



T.C.
ONDOKUZMAYIS ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
VERİ YAPILARI ÖDEV RAPORU



Ödev Başlığı: Bağlı Listeler Yardımıyla Telefon Rehberi Tasarlama

Ürünün İçermesi Gereken Özellikler: Ad, Soyad, Telefon Numarası ve E-Mail verileri ile Kişi Kaydetme, Ad veya Soyad ile Kişi Arama, Silme İşlemlerinin Gerçekleştirilmesi ve Rehberin Görüntülenebilmesi.

Ödev Kazanımları: Bağlı Listeler, Bağlı Listelerde Operasyonlar, Bağlı Listelerin Uygulaması

Yapılan Ürünün Özellikleri: Kişi Ekleme, Kişi Silme, Rehber Görüntüleme, Kişi Arama, Uygulamayı Kapatma, Her İşlem Sonrasında Konsol Ekranının Temizlenmesi.

Ürünün Kaynak Kodu ve Çalıştırılabilir Dosyaları (Linux ve Windows) “21060619_alper_karaca.zip” İsimli Sıkıştırılmış Klasörde Mevcuttur.

Tanımlı Fonksiyonlar:

- `make_new_node()`
- `add_contact()`
- `display()`
- `delete_contact`
- `search_contact()`
- `clrscr()`

Kaynak Kodunda Bulunan Fonksiyonlar ve struct Yapıları ve Kullanım Amaçları :

`Node *user_1 = NULL;` (Ana düğümü public (Her yerden erişilebilir) olarak oluştur)

`//Fonksiyon Çağrılarını yap.`

`Node *make_new_node(char[], char[], char[], char[]);`

`void add_contact();`

`void display();`

`void delete_contact();`

`void search_contact();`

`void clrscr();`

Bağlı Listenin Oluşturulması:

```
typedef struct Node {  
    char name[MAX_NAME_LEN];  
    char surname[MAX_NAME_LEN];  
    char number[15];  
    char email[MAX_EMAIL_LEN];  
    struct Node *next;  
} Node;
```

Bu kod dizesinde Node adı ile bir bağlı liste oluşturdum ve içerisinde İsim, Soyisim. Telefon numarası, Mail Adresi ve sonraki düğüme bağlanmayı sağlayacak olan adres değerini tutmasını sağlayacak referanslar girdim.

`Node *make_new_node(char na[], char su[], char nu[], char em[])` Fonksiyonu:

Node *make_new_node(char na[], char su[], char nu[], char em[]) Fonksiyonu:

```
Node *make_new_node(char na[], char su[], char nu[], char
em[]){
    Node *yenidugum = (Node*)malloc(sizeof(Node));
    strcpy(yenidugum->name, na);
    strcpy(yenidugum->surname, su);
    strcpy(yenidugum->number, nu);
    strcpy(yenidugum->email, em);
    yenidugum->next = NULL;
    return yenidugum;
}
```

Bu fonksiyon Node türünde olduğu için adresle bir işlem yapmak gerekiyor ve 4 farklı parametre alıyor. Alınan parametrelerin amacı kullanıcıdan alınan ad, soyad, numara ve e-mail girdilerini düğümümüzde bulunan ilgili kısımlara yerleştirmek. Hafızada yer tahsis etmek.

Bu dizede string.h başlık dosyası içerisinde tanımlı olan strcpy() fonksiyonunu kullandık. Bu fonksiyonu kullanmış olma sebebimiz ise C dilinde String dediğimiz değişken türünün karakter (Char) dizisi olarak tanımlanmış olmasıdır. Bu fonksiyon ise kendisine verilen karakter dizisini ise hedef bölge üzerine yazar.

Kullanım şekli:

strcpy(hedef, kaynak); şeklindedir.

Bunlara ek olarak oluşturacağımız her düğüm için bellekte “Node” kadar yer ayırıyoruz böylece bellek hataları veya yanlış erişim gibi hatalarının önüne geçiyoruz. Ardından oluşturulan düğümün sonraki adresini tutan next isimli bölgeye “NULL” değerini atıyoruz ki programımızı çalıştırırken bitiş noktası belirleyebilelim.

Son olarak da fonksiyonumuz Node tanımlı olduğu için bir Node yapılı veri döndürmemiz gerekiyor bu sebeple de yenidugum adını verdiğimiz Node türlü değişkeni döndürüyoruz.

Void add_contact(char na[],char su[],char nu[], char em[])

```
void add_contact(char na[],char su[],char nu[], char em[]){
    Node *new_node = make_new_node(na, su, nu, em);
    if(user_1 == NULL)
        user_1 = new_node;
    else{
        Node *temp = user_1;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = new_node;
    }
}
```

Bu fonksiyon void türlü bir fonksiyon ve yine bu fonksiyonda isim, soyisim, numara ve e-mail parametrelerini alarak listenin sonuna yeni bir düğüm ekleme görevini üstleniyor.

Node türünde new_node adında bir geçici düğüm tanımlıyoruz ve bunu make_new_node(na. su. nu. em)'e eşitleyerek gönderilen verilerin doğru yerlere yerleştirilmesini sağlıyoruz.

Devamındaki if bloğunda public (her yerden erişilebilir) olarak tanımladığımız user_1 isimli ana düğümümüzün doluluğunu kontrol ediyoruz. Eğer boş ise user_1 listemizi new_node ile eşitliyoruz.

Şayet liste boş değil ise de yeni temp isimli Node türlü düğüm oluşturup bunu anadüğümümüz ile doldurup içerisinde geziniyoruz. Eğer en sona ulaşırsak da sonraki düğümü new_node ile eşitliyoruz.

void display()

```
void display(){
    Node *traverse = user_1;
    if(traverse == NULL)
        printf("Rehber Boş\n");
    else{

        printf("|-----|\n|KAYITLI KULLANICILAR|
\n|-----|\n");
        int i = 1;
        while(traverse != NULL){
            if(i<10) printf("|-----|\n| [%d. Kişi] |
\n|-----|\n\n Ad: %sSoyad: %sTelefon Numarası: %s
E-Mail Adresi: %s", i,traverse->name,traverse->surname,
traverse->number, traverse->email);
            else printf("|-----|\n| [%d. Kişi] |\n
|-----|\nAd: %s Soyad: %sTelefon Numarası: %s
E-Mail Adresi: %s", i,traverse->name, traverse->surname,
traverse->number, traverse->email);
            traverse = traverse->next;
            i++;
            printf("\n");
        }
    }
}
```

Void türlü bu fonksiyonda amacımız listeyi görüntülemek ve bunu yapmak için traverse adı verilen ve ana düğüme eşit olan Node yapılı bir liste oluşturuyoruz. Ardından liste boş ise “Rehber Boş” çıktısını gösteriyoruz. Liste boş değil ise her bir kullanıcının Kayıt Sırası. Adı. Soyadı. Telefon Numarası ve E-Mail adresi çıktılarını yazdırıyoruz.

void delete_contact()

```
void delete_contact(Node** ref, char name_to_delete[],int
option){
    Node *tmp = *ref, *previous;
    if(tmp == NULL) {printf("Rehber boş !\n"); return;}
    while(tmp != NULL){
        switch (option)
        {
            case 1:
                if(!(strcmp(tmp->name, name_to_delete))){
                    *ref = tmp->next;
                    printf(" Silinen Kişi\nAdı: %sSoyadı:%s
Telefon Numarası: %sMail Adresi: %s ", tmp->name, tmp->
surname, tmp->number, tmp->email);
                    free(tmp);
                    return;
                }
                if(tmp->next == NULL)
                    printf("Aranan Kişi Rehberde Bulunamadı !
\n");
                    return;
                break;
            case 2:
                if(!(strcmp(tmp->surname, name_to_delete))){
                    *ref = tmp->next;
                    printf("Silinen Kişi\nAdı: %sSoyadı: %s
Telefon Numarası: %sMail Adresi: %s ", tmp->name, tmp->
surname,tmp->number, tmp->email);
                    free(tmp);
                    return;
                }
                if(tmp->next==NULL)
                    printf("Aranan Kişi Rehberde Bulunamadı !
\n");
                    return;
                break;
            default:
                break;
        }
        previous->next = tmp->next;
        free(tmp);
    }
}
```

void türlü bu fonksiyonun 3 parametre alıyor. Sırasıyla bağlı listenin referansı. Silinecek isim ya da soyisim ve seçilen işlem türü (Tam Sayı).

Fonksiyonun ise isim ya da soyisme göre aranılan kişiyi bulmak ve listeden kaldırmak. Öncelikle ana düğümümüzün yapısını bozmamak için tmp adında geçici bir düğüm ve previous adında önceki düğümü tutacağımız iki değişken oluşturuyoruz.

Ardından tmp listesinin doluluğunu kontrol ediyoruz eğer boş ise “Rehber Boş” Çıktısını verip fonksiyondan çıkış sağlıyoruz.

Eğer rehberde kayıtlı kişi (veri ve düğüm) var ise İsme Göre Arama ya da Soyisme Göre Arama Seçeneklerine göre sorgulama yapıyoruz. (Buranın devamında option olarak 1 değeri gönderilmiş yani isme göre işlem yapılmış olarak anlatacağım ancak aynı adımlar soyadı içinde aynı şekilde ilerliyor.)

If bloğunda string.h başlık dosyasında tanımlı olan strcmp() fonksiyonunu kullanıyoruz. Bu fonksiyon her iki stringi karşılaştırır ve aynı olmaları durumunda 0 farklı olmaları durumunda 1 değerini döndürür. C dilinde ise NULL ve 0 dışında her şey true değerine denk geldiği için ! ile oluşan değer tersini alıyoruz ve bu koşul karşılandığı sürece de anadüğümü tmp’nin sonraki düğümüne bağlıyoruz ardından silinen kişinin verilerini gösterip bellekten tmp’yi kaldırıyoruz (free fonksiyonu ile)

Eğer tüm listemizi gezinmemize rağmen listede aranan kişi bulunmadıysa “Aranan Kişi Rehberde Bulunamadı” çıktısını verip fonksiyondan çıkıyoruz. Switch-case yapısının default durumu olarak ise direkt olarak bloktan çıkıyoruz.

void search_contact()

```
void search_contact(Node **ref, char name[], int option){
    Node *crt = *ref;
    if(crt == NULL)
        printf("Rehber Boş !\n");
    while(crt != NULL){
        switch (option)
        {
            case 1:
                if(!strcmp(crt->name,name)){
                    printf("Aranan Kişi Bulundu !\n");
                    printf("Ad: %sSoyad: %sTelefon Numarası: %sE-Mail Adresi: %s", crt->name, crt->surname, crt->number, crt->email);
                    return;
                }

                if(crt->next == NULL)
                    printf("Böyle Bir Kontak Yok !");
                crt = crt->next;
                break;
            case 2:
                if(!strcmp(crt->surname,name)){
                    printf("Aranan Kişi Bulundu !\n");
                    printf("Ad: %sSoyad: %sTelefon Numarası: %sE-Mail Adresi: %s", crt->name, crt->surname, crt->number, crt->email);
                }
                if(crt->next == NULL)
                    printf("Böyle Bir Kontak Yok");
                crt = crt->next;
            default:
                break;
        }
    }
    return ;
}
```

Bu fonksiyonda kişi silme fonksiyonu ile aynı parametreleri alıyor ve aranan kişiyi bulduğu zaman bilgilerini çıktı olarak sunuyor.

crt adında Node türlü bir liste oluşturup bunu ana düğümümüzle dolduruyoruz. Eğer liste boş ise “Rehber Boş” çıktısını verip fonksiyondan çıkıyoruz. Liste dolu ise de yine delete_contact() kısmında anlattığım gibi iki karakter dizisini karşılaştırıyoruz ve eğer eşleşme olursa aranan kişinin bilgilerini ekrana yazdırıyoruz.

void clrscr()

```
void clrscr(){
    #ifdef _WIN32
        system("cls");
        //https://stackoverflow.com/questions/44679195/how-to
        -send-command-to-linux-command-from-c-program
    #else
        system("clear");
    #endif
}
```

Normalde conio.h başlık dosyasında tanımlı olması gereken clrscr() fonksiyonunu, conio.h dosyasını tüketmeme rağmen kodumda fonksiyonu kullanamadım ardından [C kodundan sistem komutu gönderme](#) ve [C dilinde işletim sistemini tanımlama](#) uzantılarındaki bilgiler aracılığıyla kodumun herhangi bir işlem sonrasında Enter tuşunu tıklanması durumunda Konsolu temizleyip ana menüye dönerek temiz bir görünüm kazanmasını amaçladım.

DIPNOT: Bu kısmın gerekliliği tartışılır olabilir sonuç olarak Veri Yapıları dersinde amacımız bellek yönetimini sağlamak ve olabildiğince az kaynak tüketerek çok işlem sağlamak (verimi arttırmak). Fakat bu yapmış olduğumuz program kullanıcı ile etkileşim gerektiren bir yapı olduğu için arayüzün güzel görünmesi için ek olarak zaman ve kaynak harcamasında bulundum. Dersin çerçevelerine ne kadar sığacağı hakkında bir bilgim olmasa da eğer bir sorun teşkil ediyorsa iletmenizi içtenlikle isterim.

int main()

```
char t_name[MAX_NAME_LEN], t_surname[MAX_NAME_LEN],
t_number[15], t_email[MAX_EMAIL_LEN], save_order[4],
selection[3], inline_selection[3];
int count = 0, total_contact = 1, in_sel;
float person, f_selection;

decision:
clrscr();
//https://iq.opengenus.org/detect-operating-sy
stem-in-c/
printf(
"|-----|
---|
\n");
printf(
"|  Lütfen İşlem Seçiminizi Bir Rakam Belirterek Yapınız
!  |
\n");
printf(
"|-----|
---|
\n");
printf("\t1)Rehbere Kişi Ekle\n\t2)Rehberde Kişi Ara\n\t
3)Rehberden Kişi Sil\n\t4)Kayıtlı Kişileri Görüntüle\n\t
5)Uygulamayı Kapat\nSeçiminiz: ");
fgets(selection, 3, stdin);
f_selection = strttof(selection, NULL);
```

Kullanıcıdan alınacak olan karakter dizilerini, rehberdeki kayıt sırasını. arama ve silme fonksiyonlarında kullanılacak seçeneği seçmek için kullanılacak olan değeri, toplam kişi sayısını, kayıt edilecek kişi sayısı ve fonksiyonlara gönderilecek değerleri tutan değişkenler tanımlıyorum.

DİPNOT: While() döngüsü yerine goto özelliğini kullandım burada bulunan decision: yapısı ise label (etiket) olarak bulunmakta işlemler tamamlandıktan sonra kodumun bu kısma tekrar dönmesini sağlıyorum. Kullanılmasının pek iyi bir durum olmadığına farkındayım.

Ekran çıktılarını vb. sunduktan sonra burada string.h dosyasında tanımlı olan fgets() fonksiyonu karşımıza çıkıyor. Fgets() kullanmamdaki asıl amaçlar ise sırasıyla:

- scanf() ile değer alırken ardından gelen scanf fonksiyonundaki değeri NULL olarak ataması
- Bir kişinin iki isminin bulunması ve scanf fonksiyonun boşluklarda (Whitespace) girilen değer sonuna geldiğini varsayması. Ancak fgets Enter ile gelen \n\0 dizisine kadar alıyor ve böylece çıktıları yazarken de her değeri yazdırdıktan sonra yeni satır oluşturma zahmetinden bizi kurtarıyor.

Ardından kullanıcıdan yapmasını istediği işlemi girdikten sonra gönderilen değer bir char olduğu için onu strtouf() fonksiyonu ile bir sayıya çeviriyorum.

```
switch ((int)f_selection)
{
    case 1:
        printf("\nKaç Kişi Kaydedilecek: ");
        fgets(save_order,4,stdin);
        person = strtouf(save_order,NULL);
        while(count < person){
            printf("\n[%d. Kişi Kaydediliyor.]\n",
, total_contact);
            printf("Ad Giriniz: ");
            fgets(t_name, MAX_NAME_LEN, stdin);
            printf("Soyad Giriniz: ");
            fgets(t_surname, MAX_NAME_LEN, stdin);
            printf("\n
Telefon Numarasını Alan Kodu İle Giriniz (Örnek: +9055555
55555): "
);
            fgets(t_number,15, stdin);
            printf("\nMail Adresi Giriniz: ");
            fgets(t_email, MAX_EMAIL_LEN, stdin);
            if(!strchr(t_email,'@')){
                printf(
"Bu Geçerli Bir Mail Adresi Değil !\n");
                printf(
"Ana Menüye Dönmek İçin Enter'e Basınız.\n");
                getchar();
                count = 0;
                goto decision;
            }
            add_contact
(t_name, t_surname, t_number, t_email);
            count++;
            total_contact++;
            clrscr();
        }
        count = 0;
        goto decision;
        break;
}
```

Switch yapısına sayıya dönüştürdüğüm değeri Integer olarak gönderip girilen değer 1'e eşit olması durumunda isim, soyisim, telefon numarası ve e-mail bilgilerini(Eğer @ karakteri yoksa geçerli bir mail adresi olmadığı için kaydetmez) isteyip bunları rehber kaydediyorum.

Kayıt işleminin ardından ise Enter tuşuna basılması durumunda ekranı temizleyip ana sayfaya dönmesini sağlıyorum.

```

case 2:
    char s_name[MAX_NAME_LEN];
    printf("Hangi Değere Göre Arama Yapılsın\n
1)Ad İle Arama Yap\n2)Soyadı İle Arama Yap\nSeçiminiz: "
);
    fgets(inline_selection,3,stdin);
    in_sel = atoi(inline_selection);
    switch (in_sel)
    {
    case 1:
        printf(
"Aranacak Kişinin İsmi Harflere Dikkat Ederek Yazınız: "
);
        fgets(s_name, MAX_NAME_LEN, stdin);
        search_contact(&user_1,s_name,in_sel);
        break;
    case 2:
        printf(
"Aranacak Kişinin Soyismini Harflere Dikkat Ederek Yazını
z: "
);
        fgets(s_name, MAX_NAME_LEN, stdin);
        search_contact(&user_1, s_name,in_sel);
        break;
    default:
        printf("Hatalı Seçim Yapıldı.");
        break;
    }

    printf("\n
Ekranı Temizleyip Ana Sayfaya Dönmek İçin Enter Tuşuna Ba
sınız !
\n");
    getchar();
    goto decision;
    break;

```

Eğer değer 2'ye eşit olursa da Rehberde Arama işlemini gerçekleştiriyor. Burada s_name adında kullanıcıdan araması istediği kişi verisi alınıyor. Ardından kullanıcının Ad ve Soyada göre arama seçeneklerinden birini seçmesi isteniyor. Seçiminin ardından Kendisinden arama yapılacak kişinin bilgisi isteniyor ve rehberde arama yapılıyor. Arama seçenekleri yanlış seçilmiş ise hata mesajı ve ana ekrana dönme uyarısı veriliyor. Enter tuşuna bastıktan sonra ekran temizlenip ana menüye dönüş sağlanıyor.

```

case 3:
    char del_name[MAX_NAME_LEN];
    printf("Hangi Değere Göre Silinsin\n
1)Ad İle Sil\n2)Soyad İle Sil\nSeçiminiz: ");
    fgets(inline_selection,3,stdin);
    in_sel = atoi(inline_selection);
    switch (in_sel)
    {
        case 1:
            printf(
"Silinecek Kişinin Adını Giriniz: ");
            fgets(del_name, MAX_NAME_LEN,stdin);
            delete_contact(&user_1, del_name, in_sel
);
            break;
        case 2:
            printf(
"Silinecek Kişinin Soyadını Giriniz: ");
            fgets(del_name, MAX_NAME_LEN, stdin);
            delete_contact(&user_1, del_name, in_sel
);
            break;

        default:
            printf("Hatalı Seçim Yaptınız !");
            break;
    }

    total_contact--;
    printf("\n
Ekranı Temizleyip Ana Sayfaya Dönmek İçin Enter Tuşuna Ba
sınız !
\n");

    getchar();
    goto decision;
    break;

```

Eğer değer 3'e eşit olursa da silme işlemi gerçekleşiyor. Yeniden arama işleminde bulunan işlemler burası için de geçerli olmakla beraber ek olarak rehberde kayıtlı kişi sayısı silinen kişi başına 1 azaltılıyor.

```
        case 4:
            display();
            printf("\n
Ekranı Temizleyip Ana Sayfaya Dönmek İçin Enter Tuşuna Bası
nız !
\n");
            getchar();
            goto decision;
            break;

        case 5:
            clrscr();
            return -1;
    default:
        printf("Yanlış Seçim Yaptınız.\n");
        printf("\n
Ekranı Temizleyip Ana Sayfaya Dönmek İçin Enter Tuşuna Bası
nız !
\n");
        getchar();
        goto decision;
        break;
}
```

Eğer değer 4'e eşit ise rehberi görüntüleyip ana menüye dönme mesajını çıktı olarak gösteriyor.

Eğer değer 5'e eşit ise ekranı temizleyip programı kapatıyor.

Şayet değer sunulan seçenekler arasından birisi değilse hata mesajı gösteriliyor ve ana menüye dönme mesajı gösteriliyor.

Program Çıktıları:

- İşlem Seçme Ekranı

```
alpeerkaraca@pop-os: ~/Desktop/data-structures-assignments/Assignment-1
-----
| Lütfen İşlem Seçiminizi Bir Rakam Belirterek Yapınız ! |
|-----|
1)Rehbere Kişi Ekle
2)Rehberde Kişi Ara
3)Rehberden Kişi Sil
4)Kayıtlı Kişileri Görüntüle
5)Uygulamayı Kapat
Seçiminiz: 
```

- Kişi Ekleme

```
alpeerkaraca@pop-os: ~/Desktop/data-structures-assignments/Assignment-1
-----
| Lütfen İşlem Seçiminizi Bir Rakam Belirterek Yapınız ! |
|-----|
1)Rehbere Kişi Ekle
2)Rehberde Kişi Ara
3)Rehberden Kişi Sil
4)Kayıtlı Kişileri Görüntüle
5)Uygulamayı Kapat
Seçiminiz: 1
Kaç Kişi Kaydedilecek: 1
[1. Kişi Kaydediliyor.]
Ad Giriniz: Alper
Soyad Giriniz: Karaca
Telefon Numarasını Ülke ve Alan Kodu İle Giriniz (Örnek: +905555555555): +905316237966
Mail Adresi Giriniz: alpeerkaraca@gmail.com
```

- **Kişi Ekleme Hata Ekranı**

```
alpeerkaraca@pop-os: ~/Desktop/data-structures-assignments/Assignment-1
|-----|
| Lütfen İşlem Seçiminizi Bir Rakam Belirterek Yapınız ! |
|-----|
1)Rehberde Kişi Ekle
2)Rehberde Kişi Ara
3)Rehberden Kişi Sil
4)Kayıtlı Kişileri Görüntüle
5)Uygulamayı Kapat
Seçiminiz: 1
Kaç Kişi Kaydedilecek: 1
[2. Kişi Kaydediliyor.]
Ad Giriniz: @
Soyad Giriniz: Test Ediliyor
Telefon Numarasını Ülke ve Alan Kodu İle Giriniz (Örnek: +90555555555): +9055555555
Mail Adresi Giriniz: testQtest.com
Bu Geçerli Bir Mail Adresi Değil !
Ana Menüye Dönmek İçin Enter'e Basınız.
|
```

- **Rehber Sonuna Kişi Ekleme**

```
alpeerkaraca@pop-os: ~/Desktop/data-structures-assignments/Assignment-1
|-----|
| Lütfen İşlem Seçiminizi Bir Rakam Belirterek Yapınız ! |
|-----|
1)Rehberde Kişi Ekle
2)Rehberde Kişi Ara
3)Rehberden Kişi Sil
4)Kayıtlı Kişileri Görüntüle
5)Uygulamayı Kapat
Seçiminiz: 1
Kaç Kişi Kaydedilecek: 1
[2. Kişi Kaydediliyor.]
Ad Giriniz: Test
Soyad Giriniz: Edilen
Telefon Numarasını Ülke ve Alan Kodu İle Giriniz (Örnek: +90555555555): +9055555555
Mail Adresi Giriniz: test@test.com|
```


- İsim İle Arama

```
alpeerkaraca@pop-os: ~/Desktop/data-structures-assignments/Assignment-1
| Lütfen İşlem Seçiminizi Bir Rakam Belirterek Yapınız ! |
|-----|
| 1)Rehbere Kişi Ekle |
| 2)Rehberde Kişi Ara |
| 3)Rehberden Kişi Sil |
| 4)Kayıtlı Kişileri Görüntüle |
| 5)Uygulamayı Kapat |
Seçiminiz: 2
Hangi Değere Göre Arama Yapılsın
1)Ad İle Arama Yap
2)Soyadı İle Arama Yap
Seçiminiz: 1
Aranacak Kişinin İsmi Harflere Dikkat Ederek Yazınız: Alper
Aranan Kişi Bulundu !
Ad: Alper
Soyad: Karaca
Telefon Numarası: +905316237966
E-Mail Adresi: alpeerkaraca@gmail.com
Ekranı Temizleyip Ana Sayfaya Dönmek İçin Enter Tuşuna Basınız !
```

- Soyisim İle Arama

```
alpeerkaraca@pop-os: ~/Desktop/data-structures-assignments/Assignment-1
| Lütfen İşlem Seçiminizi Bir Rakam Belirterek Yapınız ! |
|-----|
| 1)Rehbere Kişi Ekle |
| 2)Rehberde Kişi Ara |
| 3)Rehberden Kişi Sil |
| 4)Kayıtlı Kişileri Görüntüle |
| 5)Uygulamayı Kapat |
Seçiminiz: 2
Hangi Değere Göre Arama Yapılsın
1)Ad İle Arama Yap
2)Soyadı İle Arama Yap
Seçiminiz: 2
Aranacak Kişinin Soyismini Harflere Dikkat Ederek Yazınız: Karaca
Aranan Kişi Bulundu !
Ad: Alper
Soyad: Karaca
Telefon Numarası: +905316237966
E-Mail Adresi: alpeerkaraca@gmail.com
Böyle Bir Kontak Yok
Ekranı Temizleyip Ana Sayfaya Dönmek İçin Enter Tuşuna Basınız !
```

- **Kişi Arama Hata Ekranı**

```
alpeerkaraca@pop-os: ~/Desktop/data-structures-assignments/Assignment-1
|-----|
| Lütfen İşlem Seçiminizi Bir Rakam Belirterek Yapınız ! |
|-----|
| 1)Rehbere Kişi Ekle |
| 2)Rehberde Kişi Ara |
| 3)Rehberden Kişi Sil |
| 4)Kayıtlı Kişileri Görüntüle |
| 5)Uygulamayı Kapat |
Seçiminiz: 2
Hangi Değere Göre Arama Yapılsın
1)Ad ile Arama Yap
2)Soyadı ile Arama Yap
Seçiminiz: 1
Aranacak Kişinin İsmi Harflere Dikkat Ederek Yazınız: krc
Böyle Bir Kontak Yok !
Ekranı Temizleyip Ana Sayfaya Dönmek İçin Enter Tuşuna Basınız !
|
```

- **Rehberi Görüntüle**

```
alpeerkaraca@pop-os: ~/Desktop/data-structures-assignments/Assignment-1
|-----|
| Lütfen İşlem Seçiminizi Bir Rakam Belirterek Yapınız ! |
|-----|
| 1)Rehbere Kişi Ekle |
| 2)Rehberde Kişi Ara |
| 3)Rehberden Kişi Sil |
| 4)Kayıtlı Kişileri Görüntüle |
| 5)Uygulamayı Kapat |
Seçiminiz: 4
|-----|
| KAYITLI KULLANICILAR |
|-----|
|-----|
| [1. Kişi] |
|-----|
Ad: Alper
Soyad: Karaca
Telefon Numarası: +905316237966
E-Mail Adresi: alpeerkaraca@gmail.com
|-----|
| [2. Kişi] |
|-----|
Ad: Test
Soyad: Edilen
Telefon Numarası: +9055555555
E-Mail Adresi: test@test.com
Ekranı Temizleyip Ana Sayfaya Dönmek İçin Enter Tuşuna Basınız !
|
```

- İsim ile Kişi Sil

```
alpeerkaraca@pop-os: ~/Desktop/data-structures-assignments/Assignment-1
|-----|
| Lütfen İşlem Seçiminizi Bir Rakam Belirterek Yapınız ! |
|-----|
1)Rehbere Kişi Ekle
2)Rehberde Kişi Ara
3)Rehberden Kişi Sil
4)Kayıtlı Kişileri Görüntüle
5)Uygulamayı Kapat
Seçiminiz: 3
Hangi Değere Göre Silinsin
1)Ad ile Sil
2)Soyad ile Sil
Seçiminiz: 1
Silinecek Kişinin Adını Giriniz: Alper
Silinen Kişi
Adı: Alper
Soyadı: Karaca
Telefon Numarası: +905316237966
Mail Adresi: alpeerkaraca@gmail.com

Ekranı Temizleyip Ana Sayfaya Dönmek İçin Enter Tuşuna Basınız !
```

- Rehberi Görüntüle (Silindiğini Teyit Etmek İçin)

```
alpeerkaraca@pop-os: ~/Desktop/data-structures-assignments/Assignment-1
|-----|
| Lütfen İşlem Seçiminizi Bir Rakam Belirterek Yapınız ! |
|-----|
1)Rehbere Kişi Ekle
2)Rehberde Kişi Ara
3)Rehberden Kişi Sil
4)Kayıtlı Kişileri Görüntüle
5)Uygulamayı Kapat
Seçiminiz: 4
|-----|
| KAYITLI KULLANICILAR |
|-----|
|-----|
| [1. Kişi] |
|-----|
Ad: Test
Soyad: Edilen
Telefon Numarası: +905555555
E-Mail Adresi: test@test.com

Ekranı Temizleyip Ana Sayfaya Dönmek İçin Enter Tuşuna Basınız !
```

KAYNAKÇA:

<https://www.geeksforgeeks.org/data-structures/linked-list/>

<https://iq.opengenus.org/detect-operating-system-in-c/>

<https://stackoverflow.com/questions/45526694/c-scanf-and-whitespaces-ive-tried-almost-everything-currently-learning-abo>

https://www.tutorialspoint.com/c_standard_library/c_function_fgets.htm

<https://cplusplus.com/reference/cstdlib/strtof/>

<https://learn.microsoft.com/tr-tr/cpp/preprocessor/hash-ifdef-and-hash-ifndef-directives-c-cpp?view=msvc-170>

<https://www.programmingsimplified.com/c/conio.h/clrscr>

<https://stackoverflow.com/questions/44679195/how-to-send-command-to-linux-command-from-c-program>

<https://www.programiz.com/c-programming/c-goto-statement>