

Burak Boz

18011706

Find And Replace

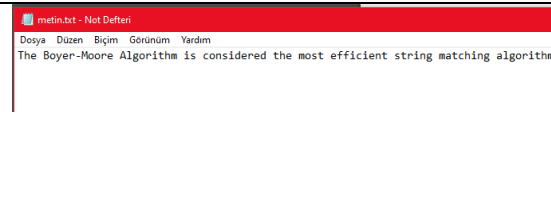
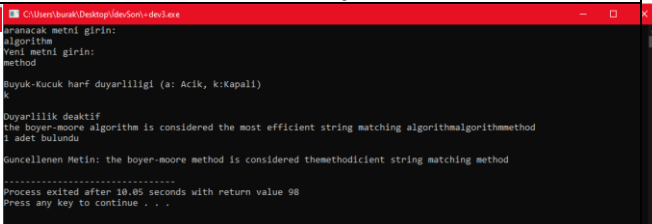
Algoritma:

Öncelikle metin.txt dosyası içerisindeki metin program içerisindeki bilgi[1000] char dizisi içerisine alınır. Kullanıcıdan aranacak kelime istenir ve okunur. Güncel kelime kullanıcıdan istenir ve okunur. Daha sonra kullanıcıdan büyük-küçük harf duyarlılığı istenir. Girilen değer "a" veya "k" olarak duyarlılık değişkeni içerisine atılır.

Kullanılan fonksiyonlar ara, duyarsızAra, güncelle ve duyarsızGuncelle dir. Kullanıcı eğer duyarsızlığı k olarak seçerse duyarsızAra ve duyarsızGuncelle fonksiyonları çalışır. Aksi durumda diğer iki fonksiyon çalışır.

Seçilen verilere göre aranan path ara fonksiyonu içerisine tüm metinle birlikte gönderilir. Ara fonksiyonu eğer kelimeyi bulabilirse güncelle fonksiyonuna başlangıç konumunu gönderir ve fonksiyon metni günceller.

Girdi ve çıktılar:

Girdi	Sonuç
	

Kodlar:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <limits.h>
```

```
#define MAXCHAR 1000
```

```
char *result; //Güncellenen metin bu pointer'da tutulacak
```

```
int guncelle(int konum,char tumText[MAXCHAR],char eskiText[MAXCHAR],char yeniText[MAXCHAR])
```

```

{
    const char *s=tumText;

    const char *oldW=eskiText;

    const char *newW=yeniText;


    int i, cnt = 0;

    int newWlen = strlen(yeniText);

    int oldWlen = strlen(eskiText);


    for (i = 0; s[i] != '\0'; i++)
    {
        if (strstr(&s[i], oldW) == &s[i])
        {
            cnt++;


            i += oldWlen - 1;
        }
    }


    result = (char *)malloc(i + cnt * (newWlen - oldWlen) + 1);


    i = 0;
    while (*s)
    {
        if (strstr(s, oldW) == s)
        {
            strcpy(&result[i], newW);

            i += newWlen;

            s += oldWlen;
        }
        else

```

```

        {
            result[i++] = *s++;
        }

    }

    result[i] = '\0';
    //printf("Guncellenen metin:%s",result);
    return 1;

}

int guncelleDuyarsiz(int konum,char tumText[MAXCHAR],char eskiText[MAXCHAR],char
yeniText[MAXCHAR])
{
    int a=0;
    while( tumText[a] ) {
        tumText[a]=putchar(tolower(tumText[a]));
        a++;
    }

    a=0;
    while( eskiText[a] ) {
        eskiText[a]=putchar(tolower(eskiText[a]));
        a++;
    }

    a=0;
    while( yeniText[a] ) {
        yeniText[a]=putchar(tolower(yeniText[a]));
        a++;
    }
}

```

```
const char *s=tumText;

const char *oldW=eskiText;

const char *newW=yeniText;
```

```
int i, cnt = 0;

int newWlen = strlen(yeniText);

int oldWlen = strlen(eskiText);
```

```
for (i = 0; s[i] != '\0'; i++)
{
    if (strstr(&s[i], oldW) == &s[i])
    {
        cnt++;

        i += oldWlen - 1;
    }
}
```

```
result = (char *)malloc(i + cnt * (newWlen - oldWlen) + 1);
```

```
i = 0;

while (*s)
{
    if (strstr(s, oldW) == s || (strstr(s, oldW)+32) == s || strstr(s, oldW) == (s+32))
    {
        strcpy(&result[i], newW);

        i += newWlen;

        s += oldWlen;
    }

    else
```

```

    {
        result[i++] = *s++;
    }

}

result[i] = '\0';

return 1;

}

int ara(char yeniText[MAXCHAR], char eskiText[MAXCHAR],char tumText[MAXCHAR])
{
    char oncesi[MAXCHAR];
    char sonrasi[MAXCHAR];
    char guncel[MAXCHAR];
    int bulunanAdet=0;

    int i = 0;

    for(i = 0;i<(strlen(tumText)-strlen(eskiText));i++)
    {
        int sayac=i,kontrol=0,sayac2=0;

        while(tumText[sayac]==eskiText[sayac2])
        {
            sayac2++;

            if(sayac2==strlen(eskiText))
            {
                kontrol=1;

```

```

        }
        sayac++;

    }
    if(kontrol==1)
    {
        int a = guncelle(i,tumText,eskiText,yeniText);
        if(a==1)
        {
            bulunanAdet++;
        }
    }
}
printf("\n%d adet bulundu\n",bulunanAdet);
return 1;
}

```

```

int araDuyarsiz(char yeniText[MAXCHAR], char eskiText[MAXCHAR],char tumText[MAXCHAR])
{
    char oncesi[MAXCHAR];
    char sonrasi[MAXCHAR];
    char guncel[MAXCHAR];
    int bulunanAdet=0;

    int i = 0;

    for(i = 0;i<(strlen(tumText)-strlen(eskiText));i++)
    {
        int sayac=i,kontrol=0,sayac2=0;

```

```

        while((tumText[sayac]==eskiText[sayac2]) ||
((tumText[sayac]+32)==eskiText[sayac2]) || (tumText[sayac]==(eskiText[sayac2]+32)) )
        {

            sayac2++;

            if(sayac2==strlen(eskiText))
            {
                kontrol=1;
            }
            sayac++;

        }
        if(kontrol==1)
        {
            int a = guncelleDuyarsiz(i,tumText,eskiText,yeniText);
            if(a==1)
            {
                bulunanAdet++;
            }
        }
    }
    printf("\n%d adet bulundu\n",bulunanAdet);
    return 1;
}

```

```

main()
{
    int konum=0;
    int arananUzunluk=0;
    char aranan[MAXCHAR];
    char yeniMetin[MAXCHAR];

```

```
printf("aranacak metni girin:\n");
gets(aranan);
arananUzunluk=strlen(aranan);
printf("Yeni metni girin:\n");
gets(yeniMetin);
FILE *fp=fopen("metin.txt","r");
char bilgi[MAXCHAR];
while(fgets(bilgi, MAXCHAR, fp) != NULL){
    fscanf(fp,"%s",bilgi);

}

char duyarlilik="";
printf("\nBuyuk-Kucuk harf duyarlilik (a: Acik, k:Kapali)\n");
duyarlilik = getchar();
if(duyarlilik=='a' || duyarlilik=='A')
{
    printf("\nDuyarlilik aktif\n");
    int kontrol = ara(yeniMetin,aranan,bilgi);
    if(kontrol==1)
    {
        printf("\nGuncellenen Metin: %s \n",result);

    }
    else
    {
        printf("Olmadi");
    }
}
else
```



```
{  
  
    printf("\nDuyarlilik deaktif\n");  
    int kontrol = araDuyarsiz(yeniMetin,aranan,bilgi);  
    if(kontrol==1)  
    {  
        printf("\nGuncellenen Metin: %s \n",result);  
    }  
    else  
    {  
        printf("Olmadi");  
    }  
}  
  
}
```