

Ad:Abdul Jalal

Soyad:Lawal

Öğrenci No:200061082

Yığın Veri Yapısı Kullanılarak HTML Taglarının Kontrolünün Yapılması

❖ Program üç tane fonksiyondan yazıldı

- add_to_stack()
- delete_from_stack()
- print_stack()
- int main()

❖ add_to_stack Fonksiyonu

- Fonksiyon char tipinde ve tag adında bir parametre alıyor
- yeni_düğüm adındaki değişkenin için malloc ile yer tasvir edip alınan tag parametresini yeni_düğümün içine yerleştirilir
- yeni_düğümün işaretçisinin içine yerleştirmeden önce düğüm boş olup olmadığını kontrol edilir eğer boş ise elemanı ilk düğüne konulur boş değilse yine düğümün başına eklenir stack veri yapısı olduğu için

```
void add_to_stack(char tag[100]){
    yigin *yeni_dugum = (yigin*)malloc(sizeof(yigin));
    strcpy(yeni_dugum->html_tag, tag);
    if (sta == NULL){
        yeni_dugum->next = NULL;
    }
    else {
        yeni_dugum->next = sta;
    }
    sta = yeni_dugum;
}
```

❖ delete_from_stack Fonksiyonu

- Fonksiyon hiç bir parametre almıyor
- önce yığının boş olduğu şartını kontrolünü sağlayıp

fonksiyon çalıştırılır

- yigin tipinde bir temp değişkeni sta değişkenine atanır
- sta değişkeni ise bir sonraki düğümüne atanır
- son olarak temp değişkeninin içeriğini free ederek düğüm silinir

```
void delete_from_stack() {  
    if (sta != NULL) {  
        yigin *temp = sta;  
        sta = sta -> next;  
        free(temp);  
    }  
}
```

❖ print_stack Fonksiyonu

- Fonksiyonun amacı stack ta elemanları yazdırmaktır
- Fonksiyon herhangi bir parametre almıyor
- Önce stackin boş olup olmadığının kontrolü yapılır boş ise eğer boş olduğunun bilgisini verilir değil ise stacktaki elemanları yazdırır ekrana

```
void print_stack() {  
    if (sta != NULL) {  
        yigin *temp = sta;  
        while (temp != NULL) {  
            printf("%s\t", temp -> html_tag);  
            temp = temp -> next;  
        }  
    }  
    else {  
        printf("stack bos");  
    }  
    printf("\n");  
}
```

❖ int main() Fonksiyonu

- Dosyayı açıp içeriğini okuyarak ptr değişkeninin içine atadım
- Dosyanın boş olup olmadığının kontrolü yapıp dosya boş ise eğer boş olduğunun bilgisi verir

- fscanf fonksiyonu ile ptr değişkeninin içindeki dosya bilgisinin tek kelime olacak şekilde alıp buf değişkeninin içine atadım
- aldım kelimenin ilk harfi < ise ve ikinci harfi / değilse stack içine atıyorum
- eğer kelimenin ilk harfi < ise ve ikinci harfi / ise
- stackin boş olup olmadığının kontrolunu yapıyorum çünkü boş ise ve öyle bir kelime tespit edilirse belki kapatılmayan yada sadece kapatılan taglar var demki
- stack boş değilse tag kelimesinin eşleşip eşlemedinin kontrolünü yapıp eğer eşleşiyorsa demek açılıp kapatılan bir tag var demeki ve onu stacktan çıkartılır değilse hiç açılmayan ama kapatılan bir tag varmış demeki ve ekrana html dosyasında taglarda bir hata olduğunu bilgisini verir ve bu işlemleri hepsini döngü içerisinde gerçekleştirdiği için bu şekilde bütün kelimeleri gezer ve stacka konulan bütün kelimelerin eşini bulup stacktan çıkarılır
- son olarak yukarıdaki döngü işlemi bittikten sonra yine stack in boş olup olmadığının kontrolu yaparak html dosyasında varsa bir hata bilgisini verir.

```
int main(){
    yigin *temp = NULL;
    FILE* ptr = fopen("file.html", "r");
    if (ptr == NULL) {
        printf("no such file.");
        return 0;
    }

    char buf[100];
    int i;
    while (fscanf(ptr, "%s", buf) == 1){
        if(buf[0] == '<' && buf[1] != '/'){
            add_to_stack(buf);
            //print_stack();
        }
        else if (buf[0] == '<' && buf[1] == '/'){
```

```

        if(sta != NULL){
            i = 2;
            while(buf[i] != '>'){
                if(buf[i] != sta -> html_tag[i-1]){
                    goto ben;
                }
                i++;
            }
            delete_from_stack();
            //print_stack();
        }
        else{
            printf("\n Html dosyasindaki açılan tüm
taglar kapatilmaddi \n");
            return 0;
        }

    }

    ben:

}

if ( sta != NULL) {
    printf("\nHtml dosyasindaki açılan tüm taglar
kapatilmadi\n");
}
else {
    printf("\nHtml dosyasindaki açılan tüm taglar
kapatildi\n");
}
return 0;
}

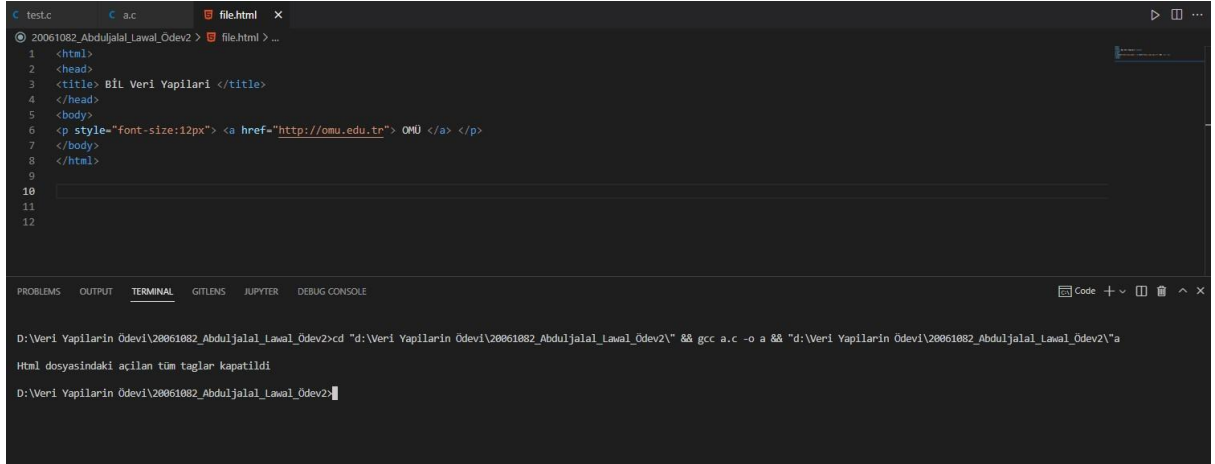
```

Çıktı örnekleri

Doğru örnek dosya içeriği

```
<html>
<head>
<title> BİL Veri Yapilari </title>
</head>
<body>
<p style="font-size:12px"> <a href="http://omu.edu.tr"> OMÜ </a> </p>
</body>
</html>
```

Program Çıktısı



The screenshot shows a Visual Studio Code editor with a C program file named 'file.html' open. The code in the editor is as follows:

```
1 <html>
2 <head>
3 <title> BİL Veri Yapilari </title>
4 </head>
5 <body>
6 <p style="font-size:12px"> <a href="http://omu.edu.tr"> OMÜ </a> </p>
7 </body>
8 </html>
9
10
11
12
```

The terminal output at the bottom shows the command executed and its result:

```
D:\Veri Yapilari\Odev1\20061082_Abduljalal_Lawal_Odev2>cd "d:\Veri Yapilari\Odev1\20061082_Abduljalal_Lawal_Odev2\" && gcc a.c -o a && "d:\Veri Yapilari\Odev1\20061082_Abduljalal_Lawal_Odev2\"a
Html dosyasındaki açılan tüm taglar kapatıldı
D:\Veri Yapilari\Odev1\20061082_Abduljalal_Lawal_Odev2>
```

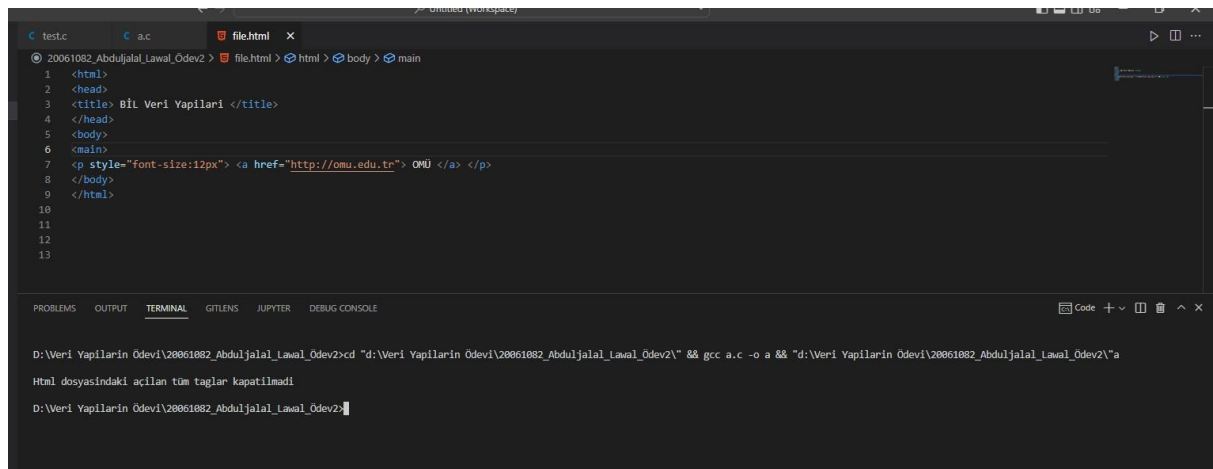
Yanlış örnek dosya içeriği

```
<html>
<head>
<title> BİL Veri Yapilari </title>
</head>
<body>

<main>

<p style="font-size:12px"> <a href="http://omu.edu.tr"> OMÜ </a> </p>
</body>
</html>
```

Program Çıktısı



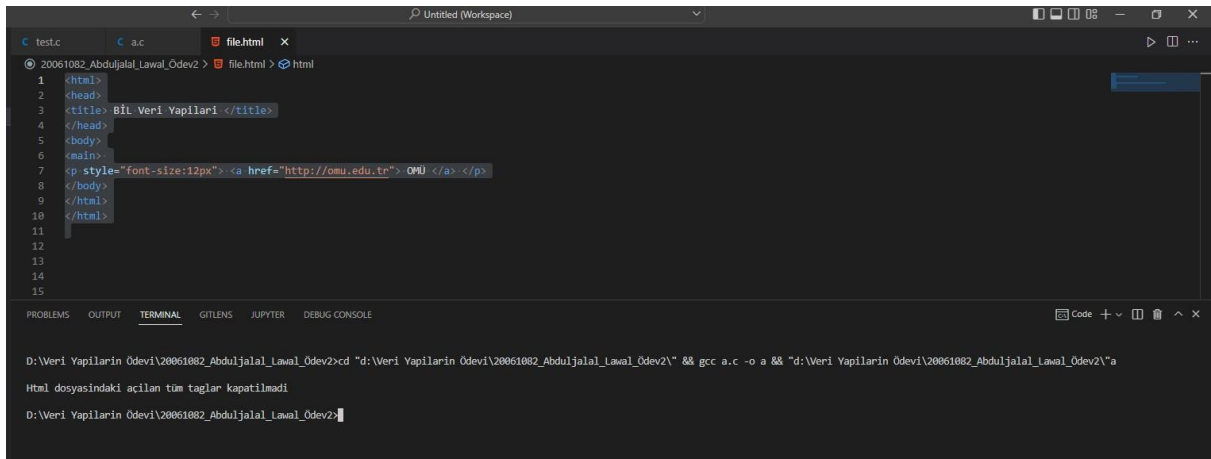
```
1 <html>
2 <head>
3 <title> BİL Veri Yapilari </title>
4 </head>
5 <body>
6 <main>
7 <p style="font-size:12px"> <a href="http://omu.edu.tr"> OMÜ </a> </p>
8 </body>
9 </html>
10
11
12
13
```

```
D:\Veri Yapılar\in Ödevi\20061882_Abduljalal_Lawal_Ödev2>cd "d:\Veri Yapılar\in Ödevi\20061882_Abduljalal_Lawal_Ödev2\" && gcc a.c -o a && "d:\Veri Yapılar\in Ödevi\20061882_Abduljalal_Lawal_Ödev2\"a
Html dosyasındaki açılan tüm taglar kapatılmadı
D:\Veri Yapılar\in Ödevi\20061882_Abduljalal_Lawal_Ödev2>
```

Yanlış örnek dosya içeriği

```
<html>
<head>
<title> BİL Veri Yapilari </title>
</head>
<body>
<main>
<p style="font-size:12px"> <a href="http://omu.edu.tr"> OMÜ </a> </p>
</body>
</html>
</html>
```

Program Çıktısı



The screenshot shows a code editor with a dark theme. The top panel displays a file named `file.html` with the following HTML code:

```
1 <html>
2 <head>
3 <title> Bil Veri Yapıları </title>
4 </head>
5 <body>
6 <main>
7 <p style="font-size:12px"> <a href="http://omu.edu.tr"> OMU </a></p>
8 </body>
9 </html>
10 </html>
11
12
13
14
15
```

The bottom panel shows the `TERMINAL` tab with the following output:

```
D:\Veri Yapilarin Odevi\20061082_Abduljalal_Lawal_Odev2>cd "d:\Veri Yapilarin Odevi\20061082_Abduljalal_Lawal_Odev2\" && gcc a.c -o a && "d:\Veri Yapilarin Odevi\20061082_Abduljalal_Lawal_Odev2\"a
Html dosyasındaki açılan tüm taglar kapatılmadı
D:\Veri Yapilarin Odevi\20061082_Abduljalal_Lawal_Odev2>
```