



VERİ YAPILARI VE ALGORİTMALAR

ÖDEV-3

**Boyer Moore Horspool
Algoritmasını Kullanarak String
Search Ve Replace Etme**

PROGRAMIN ADIM ADIM AÇIKLANMASI VE YÜRÜTÜLMESİ:

Programın temelinde 3 ana işlev yer alır bunlar;

1-Kullanıcıdan aranacak kelime,değiştirilecek yeni kelime,dosya ismi(.txt formatında) ve aramanın case sensitive olup olmayacağı kararı bilgisi alınmaktadır(0 veya 1).

2-Kullanıcıdan alınan bilgilerden sonra ismi belirtilen dosyadaki tüm karakterler tek tek bir char dizisine alınır.

Verilen kelime boyer moore horspool algoritması ile cümlede(text dizisinin içinde)bulunur ve değiştirilmesi istendiği yeni kelime ile değiştirilir.

Değiştirme işlemi ek bir dizi kullanılmadan yapılmıştır ve temelinde 3 durum vardır;

a)Aranan kelimenin değişecek kelimedenden büyük olması:

aranan kelime bulunduğu indisin başından yeni eklenicek kelimenin boyutu kadar uzağından başlanılır,textin sonuna kadar iki kelime arasındaki fark kadar sola shift edilir.Böylelikle yazılacak yeni kelimedenden sonraki kısım giderilmiş olacak.Shift işleminden sonra yeni kelime direkt aranan kelimenin bulunduğu indisin başından başlanarak yazılır.

b)Aranan kelimenin değişecek kelimedenden küçük olması:Değiştirilecek kelime aranan kelimedenden büyük ise öncelikle dizi,dizinin en sonundan başlanarak aranan kelimenin bulunduğu yere kadar dizi,iki kelimenin farkı kadar sağ shift edilir(genişletilir)bulunan indisin başına gelindiğinde durulur çünkü o indisin öncesi için herhangi bir değişikliğe gerek yoktur.Shift işleminden sonra yeni kelime aranan kelimenin bulunduğu indis itibari başlanarak direkt olarak yazılır.

c)Aranan kelime ve değişecek kelimenin uzunluklarının eşit olması:Bu durumda yapılan tek işlem vardır.Aranan kelimenin bulunduğu indisten başla,aranan kelime yada yeni kelime(ikisininde uzunluğu aynı fark etmez fakat kodda yeni kelime boyutuna kadar gidildi.)nin boyutu kadar ilerle ve yeni kelimeyi yaz.

Aşağıdaki fotoğrafta değiştirme(replace)işleminin mantığını daha detaylı anlatan bir görsel verilmiştir;

I Believe I can.....

I think~~ve~~ I can.....

I think I can.....

FIND REPLACE

Believe > think

Fark=2

I Believe I can fly.....

I Believe I c an fly.

I Believe I Believe fly

FIND REPLACE

can < Believe

Fark=5

I Believe I can...

I Believe I can

FIND REPLACE

Believe = Believe

Fark= 0

Böylelikle fonksiyonumuz arama ve değiştirme görevlerinden sonra işlemine son vermiş olur.

3-İşlem sonunda ise kaç adet değişim yapıldı, arama ve yer değiştirme olarak işlem süresini(milisaniye cinsinden)ekrana yazıp kullanıcıyı bilgilendirir.

Programda sıkça kullanılan değişkenler ve anlamları;

char fName[255]:Kullanıcıdan alınan dosyanın ismi(.txt formatında)

char find[255]:Kullanıcı tarafından aranması istenilen kelimeyi tutan dizi.

char replace[255]: Kullanıcı tarafından alınan aranan kelime ile değişecek yeni kelimeyi tutan dizi.

char text[25500]:Dosyadan okunan bütün metni tutan dizi.

int cSensitive:Kullanıcı tarafından aramanın case sensitive olup olmayacağı kararını tutan değişken(EVET=1,HAYIR=0)

struct timespec start:Zamanlama için clockumuzu başlattığımızdaki değerleri tutacak olan struct tipinde değişkenimiz.

struct timespec finish: Clocku sonlandırdığımızda oluşan değerleri tutacak olan struct tipinde değişkenimiz.

int index:Fonksiyon içerisinde arama işlemi sırasındaki mevcut indisi tutan değişken

int j:Arama sırasında shift miktarını tutan değişken.

int counter=0:Toplam bulma ve değişme miktarını tutan değişken.

int bmt[256]:Bad match tableimizi tutan değişken.

Programda main fonksiyonu hariç 1 fonksiyon bulunmaktadır;

1- **void findAndReplace(char text[],char find[],char replace[],int cSensitive):**fonksiyona gönderilen parametreler;

-Metinimizin tamamı: **char text[]**

-Aranacak kelimenin tamamı: **char find[]**

-Replace edilecek yeni kelimenin tamamı: **char replace[]**

Bu Fonksiyonda yapılan işlemler ise;

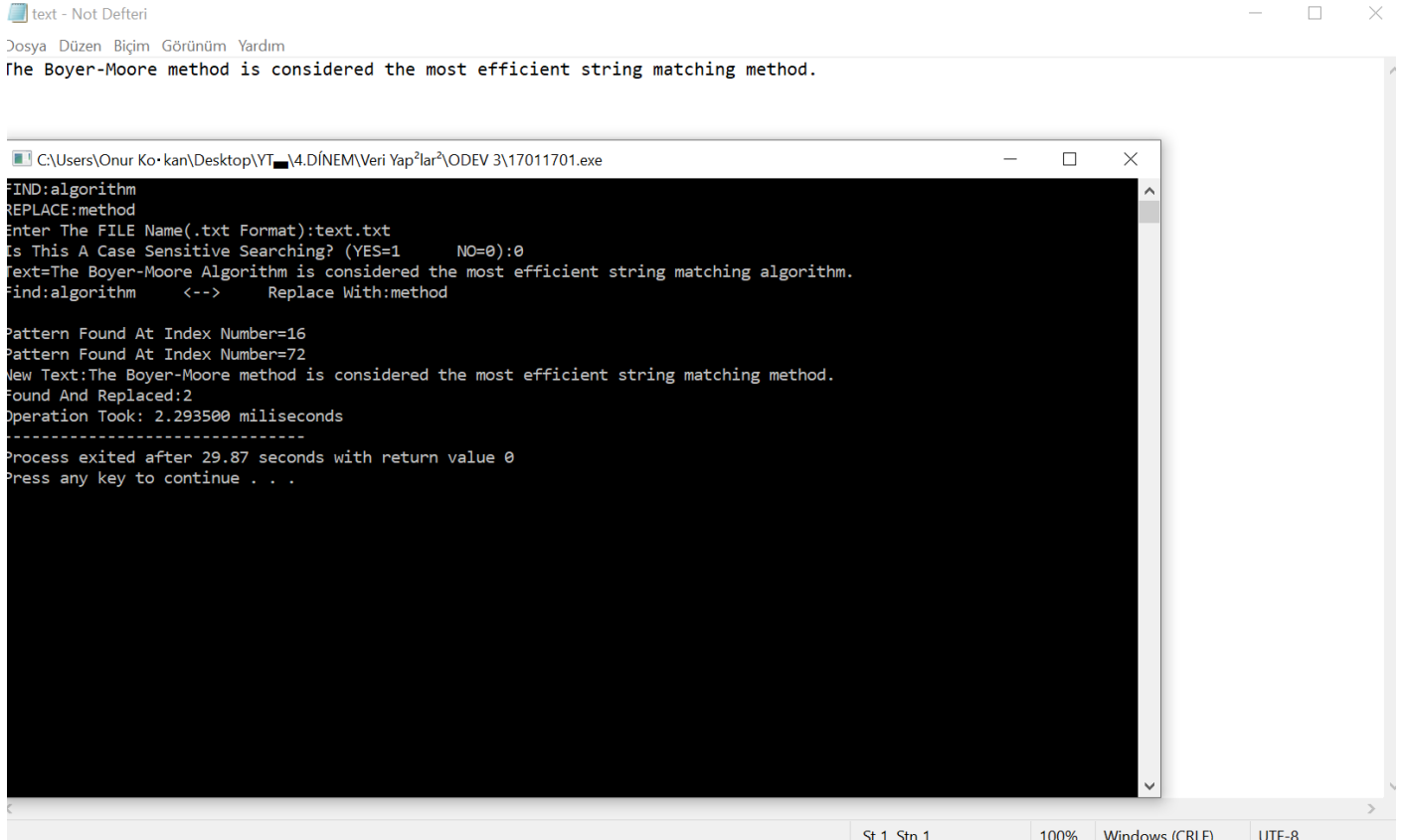
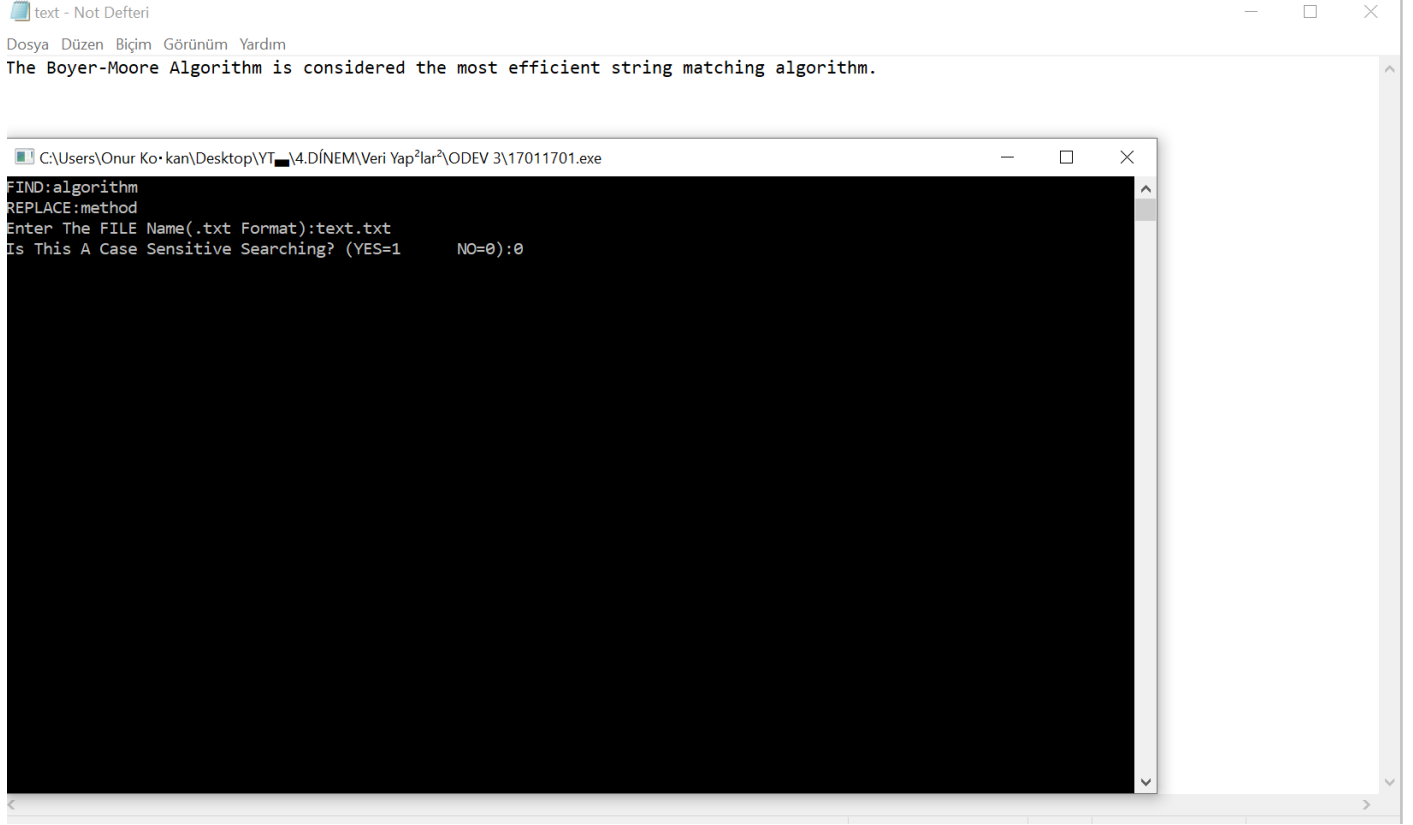
-Boyer Moore Horspool algoritması için gerekli olan 0-256 arası karakterlere(ASCII tablosunda bulunan) aranacak kelimenin uzunluğu verilir,

-Case sensitive özelliği kontrol edilir ve kelime aranır.

-Kelime bulunduktan sonra yeni değiştirilmesi istenen kelime ile(replace) değiştirilir.

ÖRNEKLERİN EKRAN GÖRÜNTÜLERİ:

Örnek 1:



The Boyer-Moore Algorithm is considered the most efficient string matching algorithm.

C:\Users\Onur Koçan\Desktop\YT\4.DİNEM\Veri Yapılar\ODEV 3\17011701.exe

```
FIND:algorithm
REPLACE:method
Enter The FILE Name(.txt Format):text.txt
Is This A Case Sensitive Searching? (YES=1 NO=0):1
```

St 1, Str 86

100%

Windows (CRLF)

UTF-8

The Boyer-Moore Algorithm is considered the most efficient string matching method.

C:\Users\Onur Koçan\Desktop\YT\4.DİNEM\Veri Yapılar\ODEV 3\17011701.exe

```
FIND:algorithm
REPLACE:method
Enter The FILE Name(.txt Format):text.txt
Is This A Case Sensitive Searching? (YES=1 NO=0):1
Text:The Boyer-Moore Algorithm is considered the most efficient string matching algorithm.
Find:algorithm <--> Replace With:method

Pattern Found At Index Number=75
New Text:The Boyer-Moore Algorithm is considered the most efficient string matching method.
Found And Replaced:1
Operation Took: 7.234400 milliseconds
-----
Process exited after 23.15 seconds with return value 0
Press any key to continue . . .
```

St 1, Str 1

100%

Windows (CRLF)

UTF-8

Örnek 2:

text - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

Wayne went to Wales to watch walruses.

C:\Users\Onur Ko•kan\Desktop\YT\4.DİNEM\Veri Yapılar\ODEV 3\17011701.exe

```
FIND:went to
REPLACE:visited
Enter The FILE Name(.txt Format):text.txt
Is This A Case Sensitive Searching? (YES=1 NO=0):0
```

text - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

Wayne visited Wales to watch walruses.

C:\Users\Onur Ko•kan\Desktop\YT\4.DİNEM\Veri Yapılar\ODEV 3\17011701.exe

```
FIND:went to
REPLACE:visited
Enter The FILE Name(.txt Format):text.txt
Is This A Case Sensitive Searching? (YES=1 NO=0):0
Text=Wayne went to Wales to watch walruses.
Find:went to <--> Replace With:visited

Pattern Found At Index Number=6
New Text:Wayne visited Wales to watch walruses.
Found And Replaced:1
Operation Took: 2.227500 milliseconds
-----
Process exited after 39.67 seconds with return value 0
Press any key to continue . . .
```

text - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

Wayne went to Wales to watch walruses.

C:\Users\Onur Ko kan\Desktop\YT\4.DİNEM\Veri Yapılar\ODEV 3\17011701.exe

```
FIND:went to
REPLACE:visited
Enter The FILE Name(.txt Format):text.txt
Is This A Case Sensitive Searching? (YES=1      NO=0):1
```

St 1, Stn 39

100%

Windows (CRLF)

UTF-8

text - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

Wayne visited Wales to watch walruses.

C:\Users\Onur Ko kan\Desktop\YT\4.DİNEM\Veri Yapılar\ODEV 3\17011701.exe

```
FIND:went to
REPLACE:visited
Enter The FILE Name(.txt Format):text.txt
Is This A Case Sensitive Searching? (YES=1      NO=0):1
Text=Wayne went to Wales to watch walruses.
Find:went to  <-->  Replace With:visited

Pattern Found At Index Number=6
New Text:Wayne visited Wales to watch walruses.
Found And Replaced:1
Operation Took: 1.993100 milliseconds
-----
Process exited after 18.11 seconds with return value 0
Press any key to continue . . .
```

St 1, Stn 1

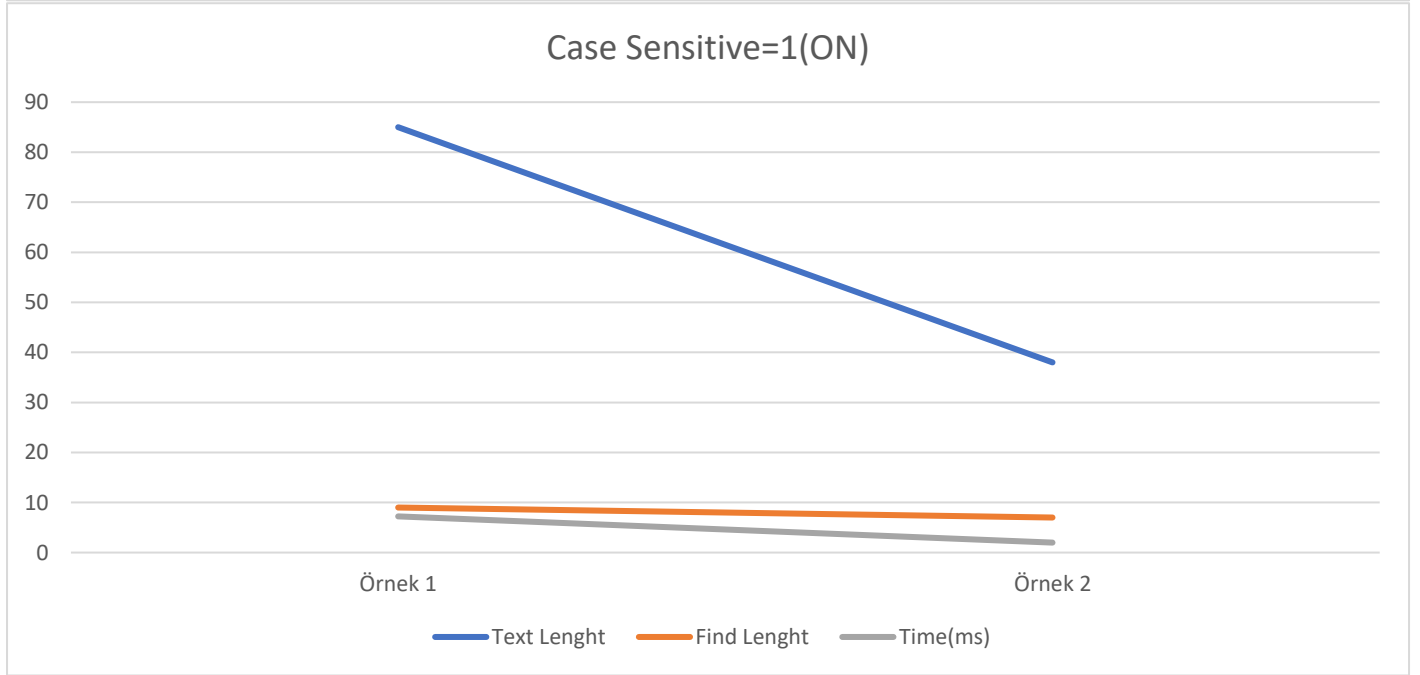
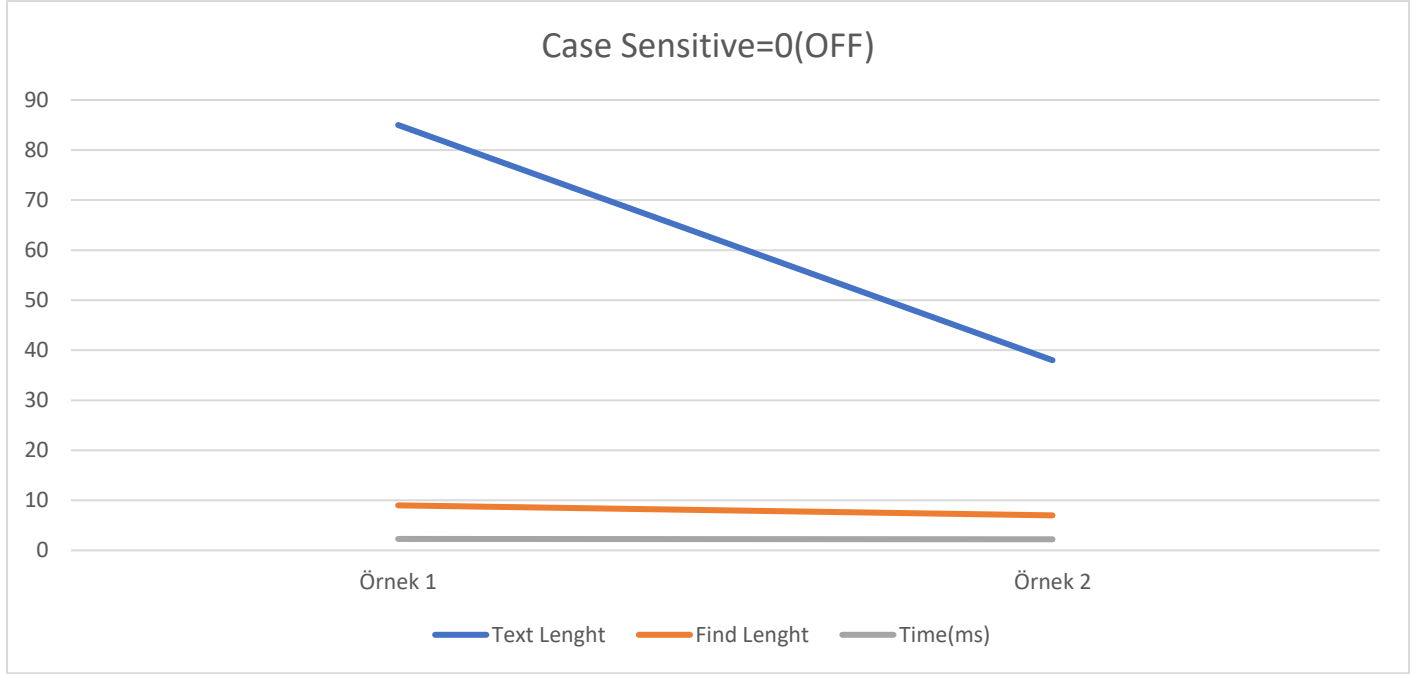
100%

Windows (CRLF)

UTF-8

ÖRNEKLERİN ÇALIŞMA ZAMANLARI GRAFİKLERİ:

Örnek No	Text Lenght	Find Lenght	Case Sensitive Option	Time(ms)
1	85	9	0	2,29ms
1	85	9	1	7,23ms
2	38	7	0	2,22ms
2	38	7	1	1,99ms



PROGRAMIN KODLARI:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <time.h> //TIME FONKSIYONLARINI KULLANMAMIZ ICIN GEREKLI KUTUPHANE

FILE *dosya; //AYNI ISIMLI DOSYA DEGISKENINI KAYBETMEMEK GLOBAL TANIMLANDI.
char fName[255]; //DOSYA ISMININ HER YERDE AYNI OLMASI ICIN GLOBAL TANIMLANDI.

void findAndReplace(char text[], char find[], char replace[], int cSensitive){
    int index, j, counter=0;
    int bmt[256];
    //BAD MATCH TABLE'IN OLUSTURULMA ISLEMLERI;
    int k;
    for(k = 0; k < 256; ++k) //TUM ASCII TABLOSUNDAKI DEGERLER KADAR DOLANIR VE
        HEPSINE ARANACAK METININ UZUNLUGU VERILIR, TABLO OLUSTURULUR
        bmt[k] = strlen(find);

    if(cSensitive==1){ //CASE SENSITIVE AKTIF DEGIL IKEN;
        for(k = 0; k < strlen(find) -1; ++k)
            bmt[find[k]] = strlen(find) - k - 1;
    }
    else{ //CASE SENSITIVE AKTIF IKEN;
        for(k = 0; k < strlen(find) -1; ++k){
            bmt[find[k]] = strlen(find) - k - 1;
            if(find[k]<=90) //ASCII KARAKTER TABLOSUNA GORE ELDE OLAN MEVCUT SAYI BUYUK
                bmt[find[k]+32]= strlen(find) - k - 1; //KUCUK-BUYUK HARFLERIN
                TABLODAKI SABIT FARKI KADAR EKLEKI, AYNI SAYININ KUCUK DEGERINE SAHIP OL
            else
                bmt[find[k]-32]= strlen(find) - k - 1; //KUCUK ISE ASCII TABLOSUNDA 32
                INDIS GERIYE GIDEREK AYNI KARAKTERIN BUYUGUNE SAHIP OL.
        }
    }
    //BAD MATCH TABLE'IN OLUSTURULMA ISLEMLERININ SONU.

    index=0; //0 DEGERI INDISI TUTAN INDEX DEGISKENIMIZE BASLANGIC DEGERI OLARAK
    VERILIR
    while(index<=(strlen(text)-strlen(find))){ //INDIS DEGERI TUM TEXTIN ILE ARANAN
        KELIMENIN FARKINDAN KUCUK OLDUGU SURECE DEVAM ET.
        //BOYLELİKLE KONTROL ESNASINDA ARANAN
        KELIMENIN BOYUTUNDAN OTURU TEXT DIZISINDE TASMA OLMAS.
        j=strlen(find)-1; //ARANAN KELIMENIN BIR EKSIGINI AL
        while(((find[j]==text[index+j]) || ((fabs(find[j]-text[index+j])==fabs('Z'-
        'z')) && (cSensitive>0 ? 0 : 1))) && (j>=0))
            j--; //J DEGISKENI 1 AZALTILIR.
        //37.SATIRDAKI WHILE:
        //ARANAN KELIMENIN J.INDISININ TEXTIN MEVCUT INDEX + J.INDISE ESIT OLMASI
        DURUMU VEYA;
        //YINE AYNI IFADENIN FARKLARININ MUTLAK DEGERLERI, ASCII TABLOSUNDA BULUNAN
        HERHANGI IKI HARFIN BUYUK VE KUCUK HALLERININ FARKININ MUTLAK DEGERINE ESITMI? VE;
        //CASE SENSITIVE ICIN GIRILEN KARAR DEGISKENININ ZITTI ALINIR " (cSensitive>0 ?
        0 : 1) " IFADESINDEN GELICEK OLAN 0 SONUCU BUYUK KUCUK HARF FARKINI INCELIYEN IFADEYIDE
        YOK EDICEGI ICIN
        //CASE SENSITIVE OLMADAN ARAMA YAPILACAKTIR.

        //BU ANLATILAN IFADELERIN HEPSI BIR PARANTEZDE ILK IFADE(ILK OPERAND) OLARAK
        DUSUNDUGUMUZDE " (j>=0) " IFADESIYLA(İKİNCİ OPERAND) İLE ANDLENİR
        //CUNKU J DEGISKENININ(ARANAN METİN-1 DEGERINI TUTAR) ODAN BUYUK VEYA ESIT
        OLMASI GEREKLİDİR
        //37.SATIRDAKI WHILE ACIKLAMASININ SONU.

        if(j<0){ //J DEGISKENININ 0 IN ALTINA GIRMESİ DEMEK PATTERNIN(FIND) BULUNMASI
        DEMEKTİR. PATTERNIN BULUNDUGU BLOK;
            printf("\nPattern Found At Index Number=%d", index);
```

```

//TEXT ICIN YAPILACAK REPLACE İSLEMLERİ(FIND<->REPLACE);
//3 DURUM OLMALIDIR;

int i;
if(strlen(find)>strlen(replace)){ //1)BULUNAN KELIMENİN KARAKTER SAYISI
DEĞİŞECEK YENİ KELIMENİNKİNDEN BÜYÜK İSE;

    for(i=index+strlen(replace);text[i]!='\0';i++){ //ARANAN KELIMENİN
BULUNDUGU INDEX + YENİ EKLENİCEK KELIMENİN BOYU KADAR İLERİSİNDEN BAŞLAYIP SONUNA KADAR
GİT
        text[i]=text[i+strlen(find)-strlen(replace)]; //TUM DİZİYİ FARKLARI
KADAR SOLA SHIFT ET.
    }

    for(i=0;i<strlen(replace);i++){ //YENİ DEĞİŞECEK KELIMENİN BOYUTU KADAR
İLERLE
        text[index+i]=replace[i]; //PATTERNİ BULDUGUN INDEX+i KADAR YENİ
KELİMEYİ YAZ
    }

}

else if(strlen(find)<strlen(replace)){ //2)YENİ KELİME,DEĞİŞECEK ESKİ
KELİMEDEN BÜYÜK İSE;

    for(i=strlen(text);i>index;i--) //TEXTİN EN SONUNDAN BAŞLAYIP PATTERNİ
BULDUGUMUZ İNDİS DEĞERİNE KADAR GERİ GERİ GELİP DİZİMİZİ SAĞ SHIFT EDİYORUZ.
        text[i+strlen(replace)-strlen(find)]=text[i]; //FARKLARI KADAR
İLERİSİNDEKİ İNDİSDEKİ DEĞERİ=MEVCUT İNDİSDEKİ DEĞERE AT,BOYLELİKLE SAĞ SHIFT EDERKEN
KARAKTER KAYBI YASAMAYIZ.

    for(i=0;i<strlen(replace);i++) //SHIFT İSLEMİNDEN SONRA HER ZAMANKİ
GİBİ YENİ KELİME MEVCUT İNDİSDEN İTİBAREN YAZILMAYA BAŞLANIR.
        text[index+i]=replace[i]; //PATTERNİN BULUNDUGU MEVCUT İNDİS +
I(0.1.2.3.....YENİ KELIMENİN BOYUTU KADAR)DEKİ DEĞERLERE SIRASIYLA YENİ KELİME
YAZILIR.
    }

    else{
        for(i=0;i<strlen(replace);i++){ //3)YENİ KELİME,ESKİ KELİMEYLE AYNI
UZUNLUGA SAHİP İSE;
            text[index+i]=replace[i]; //DİREKT OLARAK PATTERNİN BULUNDUGU
İNDİSDEN BAŞLANARAK YENİ KELİME YAZILIR.
        }
        counter++; //TOTAL BULUNMA VE DEĞİŞTİRME İSLEMLERİNİ TUTAN DEĞİSKEN 1
ARTTIRILIR.
        index+=strlen(replace)+1; //BULUNMA İSLEMLERİNDEN SONRA MEVCUT İNDİS
DEĞERİNİN YENİ KELIMENİN 1 FAZLASI KADAR ARTTIRILMASI GEREKİLİR.

    }

    else{//INDEXİN YENİDEN BELİRLENMESİ İSLEMLERİ;
        if (1>bmt[text[index+strlen(find)-1]]){ //MEVCUT İNDİSİMİZ ARADIGIMIZ
KELIMENİN UZUNLUGUNU EKLEYİP 1 ÇIKARTMAMIZ SONUCUNDA ÇIKAN DEĞER
//İNDİS KABUL EDİLİP,TUM METİNİN
BULUNDUGU TEXT DİZİSİNDE HANGİ KARAKTERE GELİYOR İSE O KARAKTERİN
//BAD MATCH TABLEİMİZDE BULUNAN(0-
256 ARASI ASCII KARAKTERLERİ İÇEREN) DEĞERİNİN 1DEN BÜYÜK VEYA KUCUK OLMA KONTROLÜ;
            index++; //1DEN KUCUK İSE İNDİSİ 1 ARTTIR
        }
        else{
            index+=bmt[text[index+strlen(find)-1]]; //1DEN BÜYÜK İSE İNDİSİ DENK
GELEN İFADE KADAR ARTTIR.
        }
    }
}

```

```

    }
    //FIND VE REPLACE ISLEMLERIMIZ TAMAMIYLA SONA ERDI SIMDI SIRA TEKRAR DOSYAMIZA
    YAZMADA;
    int a=0;
    dosya=fopen(fName,"w"); //AYNI DOSYA ISMINI WRITE MODU ILE ACIYORUZ.
    while(text[a]!='\0'){ //TEXTIN SONUNA GELENE KADAR;
        fputc(text[a],dosya); //KARAKTER KARAKTER DOSYAYA KOY
        a++;
    }
    fclose(dosya); //DOSYAYI KAPAT

    printf("\nNew Text:%s\nFound And Replaced:%d",text,counter); //TEXTIN SON
    HALINI,FOUND VE REPLACE SAYILARINI YAZDIRARAK FONKSIYONU SONA ERDIR.
}

int main(){
    char find[255],replace[255],text[25500];
    int cSensitive;
    struct timespec start; //CLOCK SAYIMI YAPICAGIMIZ ICIN SURE KAYDININ BASLANGIC
    DEGERINI TUTMAK ICIN OLUSTURDUGUMUZ STRUCT TIPINDEKI DEGISKENIMIZ.
    struct timespec finish; //AYNI SEKILDE CLOCK SAYIMI BITTIKTEN SONRAKI KAYIT SONU
    DEGERINI TUTUCAK OLAN DEGISKENIMIZ.

    printf("FIND:");
    gets(find);
    printf("REPLACE:");
    gets(replace);
    printf("Enter The FILE Name(.txt Format):");
    gets(fName); //GLOBA TANIMLANDI BOYLELIKLE FONKSIYON ICERISINDE RAHATLIKLA ISMI
    GECTIGI YERDEN ERISILEBILECEK.

    printf("Is This A Case Sensitive Searching? (YES=1 NO=0):");
    scanf("%d",&cSensitive);

    if(cSensitive==1||cSensitive==0){
        //DOSYADAKI METININ ALINMASI ISLEMI
        int i=0;
        dosya=fopen(fName,"r"); //VERILEN DOSYA ISMINDE DOSYA READ MODU ILE ACILIR
        while(!feof(dosya)){ //DOSYANIN SONUNA GELENE KADAR;
            fscanf(dosya,"%c",&text[i]); //KARAKTER KARAKTER OKU,TEXT DIZISINE
            AT(TUM METIN TEXT DIZISINDE TUTULUR)
            i++;
        }
        fclose(dosya); //DOSYA KAPATILIR.
        //DOSYADAKI METIN ALINDI text ISIMLI DIZIYE ATILDI.
        printf("Text=%s \nFind:%s <--> Replace With:%s\n",text,find,replace);

        clock_gettime(CLOCK_MONOTONIC,&start); //clock_gettime FONKSIYONU ILE SUREYI
        ANLIK YAKALAR VE start DEGISKENINE ATAR.
        findAndReplace(text,find,replace,cSensitive); //***ASIL FONKSIYONUMUZ
        clock_gettime(CLOCK_MONOTONIC,&finish); //IKINCI SUREYI YAKALARIZ VE finish
        DEGISKENINE ATARIZ
        //GEREKLI DONUSUMLER YAPILIP MILI SANIYE CINSINDEN EKRANA YAZDIRILIR,SON
        ALDIGIMIZ ZAMAN KAYDINDAN ILK ALDIGIMIZ ZAMAN KAYDINI CIKARTIRIZ(finish-start)
        printf("\nOperation Took: %lf miliseconds", (double)((long int)(finish.tv_sec-
        start.tv_sec)*100000000 + (finish.tv_nsec-start.tv_nsec))/100000);

    }

    else
        printf("Type Just 1 Or 0 For Case Sensitive Decision...");

    return 0;
}

```