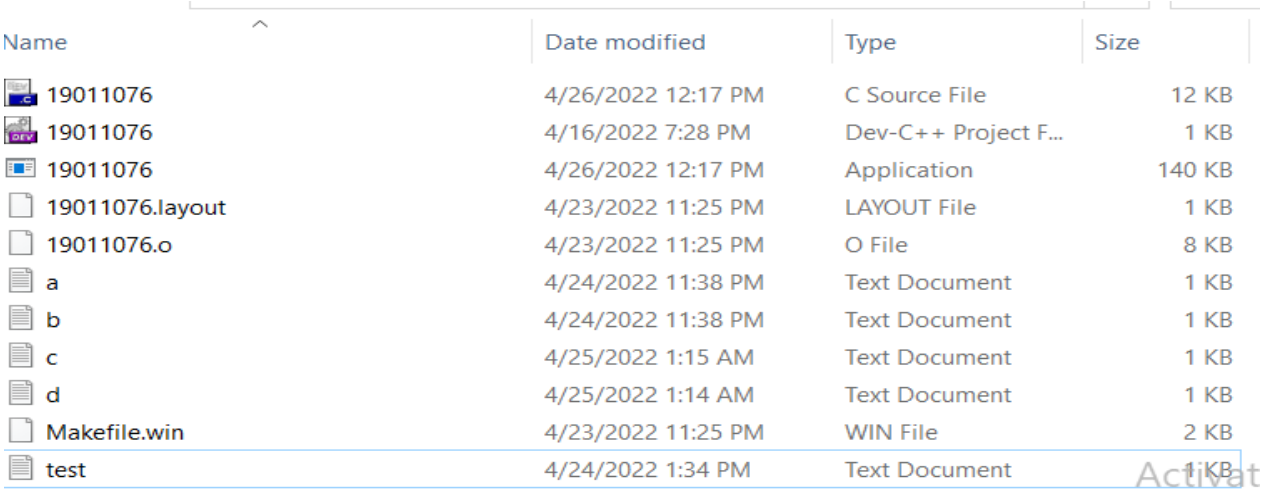
Veri Yapıları ve Algoritmalar

3. Odev

Dersin Eğitmeni: Mine Elif Karslıgil

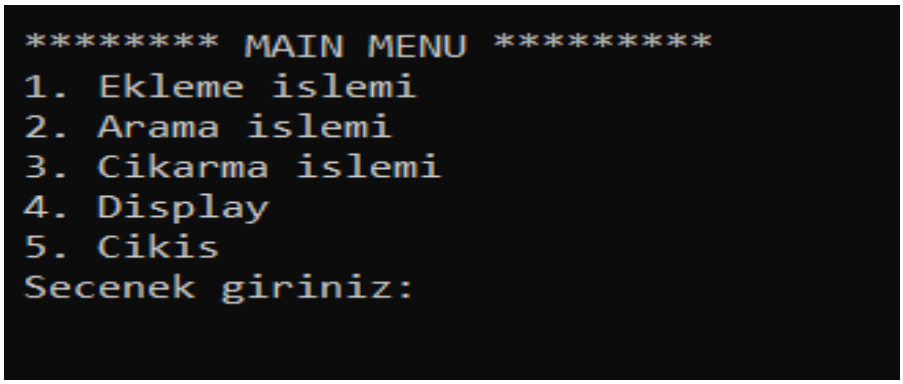
Ögrenci no:19011076

*C dosyanın bulundugu klasör:



Name	Date modified	Type	Size
19011076	4/26/2022 12:17 PM	C Source File	12 KB
19011076	4/16/2022 7:28 PM	Dev-C++ Project F...	1 KB
19011076	4/26/2022 12:17 PM	Application	140 KB
19011076.layout	4/23/2022 11:25 PM	LAYOUT File	1 KB
19011076.o	4/23/2022 11:25 PM	O File	8 KB
a	4/24/2022 11:38 PM	Text Document	1 KB
b	4/24/2022 11:38 PM	Text Document	1 KB
c	4/25/2022 1:15 AM	Text Document	1 KB
d	4/25/2022 1:14 AM	Text Document	1 KB
Makefile.win	4/23/2022 11:25 PM	WIN File	2 KB
test	4/24/2022 1:34 PM	Text Document	1 KB

*Menu görünümü:



*Ekleme işlemi:

```
******** MAIN MENU *********
 1. Ekleme islemi
 2. Arama islemi
 3. Cikarma islemi
 4. Display
 5. Cikis
 Secenek giriniz: 1
kelimeleri eklenecek olan dosyayi dosya_ismi.txt formatinda giriniz:a.txt

******** MAIN MENU *********
 1. Ekleme islemi
 2. Arama islemi
 3. Cikarma islemi
 4. Display
 5. Cikis
 Secenek giriniz: 4
beach-a.txt
was-a.txt
crowded-a.txt
snow-a.txt
leopards-a.txt
with
```

-Yukarıda “with” karşısı boş “with” kelimesini aradım zaman:

```
snow-a.txt
leopards-a.txt
with

 ********* MAIN MENU **********
 1. Ekleme islemi
 2. Arama islemi
 3. Cikarma islemi
 4. Display
 5. Cikis
 Secenek giriniz: 2
Aranacak olan kelimeyi giriniz:with
with kelimesinin gectigi dosyalar: a.txt

 ********* MAIN MENU **********
 1. Ekleme islemi
```

-Aşagıda diger dosyaları eklemeye devam ettim:

```
******** MAIN MENU *********
 1. Ekleme islemi
 2. Arama islemi
 3. Cikarma islemi
 4. Display
 5. Cikis
 Secenek giriniz: 1
kelimeleri eklenecek olan dosyayi dosya_ismi.txt formatinda giriniz:b.txt

******** MAIN MENU *********
 1. Ekleme islemi
 2. Arama islemi
 3. Cikarma islemi
 4. Display
 5. Cikis
 Secenek giriniz: 1
kelimeleri eklenecek olan dosyayi dosya_ismi.txt formatinda giriniz:c.txt

******** MAIN MENU *********
 1. Ekleme islemi
 2. Arama islemi
 3. Cikarma islemi
 4. Display
 5. Cikis
 Secenek giriniz: 1
kelimeleri eklenecek olan dosyayi dosya_ismi.txt formatinda giriniz:d.txt
```

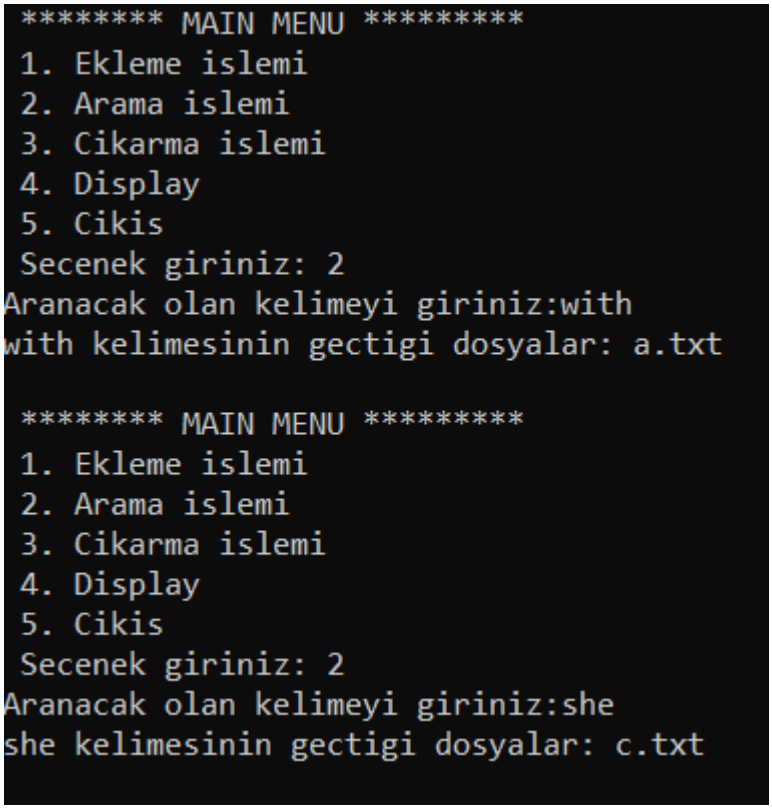
-Aşagıda display ettirdim:

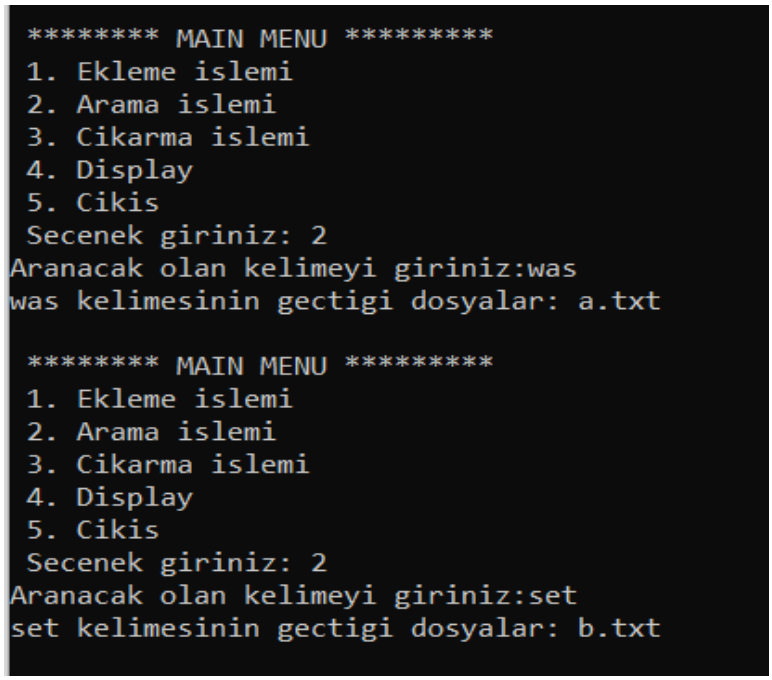
```
********* MAIN MENU **********
 1. Ekleme islemi
 2. Arama islemi
 3. Cikarma islemi
 4. Display
 5. Cikis
 Secenek giriniz: 4
beach-a.txt-c.txt-d.txt
and-b.txt
ankara-d.txt
was
crowded-a.txt
snow-a.txt
leopards-a.txt
had-b.txt
i-d.txt
in-d.txt
set
sells-c.txt
sea-c.txt
on-c.txt
love-d.txt
she
shells-c.txt
sun
so-b.txt
the-d.txt
with
```

Yanda bazı kelimelerin dosyaları bastırılmamış ama kelimeleri tek tek arattıgımda dosyalarını sıralıyor.

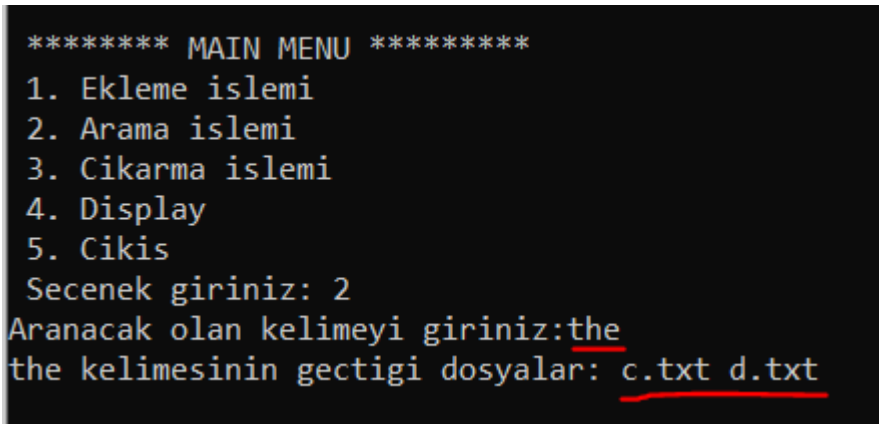
*Arama işlemleri:

-Son program üzerinden bazı kelimeleri aradım:





-“the” kelimesi iki dosyada gecmesine ragmen bütün kelimeleri display ettiğimde yanında bir dosyada var diye görünmüş. “The” search ettiğimde:



Not: Bunun gibi bazı yerlerde kod beklendiği gibi dogru çalışmıyor bunların sebebini ben cözemedim. Ama strtok fonksiyonundan dolayı dogru çalışmadığını düşünüyorum.

*pdf devam ediyor.

*Silme islemi:

-d.txt ve b.txt sırasıyla silmeye calıstım:

```
******** MAIN MENU *********
1. Ekleme islemi
2. Arama islemi
3. Cikarma islemi
4. Display
5. Cikis
Secenek giriniz: 3
kelimeleri silinecek olan dosyayi dosya_ismi.txt formatinda giriniz:d.txt
```

```
kelimeleri silinecek olan dosyayi dosya_ismi.txt forma

******** MAIN MENU *********
1. Ekleme islemi
2. Arama islemi
3. Cikarma islemi
4. Display
5. Cikis
Secenek giriniz: 4
beach-a.txt-c.txt-
and-b.txt
was
crowded-a.txt
snow-a.txt
leopards-a.txt
had-b.txt
set
sells-c.txt
sea-c.txt
on-c.txt
she
shells-c.txt
sun
so-b.txt
the-
with

******** MAIN MENU *********
```

*b.txt slime islemi:

```
******** MAIN MENU *********
 1. Ekleme islemi
 2. Arama islemi
 3. Cikarma islemi
 4. Display
 5. Cikis
 Secenek giriniz: 3
kelimeleri silinecek olan dosyayi dosya_ismi.txt formatinda giriniz:b.txt
```

```
******** MAIN MENU *********
 1. Ekleme islemi
 2. Arama islemi
 3. Cikarma islemi
 4. Display
 5. Cikis
 Secenek giriniz: 4
beach-a.txt-c.txt-
was-a.txt
crowded-a.txt
snow-a.txt
leopards-a.txt
she-b.txt
sells-c.txt
sea-c.txt
on-c.txt
shells
the
with
```

-she silmemiş.

*Display fonkisyonunu yukarlarda kullandım zaten preorder şeklinde bastırıyor.

*** Kodların acıklanması:

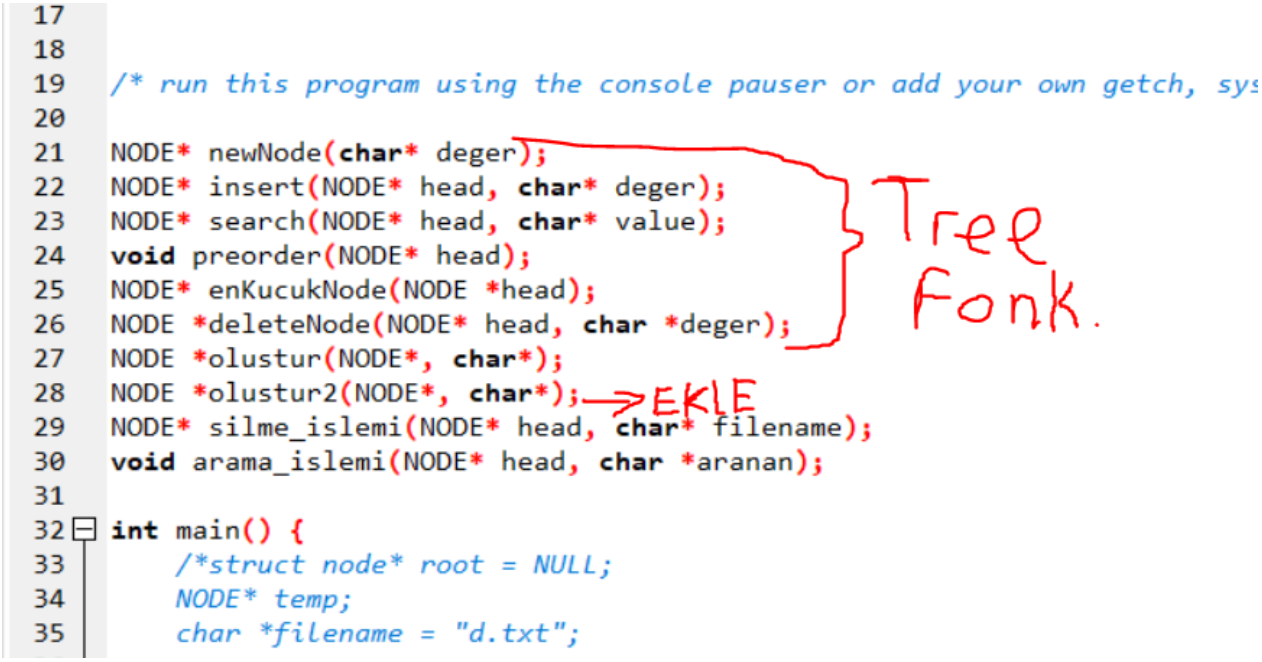
*Struct yapısı:

```
typedef struct node{
    char *data;
    struct node* left;
    struct node* right;
    char dizi[10][20];
    int top;
}NODE;
```

-burda data kellimeleri tutuyor. Left right dallanmayı saglıyor.

-dizi kelimelerin gectiği dosyaları tutuyor(bu yapıyı linked list ile yapmaya çalıştım ekleme slime kolay olsun diye sonra geri buna döndüm)

-top degeri silime ve ekleme işlemlerin kullanılmak üzere toplam dosya sayısını tutuyor.




```
17
18
19  /* run this program using the console pauser or add your own getch, sys
20
21  NODE* newNode(char* deger);
22  NODE* insert(NODE* head, char* deger);
23  NODE* search(NODE* head, char* value);
24  void preorder(NODE* head);
25  NODE* enKucukNode(NODE *head);
26  NODE *deleteNode(NODE* head, char *deger);
27  NODE *olustur(NODE*, char*);
28  NODE *olustur2(NODE*, char*);
29  NODE* silme_islemi(NODE* head, char* filename);
30  void arama_islemi(NODE* head, char *aranan);
31
32  int main() {
33      /*struct node* root = NULL;
34      NODE* temp;
35      char *filename = "d.txt";
```

-burdaki newNode fonksiyonundan deleteNode fonksiyonuna kadar hepsi binary search tree fonksiyonları bunları derste gördügümüz aynı şeyler hata olmadıgını düşünüyorum.

-diger fonksiyonlar ekleme arama slime işlemlerini yapıyor. Olustur2 fonksiyonu ekleme işlemini yapıyor fonksiyon isimini anlamlı vermemişim.

-iki tane olustur fonksiyonu var olustur ve olustur2 sadece olustur olan ıpdf sonunda acıkladım.

-tek tek fonksiyonlara bakalım.

*Olustur2(ekleme işlemi) :

```
NODE* olustur2(NODE* head, char* filename){
    int i;
    NODE* temp;
    char *kelime, *token;
    kelime = (char*)malloc(255*sizeof(char));

    FILE *oku;
    oku = fopen(filename, "r");

    fgets(kelime, 255, oku);

    fclose(oku);

    token = strtok(kelime, " ");

    if(token!=NULL){
        if(search(head, token) != NULL){
            temp = search(head, token);
            for(i=0;i<temp->top;i++){
                if(esit(temp->dizi[i], filename)){
                    return head;
                }
            }
            strcpy(temp->dizi[temp->top], filename);
            temp->top++;
        }else{
            head = insert(head, token);
            temp = search(head, token);
            strcpy(temp->dizi[temp->top], filename);
            temp->top++;
        }
    }

    while(token != NULL){
        token = strtok(NULL, " ");
        if(token!=NULL){
            if(search(head, token) != NULL){
                temp = search(head, token);
                for(i=0;i<temp->top;i++){
                    if(esit(temp->dizi[i], filename)){
                        return head;
                    }
                }
                strcpy(temp->dizi[temp->top], filename);
                temp->top++;
            }else{
                head = insert(head, token);
                temp = search(head, token);
                strcpy(temp->dizi[temp->top], filename);
                temp->top++;
```

-olustur 2 fonksionu aldıgı filename isimli dosya icerisindeki kelimeleri strtok ile tokenlere ayırıp tek tek BST'ye insert ediyor bunu yaparken eger daha once eklenmiş mi diye search edip bakıyor.

*arama fonksiyonu:

```
void arama_islemi(NODE* head, char *aranan){
    NODE* temp;

    if(aranan == NULL){
        printf("aranan kelime NULL degerde");
    }

    temp = search(head, aranan);

    if(temp == NULL){
        printf("aranan kelime treede bulunmuyor\n");
    }
    else{
        if(temp->top > 0){
            printf("%s kelimesinin gectigi dosyalar:",aranan);
        }
        else{
            printf("%s kelimesinin gectigi dosyalar yazdirilmadi\n",aranan);
        }

        int i;
        for(i=0;i<temp->top;i++){
            if(temp->data != " "){
                printf(" %s",temp->dizi[i]);
            }
        }
        printf("\n");
    }
}//end of arama_islemi fonksiyonu
```

*slime fonksiyonu:

```
NODE* silme_islemi(NODE* head, char* filename){
    char *cumle =(char*)malloc(128*sizeof(char));
    char *token;
    int i;

    NODE* temp = head;;
    FILE *oku = fopen(filename, "r");

    fgets(cumle, 127, oku);

    fclose(oku);

    token = strtok(cumle, " ");

    temp = search(head, token);

    if(temp == NULL){
        //pass
    }
    else{
        if(temp->top <= 1){
            temp = deleteNode(head, token);
        }
        else{
            for(i=0;i<temp->top;i++){
                if(esit(temp->dizi[i], filename)){
                    strcpy(temp->dizi[i], " ");//degistir
                }
            }
        }


        while(token != NULL){
            token = strtok(NULL, " ");

            if(token != NULL){

                temp = search(head, token);

                if(temp == NULL){
                    //pass
                }else{
                    if(temp->top <= 1){
                        temp = deleteNode(head, token);
                    }
                    else{
                        for(i=0;i<temp->top;i++){
                            if(esit(temp->dizi[i], filename)){
                                strcpy(temp->dizi[i], " ");//degistir
                            }
                        }
```

*slime işlemi: struct yapımda bir top degeri tanımladım bu her kelimenin kaç tane dosyada gectiğinin sayısını tutuyor. BST üzerinde bir kelimeyi sileceksem top degeri 1'in altındaysa deleteNode fonksiyonunu cagırıp cocuk sayısına göre siliyorum eger toplam dosya sayısı 1'in üzerindeyse dosyaları gezip sadece dosya ismini siliyorum.

*display(preorder fonksiyonu):

```
void preorder(NODE* head){
    int i;
    if(head!=NULL){
        printf("%s",head->data);
        while(i<head->top){
            printf("-%s",head->dizi[i]);
            i++;
        }
        printf("\n");
        preorder(head->left);
        preorder(head->right);
    }
}
```

*olustur fonksiyonu ilk calıştırıldıgında acıkladığım gibi eger a b c d txt dosyaları hazır degilse bize gönderdiğiniz test.txt dosyasından abcd txtlerini olusturmak için bir kez kullanılabilir.