyarılar kurs dosyasının var olmamasından kaynaklı olup işlem başarılı bir şekilde gerçekleşmiştir. Kullanıcıya ait bir şifre olmadığından kullanıcılar dosyasından silinmeyeceği belirtilmiştir.

```
Enter instructor's ID: sahte
Couldn't open file. Terminating the current operation with failure. (Error
Code: 67)
Deleted sahte successfully
User won't be deleted as they don't exist.
```

Değişim sonrası dosya içeriği:

```
aelbir,Ahmet,Elbir,Dr.
diri,Banu,Diri,Prof.Dr.
fcakmak,Furkan,Cakmak,Lect.
gbiricik,Goksel,Biricik,Dr.
hoilhan,Hamza Osman,Ilhan,Dr.
irem,Hafiza Irem,Turkmen,Dr.
yselcuk,Yunus Emre,Selcuk,Dr.
```

Hoca listeleme işlemi:

```
ID - Name - Surname - Title
aelbir - Ahmet - Elbir - Dr.
diri - Banu - Diri - Prof.Dr.
fcakmak - Furkan - Cakmak - Lect.
gbiricik - Goksel - Biricik - Dr.
hoilhan - Hamza Osman - Ilhan - Dr.
irem - Hafiza Irem - Turkmen - Dr.
yselcuk - Yunus Emre - Selcuk - Dr.
    === END OF DATA ===
```

Öğrenci İşlemleri

Bu menüde admin kullanıcısı öğrenci listeleme, ekleme, güncelleme ve silmenin yanında kayıt geçmişini listeleme işlemlerini yapabilir.

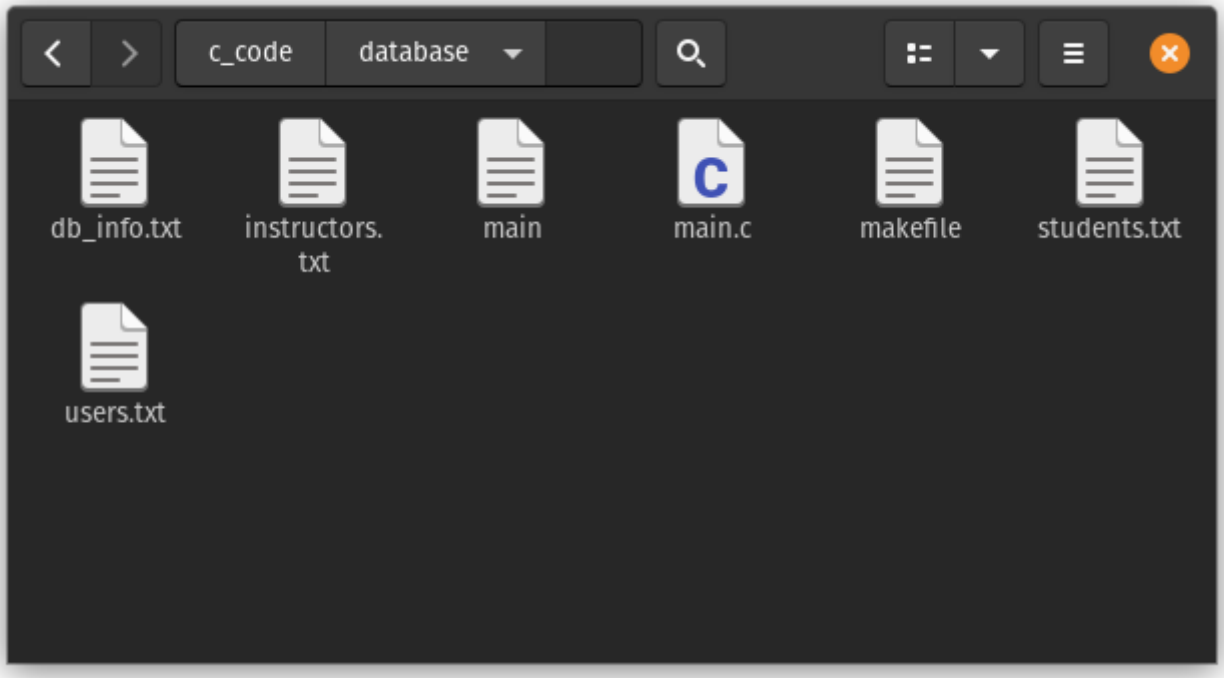
```
[1] List students
[2] Add a student
[3] Update a student
[4] Remove a student
[5] Print enrollment log
[6] Back
Select from above [1-6]:
```

Öğrenci ekleme işlemi:

İşlem ilk defa yapıldığından yeni bir dosya oluşturulacaktır.

```
Enter student details:
Number: 19011013
Name: MHD Besher
Surname: Alkurdi
Warning: Couldn't open the file students.txt.
(The following WILL delete any existing file with the same name)
Should I (re)create it? (y/n): y
info: Created students.txt successfully
```

"students.txt" dosyası oluşturuldu.



Dosya içeriği:

```
19011013,MHD Besher,Alkurdi,0,0
```

Öğrenci güncelleme işlemi:

```
Enter student details:
Number: 19011013
Press Enter to keep (old value)
(MHD Besher) Enter new  Name :
(Alkurdi) Enter new  Surname : Al kurdi
Updating as: 19011013,MHD Besher,Al kurdi,0,0
Updated 19011013 successfully
```

Dosya içeriği:

```
19011013,MHD Besher,Al kurdi,0,0
```

Öğrenci silme işlemi:

Öğrenci ekleme adımı birden fazla kez tekrarlanırsa ve 18036011 numaralı bir öğrenci oluşturulursa.
Dosya içeriği:

```
18036011,Yusuf,Arslan,0,0
19011009,Kate,Doe,0,0
19011034,Halil Ibrahim,Cengiz,0,0
19011013,MHD Besher,Al kurdi,0,0
20011045,Yasin,Yilmaz,0,0
```

Kullanıcıya ait bir şifre olmadığından kullanıcılar dosyasından silinmeyeceği belirtilmiştir.

```
Enter student details:
Number: 18036011
Deleted 18036011 successfully
User won't be deleted as they don't exist.
```

Değişim sonrası dosya içeriği:

```
19011009,Kate,Doe,0,0
19011034,Halil Ibrahim,Cengiz,0,0
19011013,MHD Besher,Al kurdi,0,0
20011045,Yasin,Yilmaz,0,0
```

Öğrenci listeleme işlemi:

```
Number - Name - Surname - Number of lessons - Total credit
19011009 - Kate - Doe - 0 - 0
19011034 - Halil Ibrahim - Cengiz - 0 - 0
19011013 - MHD Besher - Al kurdi - 0 - 0
20011045 - Yasin - Yilmaz - 0 - 0
    === END OF DATA ===
```

Kayıt geçmişi listeleme işlemi:
Bu işlem ders ekleme işleminden ve bazı öğrencilerin dersleri almasından sonra gerçekleşmiştir.

```
ID - Number - Course Code - Date - Enrolled
0 - 19011013 - BLM1012 - 2021-01-06 - 1
1 - 19011013 - BLM2041 - 2021-01-06 - 1
2 - 19011013 - BLM2512 - 2021-01-06 - 0
3 - 19011013 - BLM2521 - 2021-01-06 - 1
4 - 19011009 - BLM2051 - 2021-01-06 - 1
5 - 19011013 - BLM2031 - 2021-01-06 - 1
6 - 19011034 - BLM2031 - 2021-01-06 - 1
    === END OF DATA ===
```

Kurs İşlemleri

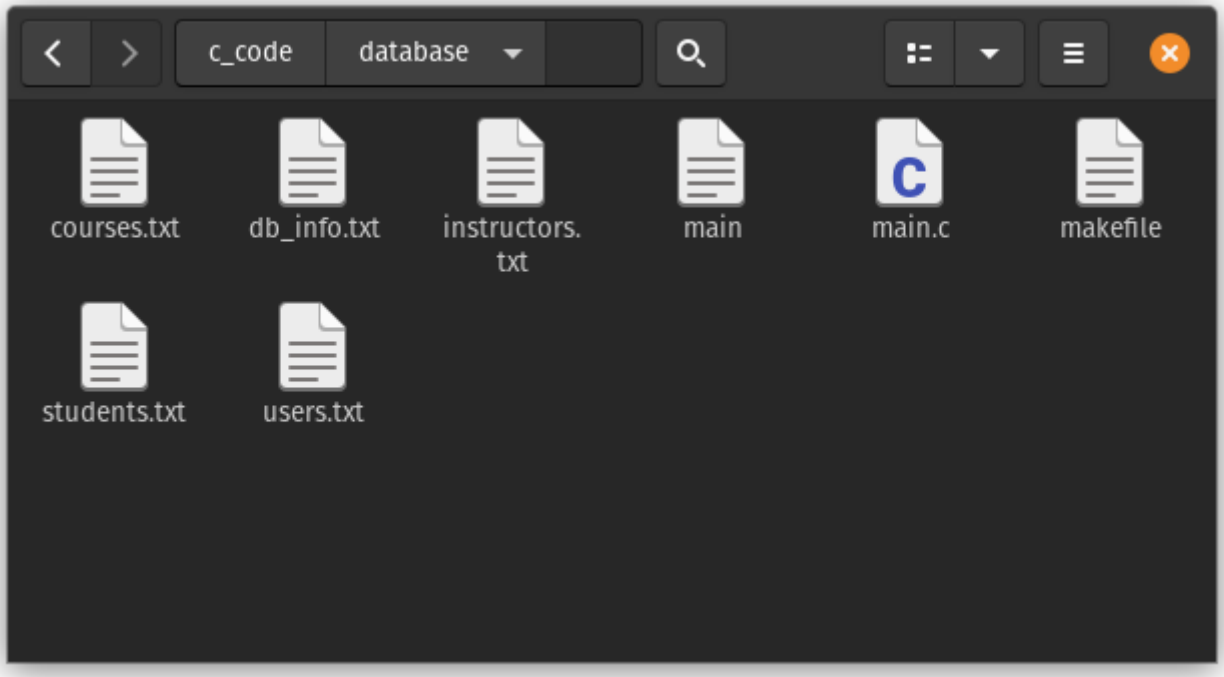
Bu menüde admin kullanıcısı kurs listeleme, ekleme, güncelleme ve silmenin yanında bir kursa kayıtlı olan öğrencileri listeleme işlemlerini yapabilir.

```
[1] List courses
[2] Add a course
[3] Edit a course
[4] Remove a course
[5] List enrolled students
[6] Back
Select from above [1-6]:
```

Kurs ekleme işlemi:
İşlem ilk defa yapıldığından yeni bir dosya oluşturulacaktır.

```
Enter course details.
Code: BLM2130
Name: Assembly
Credit: 3
Capacity: 100
Instructor ID: fcakmak
Warning: Couldn't open the file courses.txt.
(The following WILL delete any existing file with the same name)
Should I (re)create it? (y/n): y
info: Created courses.txt successfully
```

“courses.txt” dosyası oluşturuldu.



Dosya içeriği:

```
BLM2130,Assembly,3,100,fcakmak
```

Kurs güncelleme işlemi:

```
Course code: BLM2130
(Assembly) Enter new name:
(3) Enter new credit:
(100) Enter new capacity: 50
(fcakmak) Enter new instructor ID:
Updated BLM2130 successfully.
```

Dosya içeriği:

```
BLM2130,Assembly,3,50,fcakmak
```

Kurs silme işlemi:
Kurs ekleme adımı birden fazla kez tekrarlanırsa ve BLM2013 kodlu bir kurs oluşturulursa.
Dosya içeriği:

```
BLM1012,Introduction To Procedural Programming,4,40,irem
BLM2013,Not a course,4,15,aelbir
BLM2031,Structured Programming,4,80,yselcuk
BLM2041,Signals and Systems for Computer Engineers,3,75,aelbir
BLM2051,Seminar,0,20,yselcuk
BLM2130,Assembly,3,50,fcakmak
BLM2512,Data Structures and Algorithms,4,60,gbiricik
BLM2521,Discrete Mathematics,3,55,aelbir
BLM2611,Logic Circuits,4,35,hoilhan
```

```
Course code: BLM2013
Deleted BLM2013 successfully
```

Değişim sonrası dosya içeriği:

```
BLM1012,Introduction To Procedural Programming,4,40,irem
BLM2031,Structured Programming,4,80,yselcuk
BLM2041,Signals and Systems for Computer Engineers,3,75,aelbir
BLM2051,Seminar,0,20,yselcuk
BLM2130,Assembly,3,50,fcakmak
BLM2512,Data Structures and Algorithms,4,60,gbiricik
BLM2521,Discrete Mathematics,3,55,aelbir
BLM2611,Logic Circuits,4,35,hoilhan
```

Kurs listeleme işlemi:

```
Code - Name - Credit - Capacity - Instructor Id
BLM1012 - Introduction To Procedural Programming - 4 - 40 - irem
BLM2031 - Structured Programming - 4 - 80 - yselcuk
BLM2041 - Signals and Systems for Computer Engineers - 3 - 75 - aelbir
BLM2051 - Seminar - 0 - 20 - yselcuk
BLM2130 - Assembly - 3 - 50 - fcakmak
BLM2512 - Data Structures and Algorithms - 4 - 60 - gbiricik
BLM2521 - Discrete Mathematics - 3 - 55 - aelbir
BLM2611 - Logic Circuits - 4 - 35 - hoilhan
    === END OF DATA ===
```

Bir kursa kayıtlı olan öğrencileri listeleme işlemi:
Bu işlem bazı öğrencilerin dersleri almasından sonra gerçekleşmiştir.

```
Course code: BLM1012
Number - Name - Surname
19011013 - MHD Besher - Al kurdi
    ===END OF DATA===
```

Hoca

Hocanın girişi için ona tanımlanan ID kullanılır.

```
Username: yselcuk
Password:
Create a password for yselcuk: yselcuk
Success. Log in using your password.
```

Hoca kullanıcısının karşılaştığı ilk menüde yapmak istediği işlemin numarasını girerek seçebilir.

```
Welcome Dr. Yunus Emre Selcuk!
[1] List assigned courses
[2] Edit course details
[3] Export enrolled students list
[4] Log out
Select from above [1-4]:
```

Tanımlanan kursları listeleme

Bu işlemde kullanıcı ona tanımlı olan (onun verdiği) ders listesini yazdırabilir.

```
Code - Name - Credit - Capacity
BLM2031 - Structured Programming - 4 - 80
BLM2051 - Seminar - 0 - 20
    === END OF DATA ===
You are assigned 2 course(s).
```

Kurs bilgilerini değiştirme

Bu işlemde kullanıcı yalnızca ona tanımlı olan kursların kapasite ve isim bilgilerini güncelleyebilir.
Dosya içeriği:

```
BLM1012,Introduction To Procedural Programming,4,40,irem
BLM2031,Structured Programming,4,80,yselcuk
BLM2041,Signals and Systems for Computer Engineers,3,75,aelbir
BLM2051,Seminar,0,20,yselcuk
BLM2130,Assembly,3,50,fcakmak
BLM2512,Data Structures and Algorithms,4,60,gbiricik
BLM2521,Discrete Mathematics,3,55,aelbir
BLM2611,Logic Circuits,4,35,hoilhan
```

```
Course code: BLM2051
Press Enter to keep (old value)
(Seminar) Enter new  Name :
(20) Enter new  Capacity : 25
Updating as: BLM2051,Seminar,0,25,yselcuk
Updated BLM2051 successfully
```

Değişim sonrası dosya içeriği:

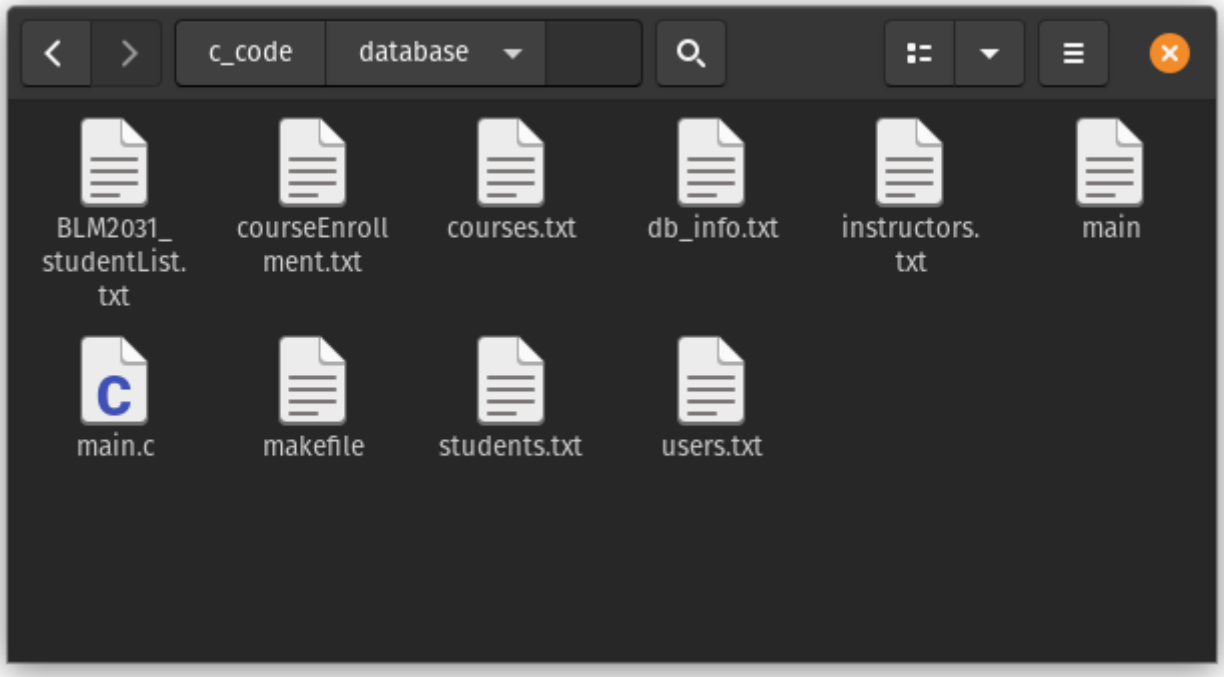
```
BLM1012,Introduction To Procedural Programming,4,40,irem
BLM2031,Structured Programming,4,80,yselcuk
BLM2041,Signals and Systems for Computer Engineers,3,75,aelbir
BLM2051,Seminar,0,25,yselcuk
BLM2130,Assembly,3,50,fcakmak
BLM2512,Data Structures and Algorithms,4,60,gbiricik
BLM2521,Discrete Mathematics,3,55,aelbir
BLM2611,Logic Circuits,4,35,hoilhan
```

Öğrenci listesi oluşturma

Bu işlem bazı öğrencilerin dersleri almasından sonra gerçekleşmiştir.

```
Course code: BLM2031
Number - Name - Surname
19011013 - MHD Besher - Al kurdi
19011034 - Halil Ibrahim - Cengiz
    ===END OF DATA===
```

“BLM2031_studentList.txt” dosyası oluşturuldu.



Dosya içeriği:

```
19011013 - MHD Besher - Al kurdi
19011034 - Halil Ibrahim - Cengiz
```

Öğrenci

Öğrenci numarasıyla giriş yapabilir.

```
Username: 19011013
Password:
Create a password for 19011013: 19011013
Success. Log in using your password.
```

İlk girişte şifre tanımlaması istenir. Öğrencinin karşılaştığı ana menü şu şekildedir:

```
Welcome MHD Besher Al kurdi!
Your credit: 0     Number of courses: 0
[1] List enrolled courses
[2] List available courses
[3] Enroll in a course
[4] Drop a course
[5] Log out
Select from above [1-5]:
```

Açılan kursları listeleme

Bu işlemde öğrenci verilen tüm dersleri listeleyebilir.

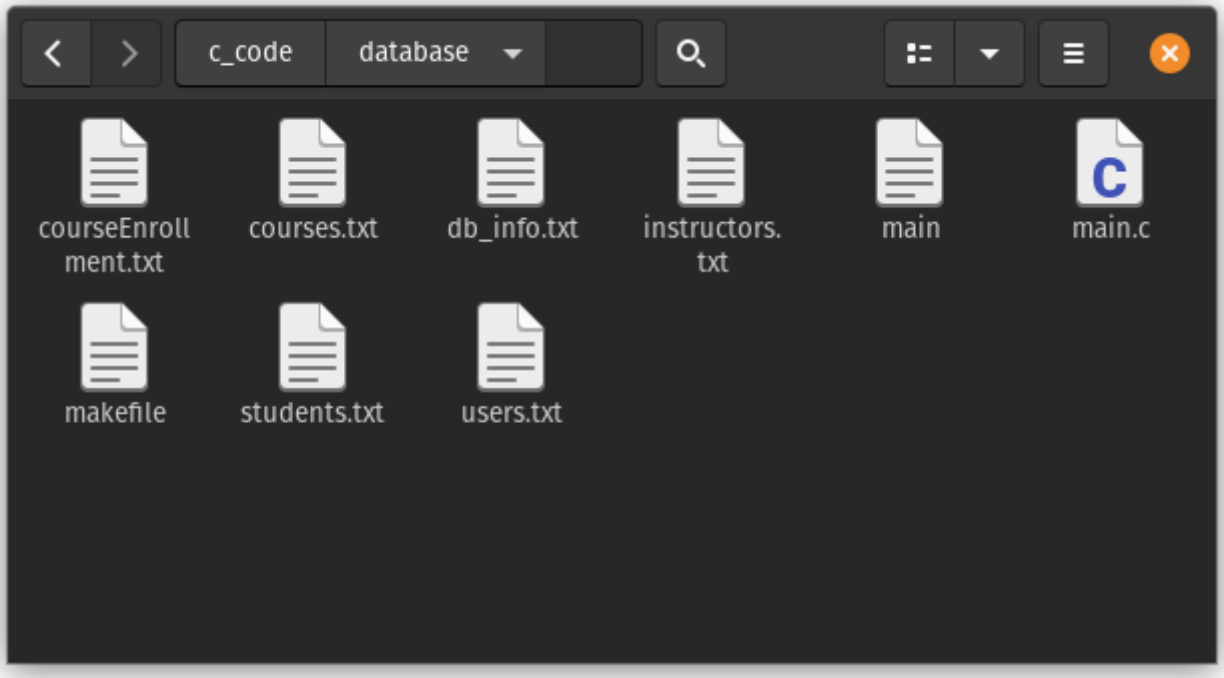
```
Code - Name - Credit - Capacity - Instructor Id
BLM1012 - Introduction To Procedural Programming - 4 - 40 - irem
BLM2031 - Structured Programming - 4 - 80 - yselcuk
BLM2041 - Signals and Systems for Computer Engineers - 3 - 75 - aelbir
BLM2051 - Seminar - 0 - 25 - yselcuk
BLM2130 - Assembly - 3 - 50 - fcakmak
BLM2512 - Data Structures and Algorithms - 4 - 60 - gbiricik
BLM2521 - Discrete Mathematics - 3 - 55 - aelbir
BLM2611 - Logic Circuits - 4 - 35 - hoilhan
    === END OF DATA ===
```

Ders ekleme

Bu işlemde öğrenci ders kaydını gerçekleştirebilir.

```
Enter course code: BLM1012
Enrolled in BLM1012 successfully.
```

“courseEnrollment.txt” dosyası oluşturulmuştur.



“courseEnrollment.txt” dosya içeriği:

```
0,19011013,BLM1012,2021-01-06,1
```

“students.txt” dosya içeriği. Kredi miktarı ve ders sayısı güncellenmiştir.

```
19011009,Kate,Doe,0,0
19011034,Halil Ibrahim,Cengiz,0,0
19011013,MHD Besher,Al kurdi,1,4
20011045,Yasin,Yilmaz,0,0
```

Ders bırakma

Öğrenci aldığı bir dersi bırakabilir.
“courseEnrollment.txt” dosya içeriği

```
0,19011013,BLM1012,2021-01-06,1
1,19011013,BLM2041,2021-01-06,1
2,19011013,BLM2512,2021-01-06,1
3,19011013,BLM2521,2021-01-06,1
```

“students.txt” dosya içeriği

```
19011009,Kate,Doe,0,0
19011034,Halil Ibrahim,Cengiz,0,0
19011013,MHD Besher,Al kurdi,4,14
20011045,Yasin,Yilmaz,0,0
```

```
Enter course code: BLM2512
Unenrolled from BLM2512 successfully.
```

Değişim sonrası “courseEnrollment.txt” dosya içeriği. İlgili kursun durumu 0 olarak güncellenmiştir.

```
0,19011013,BLM1012,2021-01-06,1
1,19011013,BLM2041,2021-01-06,1
2,19011013,BLM2512,2021-01-06,0
3,19011013,BLM2521,2021-01-06,1
```

Değişim sonrası “students.txt” dosya içeriği. Kredi miktarı ve ders sayısı güncellenmiştir.

```
19011009,Kate,Doe,0,0
19011034,Halil Ibrahim,Cengiz,0,0
19011013,MHD Besher,Al kurdi,3,10
20011045,Yasin,Yilmaz,0,0
```

Eklenen dersleri listeleme

Öğrenci aldğı dersleri bu işlemde listeler.

```
Course Code - Enrollment Date
BLM1012 - 2021-01-06
BLM2041 - 2021-01-06
BLM2521 - 2021-01-06
    === END OF DATA ===
```

Tanımlanan Fonksiyonlar

Fonksiyonları yazmak için kullanılan format aşağıdaki gibidir.

functionName

```
void functionName(type variable, type variable2)
```

Fonksiyonun açıklaması. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras rutrum dictum ipsum lacinia eleifend. Pellentesque ornare turpis non neque rhoncus.

Fonksiyonların tek dosyada tanımlanması istendiğinden okuma kolaylığı için alfabetik sıra tercih edilmiştir. Tanımlanan fonksiyonların listesi şu şekildedir:

addDataToFile

```
void addToFile(char* text, char* fileName)
```

Belirtilen dosyaya ilk sütunun sırasına uygun bir şekilde veri eklemesi yapar. Girilen verinin ilk sütunu benzersiz değilse ekleme yapmaz.

addTextToPosition

```
void addTextToPosition(FILE* filePointer, char* text, int deleteLine)
```

Gönderilen pointer'dan itibaren gönderilen yazıyı dosyaya ekler. deleteLine değişkeni 0 dışında bir değer ise filePointer ile gösterilen satırdaki veriyi siler. Dosyanın yeni içeriği "temp.txt" isimli bir dosyada tutulur.

alterCourses, alterInstructors, alterStudents, alterSettings

```
void alterCourses()
```

Birden fazla değiştirme seçeneği olan veriler için kullanıcıya yapabileceği değişiklikleri bir alt menü olarak sunan ve seçimine göre yönlendirme yapan bir fonksiyondur

changeDataCounter

```
void changeDataCounter(int dataColumn, int shouldIncrease)
```

"db_info.txt" dosyasında gönderilen dataType parametresine göre ilgili kayıt sayısını arttırır veya azaltır.

checkPassword

```
int checkPassword(char* username, char* password, int userType)
```

Verilen kullanıcı adı ve şifre parametrelerini kontrol edip eşleşme durumunda 1, yanlış şifre girilmiş ise 0, kullanıcı yok veya hata varsa -1 döndürür. userType parametresi admin için 1, hoca için 2, öğrenci için 3 olarak belirlenmiştir.

clearConsole

```
void clearConsole()
```

Kullanıcının uçbirimindeki (terminal / console) yazıları silen bir fonksiyondur. Platforma göre derleme sırasında farklı tanımlanır.

countEnrollments

```
int countEnrollments(char* courseCode)
```

courseEnrollment.txt dosyasında belirtilen ders kodu için aktif kayıt sayısını döndürür. Dosyaya erişim hatası varsa -1, hata yoksa kayıt sayısını döndürür.

countRows

```
int countRows(FILE* filePointer)
```

Gönderilen filePointer parametresinin işaret ettiği imleci dosyanın son satırının başına ilerletir. Dosyadaki satır sayısını döndürür.

createNewFile

```
int createNewFile(char fileName)
```

Yeni boş bir dosya oluşturur. Gönderilen parametreyi oluşturulan dosyanın ismi için kullanır. Başarı ile sonlanırsa 0 döner. Aksi takdirde -1 değerini döndürür.

createPassword

```
void createPassword(char* username,char* password,int userType)
```

Gönderilen parametrelere göre şifreli bir yazı oluşturarak, onu “users.txt” dosyasına ekler.

decryptPassword

```
int decryptPassword(char* username, char* password, char*
encryptedPassword);
```

Gönderilen username ve password parametrelerini deşifre eder. Kullanıcı türünü döndürür.

deleteDataFromFile

```
int deleteDataFromFile(char* key, char* fileName)
```

Belirtilen anahtara sahip olan veriyi dosyadan siler. Başarılı bir silme işlemi gerçekleşmiş ise 0 döndürür, işlem başarısız ise -1 döndürür.

deleteUser

```
int deleteUser(char* username,int userType)
```

Belirtilen kullanıcıyı “users.txt” dosyasından siler. Başarı durumunda 0, başarısızlık durumunda -1 değerini döndürür.

encryptPassword

```
char* encryptPassword(char* username, char* password, int userType)
```

Gönderilen username, password ve userType parametrelerini şifreler. Şifreli karakter dizisinin işaretleyicisini döndürür.

enrollStudent

```
int enrollStudent(char* studentNumber, char* courseCode)
```

Belirtilen öğrenci numarasına sahip olan öğrenciyi, ders kodunda bir ders açılmış ve daha önce öğrenci o derse kaydolmamış ise, onu derse kaydeder. İşlem başarısız ise -1, başarılı ise 0 döndürür.

exportEnrolledStudents

```
int exportEnrolledStudents(char* courseCode);
```

Derse kayıt olan öğrencileri listeledikten sonra onları “courseCode_studentList.txt” isim formatındaki bir dosyaya yazdırır.

findDataPosition

```
int findDataPosition(FILE* filePointer, char* key, int columnNumber)
```

Gönderilen filePointer parametresinin gösterdiği konumdan itibaren belirtilen sütunlarda arama yapar filePointer parametresinin gösterdiği imleci bulduğu ilk değerin satır konumuna ilerletir ve değeri bulana kadar gezdiği satır sayısını döndürür. Aranan değer yok ise -1 değerini döndürür.

getColumnString

```
int getColumnString(FILE* filePointer, char* key, int keyColumnNumber, int
stringColumnNumber, char* returnString)
```

Gönderilen filePointer parametresinin gösterdiği konumdan itibaren belirtilen sütunlarda key parametresine arama yapar değerin bulunduğu satırı veya boş key gönderilmiş ise ilk satırı sütunlara ayırarak stringColumnNumber sütunundaki yazıyı returnString’e yazdırır. Veri bulunamaz ise -1, bulunur ise gezdiği satır sayısını döndürür.

greetAdmin, greetStudent, greetInstructor

```
void greetAdmin(char username[], char password[])
```

Belirtilen türde gönderilen kullanıcı ismini ve şifreyi doğrulayıp başarı durumunda kullanıcının yapabileceği işlemleri listeleyip kullanılacak işlemi sorar.

hasColumn

```
int hasColumn(char* fileName,char* key,int columnNumber)
```

Gönderilen dosya isminde bir dosya varsa içinde arama yaparak key değerinin gönderilen columnNumber sütununda var olup olmadığını bulur. key varsa 1, yok ise 0, hata durumunda -1 döndürür.

isEnrolled

```
void isEnrolled(char* studentNumber, char* courseCode)
```

Öğrencinin belirtilen derse kaydolup olmadığını kontrol eder. Hata durumunda -1, kayıt durumu olumlu ise 2 değilse 1 döndürür. Böyle bir kayıt yoksa 0 döndürür. filePointer imlecini kaydın bulunduğu satırın başına getirir.

listAllData

```
void listAllData(char* fileName, const char* headerRow)
```

İlk parametre ismindeki dosyayı açıp dosyanın sonuna gelene kadar içindeki bilgileri yazdırır. Yazıların başına headerRow parametresi olarak gönderilen stringi yazdırır.

listAssignedCourses

```
void listAssignedCourses(char* instructorId)
```

Öğrencinin kayıt olduğu kursların listesini ekrana yazdırır.

listEnrolledStudents

```
int listEnrolledStudents(char* courseCode);
```

Belirtilen derse kayıt olan öğrencileri listeler.

listStudentEnrollments

```
void listStudentEnrollments(char* studentNumber)
```

Öğrencinin kayıt olduğu kursların listesini ekrana yazdırır.

parseToStruct

```
void* parseToStruct(char* stringToParse, int structType)
```

Gönderilen karakter dizisini içeriğindeki virgüllere göre ayırıp ilgili yapıya yerleştirir. structType parametresi yapının türünü belirler. 0: Student, 1: Course, 2: Instructor, 3: EnrollmentLog.
Hata durumunda NULL işaretçisi döndürür.

setColor

```
void setColor(int color)
```

Girilen sayıya göre yazıları boyayan veya orijinal rengine döndüren bir fonksiyondur. Platforma göre derleme sırasında farklı tanımlanır.

throwError

```
void throwError(char errorCode[], int isWarning, int lineNumber)
```

Program sırasında oluşan hata detaylarını kod karşılıklarına göre ekrana yazdırır. Fonksiyon yapılan hataların nereden kaynaklandıklarını tespit etmekte önemli bir rol oynuyor.

unEnrollStudent

```
int unEnrollStudent(char* studentNumber, char* courseCode)
```

Belirtilen öğrenci numarasına sahip olan öğrencinin belirtilen derse kaydı varsa o kaydı siler.

unEnrollAllStudents

```
int unEnrollAllStudents(char* courseCode)
```

Derse kaydolan tüm öğrencilerin kayıtlarını siler.

updateAttendersCredit

```
void updateAttendersCredit(char* courseCode, int creditChange)
```

Belirtilen kursa kaydolan tüm öğrencilerin kredisini creditChange parametresine göre değiştirir.

updateDataInFile

```
int updateDataInFile(char* key, char* fileName, int* editableColumns, const
char* headerRow)
```

Belirtilen fileName isimli dosyada key parametresi olarak gönderilen anahtar sütun ile ilişkili verilerin editableColumns dizisiyle belirtilen değiştirilebilir sütunlar için kullanıcıdan veri girişi istenir. headerRow parametresi kullanıcıya gireceği verinin türünü belirtmek için kullanılır. İşlem başarılı ise 0, veri dosyada yok ise 1, hata durumunda -1 döndürür.

updateStudentEnrollment

```
int updateStudentEnrollment(char* studentNumber, int courseCredit, int
enrolled, int isStudent)
```

Belirtilen öğrencinin kaydolduğu ders sayısını ve aldığı kredi sayısını günceller. Enrolled 0 dışında bir değer ise arttırır, değilse azaltır. Başarı durumunda 0, diğer durumlarda -1 değerini döndürür.

Tanımlanan Yapılar

Course

```c
typedef struct {
  char code[20];
  char name[75];
  int credit;
  int capacity;
  int instructorId;
} Course;
```

EnrollmentLog

```c
typedef struct {
  int id;
  char studentNumber[30];
  char courseCode[20];
  char date[20];
  int enrolled;
} EnrollmentLog;
```

Instructor

```c
typedef struct {
  int id;
  char name[30];
  char surname[30];
  char title[20];
} Instructor;
```

Student

```c
typedef struct {
  char number[30];
  char name[75];
  char surname[30];
  int lessonCount;
  int totalCredit;
} Student;
```

Programın Çalışması Sırasında Oluşabilecek Dosyalar

- **“instructors.txt”**
  Kayıtlı hocaların bilgilerini içerir. Her satırın içerdiği kayıttaki bilgiler sırasıyla:
  benzersiz kullanıcı adı, ad, soyad, ünvan
- **“courses.txt”**
  Kayıtlı kursların bilgilerini içerir. Her satırın içerdiği kayıttaki bilgiler sırasıyla:
  benzersiz ders kodu, ad, kredi, kapasite, hocanın kullanıcı adı
- **“students.txt”**
  Kayıtlı öğrencilerin bilgilerini içerir. Her satırın içerdiği kayıttaki bilgiler sırasıyla:
  benzersiz öğrenci numarası, ad, soyad, ders sayısı, kredi miktarı
- **“courseEnrollment.txt”**
  Yapılan ders kayıtlarının bilgilerini içerir. Her satırın içerdiği kayıttaki bilgiler sırasıyla:
  benzersiz ID, öğrenci numarası, ders kodu, kayıt tarihi, kayıt durumu
- **“users.txt”**
  Şifresi olan kullanıcıların bilgilerini şifreli bir şekilde içerir. Şifreleme fonksiyonun anlatıldığı bölümde satırların yapısından bahsedilmiştir.
- **“db_info.txt”**
  Veri tabanının ayarlarını ve genel bilgilerini içerir. Her satırın içerdiği kayıttaki bilgiler sırasıyla:
  kredi sınırı, ders sayısı sınırı, kayıtlı ders sayısı, kayıtlı hoca sayısı, kayıtlı öğrenci sayısı, log sayısı
- **“dersKodu_studentList.txt”**
  Hoca tarafından oluşturulabilen bu dosya bir derse kayıtlı olan öğrencilerin numara, isim, soyisim bilgilerini belirtilen sırayla içerir.