



***BLM 103- Algoritmalar ve Programlama I  
2019-2020 Güz Dönemi***

***Kriptografi: Sezar Şifreleme ve Çözme***

***Proje Teslim Raporu  
14 Ocak 2020***

***Aysen İpek Çakır, İrem Kalkanlı***

---

<b>1</b>	<b>GİRİŞ .....</b>	<b>1</b>
<b>1.1</b>	<b>Projenin Amacı.....</b>	<b>1</b>
<b>1.2</b>	<b>Proje Ekibi.....</b>	<b>1</b>
<b>2</b>	<b>GELİŞTİRİLEN UYGULAMA .....</b>	<b>2</b>
<b>2.1</b>	<b>Kullanılan Araçlar .....</b>	<b>2</b>
<b>2.2</b>	<b>Tasarım .....</b>	<b>2</b>
<b>3</b>	<b>SONUÇLAR.....</b>	<b>10</b>

# 1 Giriş

## 1.1 Projenin Amacı

Sezar şifreleme eski yıllarda savaş zamanında düşmanların mesajları anlamaması için yapılmış bir şifrelemedir. Projemizin amacı kullanıcıdan alınan anahtar sayı ile bu şifrelemenin yapılabileceği ve çözülebileceği bir kod yazmaktır.

## 1.2 Proje Ekibi

İrem KALKANLI(Proje Ekip Sorumlusu):

Okul numarası:190301007

Doğum Tarihi:15.01.2000

Doğum Yeri: İstanbul

Mezun Olduğu Lise: Ataşehir 3 Doğa Koleji

Aysen İpek ÇAKIR:

Okul numarası:190301001

Doğum Tarihi:20.03.2001

Doğum Yeri: Malatya

Mezun Olduğu Lise: Fethi Gemuhluoğlu Fen Lisesi

## 2 Geliştirilen Uygulama

### 2.1 Kullanılan Araçlar

Tasarım sürecinde Visual Studio kullanılmıştır. Visual Studio, birçok programlama dilini kullanarak program, uygulama ya da web sitesi yapabileceğiniz bir IDE yani entegre geliştirme ortamıdır. Microsoft Windows için bilgisayar programları, web siteleri, web uygulamaları, web hizmetleri ve mobil uygulamalar geliştirmek için kullanılır.

### 2.2 Tasarım

Sezar algoritması, ana mesajın her bir harfini belirtilen anahtar sayı kadar ileri kaydırarak şifreli mesajı oluşturmak üzeredir. Şifreyi çözmek için ise anahtar sayısı ile şifreli mesajdaki her karakterin anahtar sayısı kadar geriye giderek ana mesajı dönüştürmesi üzerinedir.

Tasarım sürecinde ilk olarak şifreleme ve şifre çözme fonksiyonlarını yazdık.

```
int sifrele(char cumle[], int sifrelenicekKarakter) {  
    if ((int)cumle[i] > 64 && (int)cumle[i] < 91) {  
        for (i = 0; cumle[i]; i++) {  
            if (cumle[i] == ' ')  
            {  
                cumle[i] == ' ';  
            }  
            else  
            {  
                char b;  
                b = (char)(((int)cumle[i] - 65 + sifrelenicekKarakter) % 26 + 65);  
                cumle[i] = b;  
            }  
        }  
    }  
}
```

### Şekil 1

Şifreleme işlemini yapan fonksiyonu yazarken öncelikle şifrelenecek karakterin büyük veya küçük harf olmasına dikkat ederek farklı işlemler yaptırıldı. Şekil 1 deki kod parçası şifreleme fonksiyonunda büyük harflere yapılacak olan işlemi göstermektedir.

Öncelikle boşluk karakterinin şifrelenmemesi için ayrı bir işlem yaptırıldı. Sonrasında dosyadan okunan her harfi tek tek kullanıcıdan alınan anahtar sayıya göre şekil 1 deki işlem yaptırılarak şifrelenmiş haline getirdik ve cumle[] dizisi yerine şifrelenmiş cümleyi yazdırdık. Böylelikle şifreleme işi tamamlanmış oldu.

```
else if ((int)cumle[i] > 96 && (int)cumle[i] < 123) {  
    for (i = 0; cumle[i]; i++) {  
        if (cumle[i] == ' ')  
        {  
            cumle[i] == ' ';  
        }  
        else {  
            char b;  
            b = (char)(((int)cumle[i] - 97 + sifrelenicekKarakter) % 26 + 97);  
            cumle[i] = b;  
        }  
    }  
}
```

## Şekil 2

Şekil 2 deki kod parçası şifreleme fonksiyonunda küçük harflere yapılacak olan işlemi göstermektedir. Öncelikle boşluk karakterinin şifrenmemesi için ayrı bir işlem yaptırıldı. Sonrasında dosyadan okunan her harfi tek tek kullanıcıdan alınan anahtar sayıya göre şekil 2 deki işlem yaptırılarak şifrenmiş haline getirdik ve cumle[] dizisi yerine şifrenmiş cümleyi yazdırdık. Böylelikle şifreleme işi tamamlanmış oldu.

```
int sifrecoz(char cumle[], int sifrelenicekKarakter) {  
    if ((int)cumle[i] > 64 && (int)cumle[i] < 91) {  
        for (i = 0; cumle[i]; i++) {  
            if (cumle[i] == ' ')  
            {  
                cumle[i] = ' ';  
            }  
            else  
            {  
                char b;  
                b = (char)(((int)cumle[i] - 65 - sifrelenicekKarakter % 26 + 26) % 26) + 65;  
                cumle[i] = b;  
            }  
        }  
    }  
}
```

### Şekil 3

Şifre çözme işlemini yapan fonksiyonu yazarken öncelikle şifresi çözülecek karakterin büyük veya küçük harf olmasına dikkat ederek farklı işlemler yaptırıldı. Şekil 3 deki kod parçası şifreçöz fonksiyonunda büyük harflere yapılacak olan işlemi göstermektedir.

Öncelikle boşluk karakterinin şifresinin çözülmemesi için ayrı bir işlem yaptırıldı. Sonrasında dosyadan okunan her harfi tek tek kullanıcıdan alınan anahtar sayıya göre şekil 3 deki işlem yaptırılarak şifresini çözdürdük ve cumle[] dizisi yerine şifresi çözülmüş cümleyi yazdırdık. Böylelikle şifre çözme işi tamamlanmış oldu.

```
    }  
    else if ((int)cumle[i] > 96 && (int)cumle[i] < 123) {  
        for (i = 0; cumle[i]; i++) {  
            if (cumle[i] == ' ')  
            {  
                cumle[i] = ' ';  
            }  
            else {  
                char b;  
  
                b = (char)((((int)cumle[i] - 97 - sifrelenicekKarakter % 26 + 26) % 26) + 97;  
  
                cumle[i] = b;  
            }  
        }  
    }  
}
```

#### Şekil 4

Şekil 4 deki kod parçası şifreçöz fonksiyonunda küçük harflere yapılacak olan işlemi göstermektedir.

Öncelikle boşluk karakterinin şifresinin çözülmemesi için ayrı bir işlem yaptırдық. Sonrasında dosyadan okunan her harfi tek tek kullanıcıdan alınan anahtar sayıya göre şekil 4 deki işlem yaptırılarak şifresini çözdürdük ve cumle[] dizisi yerine şifresi çözülmüş cümleyi yazdırdık. Böylelikle şifre çözme işi tamamlanmış oldu.



Fonksiyon oluřturma iřlemlerimiz bittikten sonra bir main fonksiyonu atıkt. ncelikle kullanııcıdan hangi iřlemi yapmak istediđini belirten sayıyı aldık. Őifreleme iřlemi iin 1 sayısını Őifre özme iřlemi iin 2 sayısını belirledik.

```
int main(void)
{
    FILE* dosya;
    FILE* dosya2;
    char cumle[250];
    int menuSecim, sifrelenicekKarakter;
    printf("Lutfen islem turunu seciniz:\nSifrelemek icin ->1\nSifre Cozmek icin ->2");
    scanf_s("%d", &menuSecim);
    if (menuSecim == 1) {

        int rtn = fopen_s(&dosya, "giris.txt", "r");
        fopen_s(&dosya2, "sifreli.txt", "w");
        if (rtn != 0) {
            return 0;
        }
        printf("Kac karaktere gore sifrelenmesini istersiniz ? :");
        scanf_s("%d", &sifrelenicekKarakter);
        while (fscanf_s(dosya, "%[^\n]", cumle, 250) != EOF)
        {

            sifrele(cumle, sifrelenicekKarakter);

            fprintf(dosya2, "%s", cumle);

        }
        fclose(dosya);
        fclose(dosya2);
    }
}
```

## Őekil 5

Őekil 5 deki kod parası Őifreleme seildiđi zaman yapılan iřlemin gerekleřtirildiđi kod parasıdır. ncelikle kullanııcıdan ka karaktere gre Őifrelenmesini istediđi yani anahtar sayı alınır. Sonrasında giris.txt dosyasını okuma iřlemi gerekleřtirilir ve daha sonra Őifrele fonksiyonu ađırılır. Őifreleme iřlemi gerekleřtirildikten sonra Őifrelenen cmleler sifreli.txt adlı dosyaya yazılır. Bylelikle Őifreleme iřlemi gerekleřtirilmiř olur.

```
else if (menuSecim == 2) {
    fopen_s(&dosya2, "sifreli.txt", "r");
    int rtn = fopen_s(&dosya, "cikis.txt", "w");
    if (rtn != 0) {
        return 0;
    }
    while (fscanf_s(dosya2, "%[^\\n]", cumle, 250) != EOF)
    {
        printf("Kac karaktere gore cozulmesini istersiniz ? :");
        scanf_s("%d", &sifrelenicekKarakter);
        sifrecoz(cumle, sifrelenicekKarakter);

        fprintf(dosya, "%s", cumle);
    }
    fclose(dosya2);
    fclose(dosya);
}
```

## Şekil 6

Şekil 6 daki kod parçası şifre çözme seçildiği zaman yapılan işlemin gerçekleştirildiği kod parçasıdır. Öncelikle kullanıcıdan kaç karaktere göre şifrenin çözülmesinin istediği yani anahtar sayı alınır. Sonrasında sifreli.txt dosyasını okuma işlemi gerçekleştirilir ve daha sonra şifrele fonksiyonu çağırılır. Şifre çözme işlemi gerçekleştirildikten sonra şifrelenen cümleler cikis.txt adlı dosyaya yazılır. Böylelikle şifre çözme işlemi gerçekleştirilmiş olur.

```
else  
{  
    printf("Hatali secim.");  
}
```

### Şekil 7

Şekil 7 de kullanıcıdan hangi işlemi gerçekleştirmek istediği sorulduğunda 1 veya 2 sayısını girilmezse kullanıcıyı hatalı seçim yaptığı belirten kod parçası gösterilmiştir.

### 3 Sonular

Proje sonucunda Sezar Őifreleme ve Őifreleme kodu yazma hakkında ilgi sahibi olmuŐ olduk. Ayrıca birden fazla kiŐiyle kod yazmayı ve bir kodla gnlerce uŐraŐıp geliŐtirmeyi ğrenmiŐ olduk. İleride yapacaėımız projelerde nelerle karŐılaŐacaėımızla alakalı deneyime sahip olduk.

Hazırlanan sunum video'su adresi:

<https://www.youtube.com/watch?v=S0dbjRSvVsA&feature=youtu.be>

Dosyaların github adresi: <https://github.com/iremkalkanli/BLM-103-Sezar-Sifreleme>