

# Yorum Satırları Üzerinden Kodun Açıklaması

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 struct n{
6     char adres[20]; //Sayfaları tutacağımız bölüm
7     int sayac; //Sayfaların kaç kere girdi olarak verildiğinin tutulduğu bölüm
8     struct n* next; //Sonraki düğümü (düğüm yoksa NULL) gösteren işaretçi
9     struct n* prev; //Önceki düğümü (düğüm yoksa NULL) gösteren işaretçi
10 };
11 typedef struct n node;
12
13 void bastir (node *r){ //Ekranı tüm listeyi yazdıran fonksiyon
14     while(r != NULL){
15         printf("%s,%d\t",r->adres,r->sayac);
16         r = r->next;
17     }
18     printf("\n");
19 }
20
21 node * ekle(node *r, char x[], int t, int l, int *dif){ //Yapılan her istek sonunda listenin yapısını değiştiren fonksiyon
22     node *second_last, *sil;
23
24     if (r == NULL){ //Linked list boş ise ilk olarak bu şekilde işleme alınıyor
25         r = (node *)malloc(sizeof(node));
26         r->next = NULL;
27         r->prev = NULL;
28         strcpy(r->adres,x); //Sayfa düğümün gözüne yazılıyor
29         r->sayac = 1; //İlk defa geldiği için sayac 1 oluyor
30         bastir(r); //Her girdiden sonra tüm liste ekrana yazdırılıyor
```

```
31     return r;
32 }
33
34 node *iter = r; //ekleme, çıkarma gibi işlemlerde yardımcı olmasına adına oluşturulmuştur
35
36 while(iter->next != NULL){ //Burada, son düğüm kadar olan düğümlere bakılıyor, daha önce girilen bir sayfa varsa sayac 1 artırılıyor
37     if ( !strcmp(x,iter->adres) ){
38         iter->sayac++;
39         bastir(r);
40         return r;
41     }
42     iter = iter->next;
43 }
44
45 if ( !strcmp(x,iter->adres) ){ //Son düğüm özel olarak burada değerlendiriliyor
46     iter->sayac++;
47
48     if ((iter->sayac > t){ //Burada girilen sayfaların eşik değerini aşmış aşmadığı kontrol ediliyor, aşmışsa istenildiği gibi 'head node' oluyor
49         iter->prev->next=NULL;
50         iter->prev=NULL;
51         iter->next=r;
52         iter->next->prev=iter;
53         bastir(iter); //Her girdiden sonra tüm liste ekrana yazdırılıyor
54         return iter;
55     }
56     //Eşik değerini aşmayan sayfalar buradan devam ediyor
57     bastir(r); //Her girdiden sonra tüm liste ekrana yazdırılıyor
58     return r;
59 }
60 //Eğer girilen sayfa daha önce girilmediyse yukarıdaki koşulu sağlayamıyor ve yoluna buradan devam ediyor
```

```
61 node *temp = (node *)malloc(sizeof(node)); //Daha önce girilmediği için burada yeni bir düğüm oluşturuluyor.
62 strcpy(temp->adres,x);
63 temp->sayac = 1;
64 (*dif)++; //dif değişkeni, buffer'da kaç farklı girdi olduğunun kontrolünde kullanılıyor, her farklı girdide 1 artırılıyor.
65
66
67 if (*dif>l){ //dif değişkeninin değeri eğer buffer kapasitesini geçerse, istenildiği gibi sondaki düğüm siliniyor
68     second_last = r;
69     while (second_last->next->next != NULL) //Sondan ikinci düğüme gidiyor
70         second_last = second_last->next;
71
72     sil = second_last->next; //Sonuncu düğüm silinmek için seçiliyor
73     second_last->next = NULL;
74     free(sil); //Ve sonuncu düğüm siliniyor
75 }
76
77 temp->prev = NULL; //Bir üstteki yoruma ek olarak istenildiği gibi 'head node' olarak atıyor
78 temp->next = r;
79 r->prev=temp;
80 bastir(temp); //Her girdiden sonra tüm liste ekrana yazdırılıyor
81 return temp;
82 }
83
84 void temizle(node *r){ //Bu fonksiyon yardımıyla, kullanıcı eğer isterse listeyi temizleyebiliyor
85     node *iter;
86
87     iter = r->next;
88     while (iter != NULL) { //Son düğüme kadar gidiliyor
89         r->next = iter->next; //Aradaki düğümler yalnız bırakılıp silmeye hazırlanıyor
90         iter->next = NULL;
```

```

91     free(iter);
92     iter = r->next; //Bir sonraki düğümü silmek için ilerleniyor
93 }
94 r->next=NULL;
95 r->prev=NULL;
96 free(r); //Son düğüm siliniyor
97 printf("Liste silinmiştir!");
98 }
99
100 int main (){
101     node *root;
102     root = NULL;
103     int i, t, l, dif=1, listeyi_sil=3, secenek=4;
104     char girdi[20], kontrol[]="cikis", ch='A';
105     FILE *fp;
106
107     while (secenek != 0 && secenek != 1){
108         printf("String olarak okumak için 0'a, 'input.txt' dosyasından okumak için 1'e basınız!\n");
109         scanf("%d",&secenek);
110         if ( secenek != 0 && secenek != 1)
111             printf("\nHatalı bir giriş yaptınız! Lütfen tekrar deneyiniz!\n");
112     }
113
114     if (secenek == 0){
115
116         printf("Sayfalar için esik degerini (T) giriniz: ");
117         scanf("%d",&t);
118         printf("Buffer'in kapasitesini (L) giriniz: ");
119         scanf("%d",&l);

```

```

121
122         printf("Lütfen sayfa girişi yapınız, sayfa girişi bittiginde 'cikis' yazıp işlemi bitirebilirsiniz.\n");
123         scanf("%s",girdi);
124         while ( strcmp(girdi,kontrol) ){
125             root = ekle(root,girdi,t,l,&dif);
126             scanf("%s",girdi);
127         }
128     }
129
130     else{
131         fp = fopen("input.txt","r"); //Dosya sorunsuz bir şekilde açılmadıysa program devam edemez
132         if (fp == NULL){
133             printf("Dosya açılmadı!");
134             return 0;
135         }
136         else{
137
138             do {
139                 fscanf(fp, "T=%d , L=%d", &t, &l); //input dosyasından eşik değeri ve buffer'ın kapasitesi alınıyor
140                 if (fscanf(fp,"%s",girdi)){
141                     root = ekle(root,girdi,t,l,&dif); //Girdiler alınıp işleniyor
142                 }
143             } while (!feof(fp));
144         }
145         fclose (fp);
146     }
147
148     while (listeyi_sil != 0 && listeyi_sil != 1){
149         printf("\nListeyi temizlemek ister misiniz? Temizlemek için 1'e, temizlemeden sonlandırmak için 0'a basınız!\n");
150         scanf("%d",&listeyi_sil);

```

```

151         if (listeyi_sil != 0 && listeyi_sil != 1)
152             printf("\nHatalı bir giriş yaptınız!");
153     }
154
155     if (listeyi_sil == 1){
156         temizle(root);
157         printf("\nProgram başarıyla sonlandı. Programdan çıkılıyor...");
158     }
159     else{
160         printf("\nProgram başarıyla sonlandı. Programdan çıkılıyor...");
161     }
162     return 0;
163 }

```

## Çıktılar

1. Girdi)

T=2 , L=3

AB BA CY AB CY XYZ BA XYZ BA

```
C:\Users\alias\Desktop\Hw1_18011091\18011091.exe
String olarak okumak için 0'a, 'input.txt' dosyasından okumak için 1'e basınız!
1
AB,1
BA,1      AB,1
CY,1      BA,1      AB,1
CY,1      BA,1      AB,2
CY,2      BA,1      AB,2
XYZ,1     CY,2      BA,1
XYZ,1     CY,2      BA,2
XYZ,2     CY,2      BA,2
BA,3      XYZ,2     CY,2
```

2. Girdi)

T=3, L=4

A B A AA BBB B A AB A B A BB

```
C:\Users\alias\Desktop\Hw1_18011091\18011091.exe
String olarak okumak için 0'a, 'input.txt' dosyasından okumak için 1'e basınız!
1
A,1
B,1      A,1
B,1      A,2
AA,1     B,1      A,2
BBB,1    AA,1     B,1      A,2
BBB,1    AA,1     B,2      A,2
BBB,1    AA,1     B,2      A,3
AB,1     BBB,1    AA,1     B,2
A,1      AB,1     BBB,1    AA,1
B,1      A,1      AB,1     BBB,1
B,1      A,2      AB,1     BBB,1
BB,1     B,1      A,2      AB,1
```