



**T.C.
SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR ve BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**VERİ YAPILARI DERSİ
ARA SINAV PROJESİ – RAPOR**

**İKİ YÖNLÜ DAİRESEL BAĞIL LİSTE İLE OBEB BULARAK
ŞİFRELEME YAPAN C++ KONSOL UYGULAMASI**

Hazırlayan Öğrencinin;

Adı: Oğuzhan

Soyadı: Tohumcu

Numarası: B181210397

Serdivan / SAKARYA

Ağustos, 2020

İSTENENLER ve ALGORİTMA

Elimizde şifrelenecek bir metnin ASCII kod karşılıklarının bulunduğu bir metin dosyası mevcuttur. Bu metin dosyasını açıp her bir karakteri tek tek bir listeye atmamız isteniyor. Bu işlemi yaparken mevcut algoritmaya göre eklememiz gerekiyor.

Algoritmaya göre; gelen harfin ASCII kod karşılığı o an listede bulunan sayılar ile listenin başından başlayarak karşılaştırılır. Eğer liste boş ise sayımız direkt olarak listeye eklenip 'enbuyukobeb' 0 olarak atanır. Diğer sayılar için karşılaştırma yapılırken; elimizde bulunan sayı ile listede geldiğimiz sıradaki sayının **OBE**'i alınır. Bu obeb 'enbuyukobeb' den büyük ise, listenin diğer elemanlarına bakılmaz. Listedeki sayımız ile elimizde bulunan sayının modu alınır. Bu mod 0 çıkarsa elimizdeki sayı, listede geldiğimiz sıranın sağ tarafına eklenir. Eğer mod 0'dan farklı bir sayı çıkar ise, bu kez mod sayısı kadar listenin sol tarafına doğru ilerlenir.

İlerleme (iteration) sonucunda, listenin en başına gelmişse elimizdeki sayı listenin en başına eklenir. Eğer ilerleme sonucunda listenin başka bir yerine gelindi ise, bu defa sayımız gelineen sıranın sağ tarafına eklenir. Listeyi gezerken, elimizdeki sayı ile listedeki hiçbir sayının **OBE**'i 'enbuyukobeb' den büyük değilse ve listenin sonuna gelindi ise bu kez listenin sonunda bulunan sayı ile elimizdeki sayının modu alınır. Bu mod 0 çıkarsa yine sayımız, listede gelineen sayının sağ tarafına eklenir. Mod 0'dan farklı bir sayı çıkarsa; listedeki mevcut sayıdan sola doğru mod sayısı kadar ilerlenir. Bu ilerleme ile listenin başına gelindi ise, sayımız listenin en başına eklenir. Fakat bu ilerleme ile listenin farklı bir yerine gelindi ise sayımız, gelineen yerin sağ tarafına eklenir.

Bu şekilde dosyadaki bir satırda bulunan tüm sayılar algoritmaya göre listeye eklenir. Bu işlem bitirildiğinde listenin başından başlayarak, listenin içerisindeki sayıların **ASCII** kod karşılıkları ekrana yazdırılır. Daha sonra dosyanın ikinci satırına geçilir. Ve aynı algoritma ile tüm işlemler tekrarlanır. Programımız çalışmayı bitirdiğinde, dosya içerisindeki tüm şifreleri öğrenmiş oluyoruz.

YAZILIM GELİŞTİRME ve ÖĞRENİLENLER

Bu projeyi yaparken birçok konu öğrendim. Büyük, karmaşık yapıları bir proje kodlarken; tüm kodların tek bir dosyada olması sorun teşkil eden bir durumdur. Örneğin; projenin bir kısmında değişiklik yapılacağı zaman, eğer projemizi tek bir dosya içerisinde kodladıysak bu değişikliği yapmak biraz zor olabiliyor.

Birbirine bağımlı çok fazla kod parçası olduğu için tek bir değişiklik tüm projenin bozulmasına yol açabilir. Bu yüzden header dosyaları kullanılabilir. Bir projede yapılacak işleri küçültüp, farklı farklı dosyalara bölmek ve anlamlı isimlendirdiğimizde işlerimizi bir hayli hızlandırır. Projenin bir kısmında değişiklik yapılacağı zaman sadece o kısmın bulunduğu dosya içerisinde değişikliklerimizi yaparız ve projenin diğer parçalarının bozulmasını önleriz.

Diğer bir öğrendiğim konu ise; dosyadan aldığımız satır metnini direkt olarak kullanamıyoruz. Burada 2 problem var.

İlki, gelen satır verileri string olarak geliyor. Diğer ise, satırdaki tüm veriler tek bir veri olarak geliyor. Bu yüzden öncelikle metin boşluk karakterine göre parçalara ayırıp, int'e çevirerek bir diziye eklemeliyiz. Bu işlemi aşağıda bulunan koda göre yapıyoruz.

```

search = " ";
currPos = 0;
k = 0;
prevPos = 0;
sayilar = new CircularDoublyLinkedList();

do {
    spacePos = satir.find(search, currPos);
    if(spacePos >= 0) {
        currPos = spacePos;
        stringstream cevir(satir.substr(prevPos, currPos - prevPos));
        cevir >> arr[k];
        currPos++;
        prevPos = currPos;
        k++;
    }
} while(spacePos >= 0);

stringstream cevir (satir.substr(prevPos, satir.length()));
cevir >> arr[k];

```

Bu kod parçasını açıklamak gerekirse, ilk olarak bir dizi oluşturdum. Daha sonra aranacak kelime (bizde boşluk olacak) için bir değişken tanımlaması yaptım. Verilen metin içerisinde gezelmek için **spacePos** (boşluğun pozisyonu) ve **currPos** (metin içerisinde bulunduğumuz pozisyon) değişkenlerini tanımladım.

Daha sonra bir do-while döngüsü ile metin içerisinde gezinip boşluk karakterine kadar olan metni alıp, int olarak çevirerek **arr** isimli dizimize ekledim. Bu işlemler sonrasında sırasıyla dizi elemanlarımızı listeye eklemek üzere gönderdim.

EKRAN ÇIKTILARI

Sayilar.txt dosyasını okuttuğumda, aşağıdaki gibi doğru olarak ekran çıktılarını aldım.

```

1  /**
2  * @file CircularDoublyLinkedList.cpp
3  * @description Dairesel çift yönlü bağlı liste oluşturarak ek,
4  * @description ekleme, yazdırma, silme metodlarının implementasyonu
5  * @course 1. Öğretim - A Grubu
6  * @assignment Vize Projesi
7  * @date 11.08.2020
8  * @author Oguzhan Tohumcu / oguzhan.tohumcu@ogr.sakarya.edu.tr
9  */
10
11 #include <iostream>
12 #include "CircularDoublyLinkedList.hpp"
13
14 using namespace std;
15
16 // Constructor
17 CircularDoublyLinkedList::CircularDoublyLinkedList()
18 {
19     head = NULL;
20     size = 0;
21     listObek = 0;
22 }
23
24 // OREK bulma işlemi
25 int CircularDoublyLinkedList::obek(int sayi1, int sayi2)
26 {
27     int buyukSayi;
28     if(sayi1 > sayi2)
29         buyukSayi = sayi1;
30     else
31         buyukSayi = sayi2;
32     for(buyukSayi; buyukSayi > 0; buyukSayi--) {
33         if( (sayi1 % buyukSayi == 0) && (sayi2 % buyukSayi == 0) )
34             break;
35     }
36     return buyukSayi;
37 }
38
39 // Gelen item'ın kurulan algoritma ile CDLL'e eklenmesi
40 void CircularDoublyLinkedList::insert(int item) {
41     int mod = 0;
42 }

```

```

PS C:\Users\Acer> cd .\Desktop\
PS C:\Users\Acer\Desktop> cd .\Vize_Projesi_B181210397\
PS C:\Users\Acer\Desktop\Vize_Projesi_B181210397> mingw32-make
g++ -c -o ./lib/CircularDoublyLinkedList.o -I ".\include" ./src/CircularDoublyLinkedList.cpp
g++ -c -o ./lib/ProgramController.o -I ".\include" ./src/ProgramController.cpp
g++ -c -o ./lib/Program.o -I ".\include" ./src/Program.cpp
g++ ./lib/CircularDoublyLinkedList.o ./lib/ProgramController.o ./lib/Program.o -o ./bin/program1
./bin/program1.exe
Sifre: 2?yA0AAJ3KKOU\SRb
Sifre: A2_?yA0AAJ3KKOUFU\s9S`Rbbx
Sifre: yAAAKSR
Press any key to continue . . .

```

500 satırlı ve her satırında 250 sayı bulunan Sayilar2.txt dosyasını dendiğimde, şekildeki gibi büyük boyutlu ve algoritmaya göre doğru çalışan bir çıktı elde ettim.

